

# Social Choice for Social Good

## Proposals for Democratic Innovation from Computer Science

**Paul Gözl**

CMU-CS-22-133

August 2022

Computer Science Department  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213

**Thesis Committee:**

Ariel Procaccia (Chair)

Anupam Gupta

Tuomas Sandholm

Avrim Blum (TTIC)

David Parkes (Harvard University)

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy.

---

This research was sponsored by the National Science Foundation under award numbers CCF-1525932, IIS-1714140, and CCF-1733556; by the Office of Naval Research under award numbers N000141613075 and N000141712428; by scholarships from the German Academic Scholarship Foundation and the Dr. Jürgen and Irmgard Ulderup Foundation; and by a J.P. Morgan Chase AI Research Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the US government, or any other entity.



---

© Paul Gözl (2022)

**Keywords:** Theoretical Computer Science; Artificial Intelligence; Computational Social Choice; Democratic Innovations; Deliberative Minipublics

# Abstract

Driven by shortcomings of current democratic systems, practitioners and political scientists are exploring *democratic innovations*, i.e., institutions for decision-making that more directly involve constituents. In this thesis, we support this exploration using tools from computer science, via three approaches: we *design* practical algorithms for use in democratic innovations, we mathematically *analyze* the fairness properties of proposed decision-making processes, and we *identify* extensions of such processes that satisfy desirable properties. Our work mixes techniques from computational social choice, algorithms, optimization, probabilistic modeling, and empirical analysis.

In Part I, we apply the first two approaches to *citizens' assemblies*, which are randomly selected panels of constituents who deliberate on a policy issue. We analyze existing algorithms for the random selection of these assemblies, and we design new algorithms for this task that are provably fair and now widely used in practice. In addition, we design algorithms for partitioning assembly members into deliberation groups, which allow more members to interact than before.

Part II identifies extensions to *liquid democracy* and *legislative apportionment*. First, we demonstrate that a variant of liquid democracy, in which agents are asked for two potential delegates rather than a single delegate, reduces the concentration of power observed in classic liquid democracy. Second, we extend legislative elections over parties to approval ballots, and give apportionment methods for this setting that satisfy strong proportionality axioms. Finally, we extend a proposal for the randomized apportionment of legislative seats over states to satisfy additional monotonicity axioms.

In Part III of this thesis, we engage with a specific policy topic, *refugee resettlement*. We design algorithms for allocating resettled refugees to localities in a country, which improves these refugees' chances of finding employment over the status quo and is now being used by a major US resettlement agency.



# Acknowledgements

I am deeply grateful to Ariel Procaccia, who has been a wonderful advisor, but also much more than this word would suggest. Thank you, Ariel, for introducing me to the tremendous breadth of your interests; for demonstrating the value of finding new questions, not just answers; and for bringing healthy skepticism to my optimistic suggestions and optimistic suggestions to my skeptical responses. Most of all, thank you for being excellent company through the emotional highs and lows of doing research. I could not have asked for a better advisor and can only hope to do justice to you as a role model.

Many thanks to the other members of my thesis committee — Anupam, Avrim, David, and Tuomas — for their perspective and support. I am very lucky to have closely worked with two of you: with Anupam during our work on [Chapters 2](#) and [3](#) and with Avrim during my internship at TTIC. Both of you have impressed me when you spotted connections that caught me by surprise and opened up entirely new directions for the project. David and Tuomas, thank you for your insightful questions and comments — I have not talked enough to either of you, but hope to make up for it!

Thank you to my lovely collaborators, which (besides those already named) are Narges Ahani, Yushi Bai, Jake Barrett, Gerdus Benadè, Markus Brill, Uri Feige, Bailey Flanigan, Kobi Gal, Nika Haghtalab, Daniel Halpern, Rose Hong, Anson Kahng, Greg Kehne, Lucas Leopold, Wanyi Li, Simon Mackenzie, Evi Micha, Dominik Peters, Tim Randolph, Gili Rusak, Ulrike Schmidt-Kraepelin, Nisarg Shah, Alex Teytelboym, Andy Trapp, Jamie Tucker-Foltz, and Lily Xu. Thank you for all the things you have taught me, for the ping pong of ideas that makes a research project, and for some late nights working together (in the office or on Zoom). A few names feel like they belong on this list but are not on there yet — let's add them soon? Besides my academic collaborators, I am equally grateful to the many practitioners I have had the pleasure of meeting, and in particular to Adam Cronkright, Brett Hennig, and Philipp Verpoort. Your vision, enthusiasm, and attention to detail are inspiring and motivate me to push further in my own work.

Since I arrived in Pittsburgh, I have felt welcome at CSD, and have been glad to work there. I realize that this did not happen on its own, and I thank everybody who has contributed: Thanks to the great staff, in particular to Deb Cavlovich, Pat Loring, and Charlotte Yano. Thank you to the students and faculty volunteering their time, skills, and houses to create social cohesion, and many thanks to those who push our community to be more inclusive and welcoming. I am also indebted to all those who set me up for CMU in the first place: those who encouraged me to develop my passions, who taught me to do math, computer science, and research, who saw possibilities for me that I did not see myself, and who vouched for me.

I also want to thank the MD4SG group on Civic Participation, which I have had the privilege of co-organizing over the past two years. I truly cherish this community of like-minded researchers, and I am again and again amazed by what each of you brings to group's conversations. Thank you, Anson, for your work in organizing this group so far, and thank you, Manon, for taking over his role right now.

To my friends in the States and Europe (whom I cannot list without doing somebody injustice): Life would be less joyful and less interesting without you. Thank you for conversations running over time, for hikes and bike rides, for visiting and hosting me, and for sharing food, music, and feelings. Through you, I have learned about the world, life, and life as a researcher, and I cannot thank you enough for that. You're always welcome, so please come visit!

Where even to begin thanking my family? Dear Micha, Willy, Mama, and Papa; cher Thomas et chère famille française; liebe Großfamilie und de facto Großfamilie, you all have been shaping my values, fostering my curiosity, and supporting me throughout my life. Thank you for the different places I can be at home at, and thank you for still surprising me with where life takes you. Finally, I deeply thank a person I love, who has a beautiful heart ♥



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of Thesis Contribution and Structure . . . . .	2
1.2 Real-World Impact . . . . .	9
1.3 Bibliographic Notes . . . . .	11
<b>PROPOSALS FOR CITIZENS' ASSEMBLIES</b>	<b>12</b>
<b>2 Fair Algorithms for Selecting Citizens' Assemblies</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.1.1 Related Work . . . . .	15
2.2 Contribution I: Algorithmic Framework . . . . .	15
2.2.1 Definitions . . . . .	15
2.2.2 Formulating the Optimization Task . . . . .	15
2.2.3 Solving the Optimization Task . . . . .	16
2.3 Contribution II: Deployable Selection Algorithm . . . . .	18
2.4 Effect of Adopting LEXIMIN over LEGACY . . . . .	19
2.5 Discussion . . . . .	20
<b>3 Neutralizing Self-Selection Bias</b>	<b>22</b>
3.1 Introduction . . . . .	22
3.1.1 Related Work . . . . .	24
3.2 Model . . . . .	24
3.3 Selection algorithm . . . . .	25
3.3.1 Algorithm Part I: Assignment of Marginals . . . . .	26
3.3.2 Algorithm Part II: Rounding of Marginals . . . . .	27
3.4 Learning Participation Probabilities . . . . .	28
3.5 Experiments . . . . .	29
3.6 Discussion . . . . .	32
<b>4 Benefits of Stratified Sampling</b>	<b>34</b>
4.1 Introduction . . . . .	34
4.1.1 Our Approach and Results . . . . .	35
4.1.2 Background on Stratified Sampling . . . . .	36
4.2 Model . . . . .	37
4.3 Warming Up in a Continuous Setting . . . . .	38
4.4 Main Result: The Variance of Stratified Sampling . . . . .	39
4.4.1 Block Rounding . . . . .	39
4.4.2 Variance Upper Bound . . . . .	40
4.5 General Selection Algorithms . . . . .	44

4.6	Experiments . . . . .	46
4.6.1	Random Stratification . . . . .	47
4.6.2	Case Study: Comparison of Stratification Methods . . . . .	47
4.7	Discussion . . . . .	50
<b>5</b>	<b>Improving Deliberation Groups</b>	<b>52</b>
5.1	Introduction . . . . .	52
5.1.1	Our Approach and Results . . . . .	53
5.1.2	Related Work . . . . .	53
5.2	Model . . . . .	54
5.3	Expressing the Table Allocation Problem as Concave Balancing . . . . .	55
5.4	Optimizing a Specific Saturation Function . . . . .	56
5.5	Simultaneously Optimizing All Saturation Functions . . . . .	58
5.6	Implementation and Empirical Results . . . . .	60
5.6.1	How Well Does the Greedy Algorithm Optimize Its Objective? . . . . .	61
5.6.2	Comparison across Table-Allocation Algorithms . . . . .	62
5.7	Discussion . . . . .	63
 <b>PROPOSALS FOR OTHER ASPECTS OF DEMOCRACY: LIQUID DELEGATION AND APPORTIONMENT</b>		<b>64</b>
<b>6</b>	<b>Avoiding the Concentration of Power in Liquid Democracy</b>	<b>65</b>
6.1	Introduction . . . . .	65
6.1.1	Our Approach and Results . . . . .	66
6.1.2	Related Work . . . . .	67
6.2	Algorithmic Model and Results . . . . .	68
6.2.1	Problem Statement . . . . .	69
6.2.2	Connections to Confluent Flow . . . . .	69
6.2.3	Algorithms . . . . .	71
6.2.4	Hardness of Approximation . . . . .	72
6.3	Probabilistic Model and Results . . . . .	75
6.3.1	The Preferential Delegation Model . . . . .	76
6.3.2	Lower Bounds for Single Delegation ( $k = 1, \gamma = 0$ ) . . . . .	77
6.3.3	Upper Bound for Double Delegation ( $k = 2, \gamma = 0$ ) . . . . .	80
6.4	Empirical Results . . . . .	85
6.4.1	Multiple vs. Single Delegations . . . . .	85
6.4.2	Evaluating Mechanisms . . . . .	86
6.5	Discussion . . . . .	90
<b>7</b>	<b>Party-List Apportionment with Approval Votes</b>	<b>93</b>
7.1	Introduction . . . . .	93
7.1.1	Related Work . . . . .	94
7.1.2	Relation to Other Settings . . . . .	95
7.1.3	Contributions . . . . .	96
7.2	Model . . . . .	96
7.3	Constructing Party-Approval Rules via Multiwinner Voting Rules . . . . .	97
7.3.1	Preliminaries . . . . .	98
7.3.2	Embedding Party-Approval Elections . . . . .	98
7.3.3	PAV Guarantees Core Stability . . . . .	99
7.4	Constructing Party-Approval Rules via Portioning and Apportionment . . . . .	102
7.4.1	Preliminaries . . . . .	102
7.4.2	Composed Rules That Fail EJR . . . . .	104

7.4.3	Composed Rules That Satisfy EJR . . . . .	105
7.5	Computational Aspects . . . . .	106
7.6	Discussion . . . . .	109
<b>8</b>	<b>Monotone Randomized Apportionment</b>	<b>110</b>
8.1	Introduction . . . . .	110
8.1.1	Our Approach and Results . . . . .	112
8.1.2	Related Work . . . . .	113
8.2	Model . . . . .	114
8.3	Population Monotonicity . . . . .	116
8.3.1	Population Monotonicity Is Incompatible with Quota . . . . .	116
8.3.2	A Population Monotone and Ex Ante Proportional (But Not Quota) Method . . . . .	117
8.4	House Monotonicity . . . . .	119
8.4.1	Examples of Pitfalls . . . . .	119
8.4.2	Cumulative Rounding . . . . .	120
8.4.3	House Monotone, Quota-Compliant, and Ex Ante Proportional Apportionment . . . . .	122
8.5	Proof of Cumulative Rounding . . . . .	125
8.6	Other Applications of Cumulative Rounding . . . . .	128
8.6.1	Sortition of the European Commission . . . . .	128
8.6.2	Repeated Allocation of Courses to Faculty Or Shifts to Workers . . . . .	130
8.7	Discussion . . . . .	131
	 <b>PROPOSAL FOR REFUGEE RESETTLEMENT</b>	 <b>132</b>
<b>9</b>	<b>Online Refugee Placement</b>	<b>133</b>
9.1	Introduction . . . . .	133
9.1.1	Related Work . . . . .	134
9.1.2	Chapter Outline . . . . .	136
9.2	Institutional Background . . . . .	136
9.3	Model . . . . .	137
9.4	Online Bipartite Matching ( $s_i = 1$ ) . . . . .	138
9.4.1	Algorithmic Approach . . . . .	138
9.4.2	Empirical Evaluation . . . . .	141
9.5	Non-Unit Cases ( $s_i \geq 1$ ) . . . . .	144
9.6	Batching . . . . .	145
9.6.1	Empirical Evaluation . . . . .	146
9.7	Uncertainty in the Number of Future Arrivals . . . . .	147
9.7.1	Relying on Capacities . . . . .	147
9.7.2	Arrival Misestimation on Bootstrapped Data and Incorporating Uncertainty . . . . .	150
9.7.3	Better Knowledge of Future Arrivals . . . . .	152
9.8	Implementation in Annie™ MOORE . . . . .	153
9.9	Discussion . . . . .	155
	 <b>CONCLUSIONS</b>	 <b>156</b>
<b>10</b>	<b>Conclusions and Future Work</b>	<b>157</b>
10.1	How Can Computer Science Guide Democratic Innovations? . . . . .	157
10.2	Future Directions for the Study of Democracy in Computer Science . . . . .	159
	 <b>Bibliography</b>	 <b>162</b>

**APPENDIX**

**183**

<b>A</b>	<b>Appendix for Chapter 2: Fair Algorithms for Selecting Citizens' Assemblies</b>	<b>184</b>
A.1	Desiderata for Sortition in the Political Science Literature . . . . .	184
A.1.1	Properties of Idealized Sortition . . . . .	184
A.1.2	Beyond Idealized Sortition, and the Objective of Maximal Fairness . . . . .	187
A.2	Additional Figures and Tables . . . . .	188

# Introduction

# 1

Imagine living in a society in which democracy lived up to its ideals: constituents would have equal power in making collective decisions, these decisions would account for all constituents' perspectives, and constituents could accept decisions as legitimate even when they personally disagree with them. Such a society could turn problems that seem insurmountable today into opportunities: instead of climate change, we could live in a world of clean air and sustainable wealth; in a pandemic, collectively balance safety and freedom; establish policies that mitigate the harms of racism, sexism and other forms of prejudice; and provide equitable opportunity to all.

Current democratic systems claim that, by electing representatives, they already implement democracy to its full extent, and yet they fall far short of the ideals above. Following the "folk theory" of democracy [AB16], competitive elections empower a majority of voters to choose their preferred policy by voting for the right candidate, elections incentivize representatives to defend their constituents' interests, and the neutral process of voting should make decisions legitimate. The data, however, show that each of these points comes with major caveats,<sup>1</sup> which substantially reduce society's ability to act in concert. Improving the functioning of democracy is urgently needed, because democracy has been losing ground to authoritarian forms of governance over the last decade: according to the Varieties of Democracy Institute, the "level of democracy enjoyed by the average global citizen in 2021 is down to 1989 levels," eradicating "30 years of democratic advances" [BAL+22].

Motivated by this growing threat to global democracy, political scientists and practitioners have been developing *democratic innovations* [EE19; Lan20; Smio9], proposed institutions that would more directly involve constituents in decision making. When attempted in practice, some of these innovations have achieved resounding success, suggesting that they can become powerful instruments for strengthening democracy. Below we introduce three prominent democratic innovations, the first and third of which feature in this thesis:

**Citizens' assemblies:** A panel of randomly selected constituents deliberates on a policy question and formulates joint recommendations to decision makers. Empirical research shows that deliberation in citizens' assemblies is of high quality, that citizens' assemblies find common ground despite polarization, and that they produce decisions rooted in considered judgement [DBC+19]. Due to this success, citizens' assemblies have become more numerous and prominent around the world [OEC20].

**Participatory budgeting:** Constituents deliberate and vote on how part of the public budget should be spent, a decision that is classically made through intransparent negotiations between politicians. In its birthplace, the city of Porto Alegre in Brazil, participatory budgeting "has deepened democracy, promoted social justice, improved how local states function, and made government officials accountable to their constituents" [Wamo7], and more than 1 500 cities worldwide have since adopted forms of participatory budgeting [GB20].

1.1 Overview . . . . .	2
1.2 Real-World Impact . . . . .	9
1.3 Bibliographic Notes . . . . .	11

[AB16] Achen and Bartels (2016): *Democracy for Realists*.

1: Voting behavior is influenced by irrelevant factors, such as the result of sports games [HMM10]. Across US states, policy and the majority preference agree no more frequently than they disagree [LP12]. A large share of the US population is dissatisfied with all candidates in the election [Fin16]. Politicians may dramatically misjudge their constituents' opinions [BS18], and, across 27 democratic nations, only a median of 35% agree that their elected officials "care what ordinary people think" [WSC19]. Around the world, constituents have low trust in legislatures and political parties [OEC22], which has been shown to increase constituents' support for breaking the law [MH11]. Constituents in democracies around the world feel that they do not have a say in government decision making [OEC22].

[BAL+22] Boese et al. (2022): *Democracy Report 2022*.

[EE19] Elstub and Escobar (2019): *Handbook of Democratic Innovation and Governance*.

[Lan20] Landemore (2020): *Open Democracy*.

[Smio9] Smith (2009): *Democratic Innovations*.

[DBC+19] Dryzek et al. (2019): *The Crisis of Democracy and the Science of Deliberation*.

[OEC20] OECD (2020): *Innovative Citizen Participation and New Democratic Institutions*.

[Wamo7] Wampler (2007): *Participatory Budgeting in Brazil*.

[GB20] Ganuza and Baiocchi (2020): *The Power of Ambiguity*.

**Liquid democracy:** Constituents can directly vote on many decisions as in direct democracy, or may temporarily delegate their vote to other constituents. This is meant to provide the practical benefits of political representation while “maximally [lowering] the barriers to entry to the status of elected representative” [Lan20]. Liquid democracy has been implemented inside political parties, within companies, and as a form of civic participation in regional governments [Pau20].

In this thesis, we use techniques from computer science to support and contribute to this crucial process of democratic innovation. We do so by following three approaches:

- ▶ We design practical algorithms for use in democratic innovations (Chapters 2, 3, 5, and 9),
- ▶ we mathematically analyze the fairness properties of proposed decision-making processes (Chapter 4), and
- ▶ we identify extensions of such processes that satisfy desirable properties (Chapters 6 to 8).

In Section 10.1, we reflect more broadly on what computer science can offer to support democratic innovations. As will be illustrated throughout this thesis, researching democratic innovations is also fruitful for computer science: our work contributes new questions, technical results, and opportunities for practical impact to the field.

## 1.1 Overview of Thesis Contribution and Structure

### Part I: Practical and Theoretical Suggestions for Citizens’ Assemblies

This thesis puts a particular focus on citizens’ assemblies,<sup>2</sup> which I believe to be the most promising direction for augmenting democracy. This belief is rooted in a variety of observations made during my thesis research through studying political science and political theory, collaborating with practitioners in the selection of citizens’ assemblies, reading the final reports produced by citizens’ assemblies, and watching recordings of citizens’ assembly deliberation.

Based on similar impressions, prominent political scientists have advocated for proliferating the use of citizens’ assemblies [DBC+19; Fis09; Lan20; Man10], and an increasing number of public authorities now commission citizens’ assemblies [OEC20]. Some proponents suggest that randomly-selected deliberative bodies can play roles beyond the advisory role of classic citizens’ assemblies. For instance, schools in Bolivia have selected random student governments [PKC20], and regions in Belgium and France recently instituted permanent political bodies composed of random constituents [NR19; OEC21]. Beyond current experimentation, advocates have proposed designs in which groups of random citizens take on the role of union representatives [Pek19], legislative chambers [GW18], or hiring boards for positions in the executive branch [BS14].

[Lan20] Landemore (2020): *Open Democracy*.

[Pau20] Paulin (2020): *Ten Years of Liquid Democracy Research*.

2: Randomly selected deliberative bodies bear a variety of names including *citizens’ assemblies/panels/juries, consensus conferences, deliberative polls, and planning cells* [OEC20]. While we refer to these bodies as citizens’ assemblies throughout this thesis, our work applies to the broader category of *deliberative minipublics* [SS18b].

[DBC+19] Dryzek et al. (2019): *The Crisis of Democracy and the Science of Deliberation*.

[Fis09] Fishkin (2009): *When the People Speak*.

[Lan20] Landemore (2020): *Open Democracy*.

[Man10] Mansbridge (2010): *Deliberative Polling as the Gold Standard*.

[OEC20] OECD (2020): *Innovative Citizen Participation and New Democratic Institutions*.

[PKC20] Pek et al. (2020): *Democracy Transformed*.

[NR19] Niessen and Reuchamps (2019): *Designing a Permanent Deliberative Citizens’ Assembly*.

[OEC21] OECD (2021): *Eight Ways to Institutionalise Deliberative Democracy*.

[Pek19] Pek (2019): *Rekindling Union Democracy Through the Use of Sortition*.

[GW18] Gastil and Wright (2018): *Legislation by Lot*.

[BS14] Bouricius and Schecter (2014): *An Idealized Design for Government. Part 2*.

The first part of this thesis studies different aspects of citizens’ assemblies and derives suggestions for how they are currently organized or could be organized in the future. The first three chapters in this part focus on *sortition*, i.e., the random selection of a citizens’ assembly’s members. This work is driven by the tensions between two primary objectives in sortition: *representativeness*, which means that groups in the population should be present on the panel roughly in proportion to their share of the population, and *equality*, which means that no individual or group should receive preferential treatment over others. Each of these three chapters makes different assumptions about what assembly organizers can do, about the constraints on representativeness, and on what form of equality they strive for:

Chapter	Organizers select from...	Representativeness constraints...	Aim to equally include...
Chapter 2	a given pool of volunteers	are already given	volunteers in the pool
Chapter 3	a pool of volunteers produced by randomness and self-selection	can still be set	agents in the population
Chapter 4	the entire population	can still be set	unidentified groups in the population

Whereas these three chapters study the selection of the assembly members, Chapter 5 investigates a problem faced by practitioners later in the process, namely, how to partition the assembly members into discussion groups.

## Chapter 2: Fair Algorithms for Selecting Citizens’ Assemblies

Most organizations select the members of a citizens’ assembly through a three-step process, which is illustrated in Figure 1.1: First, practitioners send a large number of invitations to a random subset of the population, typically by random mail. Second, if the recipient of an invitation is willing to serve on a panel, they can opt into a *pool* of volunteers. Third, a *panel* of predetermined size is sampled from the pool.

This chapter develops multiple *selection algorithms*, algorithms which perform the third step of sampling the panel. As mentioned above, the process should satisfy the objectives of representativeness and equality. Following current practice, representativeness is enforced by a collection of *quotas* on different features,<sup>3</sup> which the chosen panel must deterministically satisfy. The principle of equality, by contrast, had not been systematically quantified before our work. We capture it through what we call a *fairness measure*, which is a concave objective function taking each pool member’s probability of selection as its input. Within what the representativeness constraints allow, the concave objective encourages that pool members be selected with similar probabilities.

In this chapter, we develop an algorithmic framework of selection algorithms that are *maximally fair*, i.e., which draw the panel with such a probability distribution that a given fairness measure is maximized. This framework combines techniques from convex optimization, column generation, and integer linear programming in a novel way, and yields practically efficient selection algorithms for a range of fairness measures inspired by the literature on fair division. Furthermore, we develop an efficient implementation for one selection algorithm, LEXIMIN. Using data from ten citizens’ assemblies, we demonstrate that this algorithm

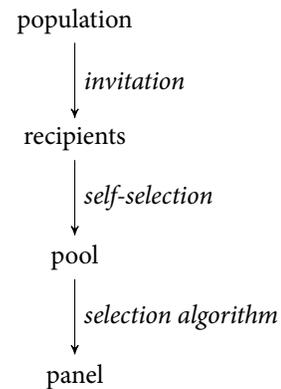


Figure 1.1: Steps of panel selection.

3: For example, the quotas might require that between 14 and 16 of the 30 panel members be female, that exactly 5 members be above 65 years old, and so forth.

yields vastly more equitable selection probabilities than the previous state of the art (without compromising on representativeness). Multiple organizations have adopted LEXIMIN and have used it for selecting dozens of assemblies around the world.

### Chapter 3: Neutralizing Self-Selection Bias

A possible objection to the approach taken in the previous section is that the notion of equality it provides only extends to members of a given pool. This is in line with the thinking of some practitioners we have talked to, for whom fairness among *volunteers* is the objective. But an equally valid perspective is that the objective is to give members of the *population* an equal chance of making it through all three stages of the pipeline (Figure 1.1), which we call their *end-to-end probability*.<sup>4</sup> In particular, this perspective suggests that, when an agent who is a priori unlikely to self-select into the pool joins the pool against the odds, the selection algorithm should select this agent with particularly high probability in order to make up for the agent’s tendency not to join the pool.

In this chapter, we develop a methodology for sampling a citizens’ assembly that approximately equalizes end-to-end probabilities. That is, we design a polynomial-time sampling algorithm, and prove that, as the number of invitations sent out grows large enough, (1) the end-to-end probabilities when using the selection algorithm converge to an equal value, (2) the process satisfies quotas with increasingly tight gaps between upper and lower quotas, and (3) the probability that the selection algorithm cannot produce a panel within these quotas<sup>5</sup> goes to zero. This algorithm is also based on column generation, but combines it with a famous result from discrepancy theory by Beck and Fiala [BF81].

Since the results in the previous section assumed that the organizers knew each pool members’ a-priori probability of opting in — which is not the case in practice — we complement these results with a learning approach for estimating such probabilities from data. The challenge in this learning task is that practitioners observe only the characteristics of those recipients who opt in (“cases”), not of those who decline to do so (“controls”). In many cases, survey data allows organizers to observe the features of a random sample of the population, but for this sample it is not known whether they would have opted into the pool if invited. In the language of econometrics, these data are “contaminated controls” [LI96], and there are generic, iterative ways of approximately estimating parameters in this setting. Assuming a natural parametric model for how an individual’s features determine their probability of opting in, we show that the maximum likelihood estimator can instead be found by minimizing a single convex function in our setting. We apply this learning algorithm to data from a real citizens’ assembly combined with public survey data. The determined parameters of the parametric model allow us to create a semi-synthetic population, for which the selection algorithm gets very close to equalizing end-to-end probabilities.

### Chapter 4: Benefits of Stratified Sampling

The selection process in the last two chapters was quite complicated, mainly because of the high level of self-selection in the selection pipeline (Figure 1.1).

4: In part, these perspectives differ in whether declining an invitation is seen as a voluntary action signaling disinterest or whether it shows that the individual is unable to participate. Typically, organizers financially compensate panel members, cover travel and accommodation costs, and provide childcare to enable more individuals to participate. See Jacquet [Jac17] for a study of why agents decline to participate.

5: Since who receives the invitations and whether the recipients opt in is random, this event cannot be entirely excluded.

[BF81] Beck and Fiala (1981): “Integer-making” Theorems.

[LI96] Lancaster and Imbens (1996): *Case-Control Studies with Contaminated Controls*.

This chapter studies sortition in an alternative setting without self-selection, i.e., where the selection algorithm can directly choose the panel members from the population. This setting might be either applicable if serving on a panel is made mandatory as some authors suggest [Leio4; Mal15], or approximately applicable if practitioners can substantially raise the rate of opting in.<sup>6</sup> In principle, our selection algorithms from Chapter 2 still apply, by considering the entire population as a pool, albeit at prohibitive computational cost. More importantly, this approach would be overly complicated given that, in a model without self-selection, there is no tension between equal selection probabilities and representativeness.

In this chapter, we compare the two selection algorithms generally discussed in the political theory literature: *uniform sampling* and *stratified sampling*. Since both algorithms select constituents with perfectly equal probabilities, we study how well the algorithms satisfy representativeness. Uniform sampling satisfies a powerful *stochastic* guarantee on representativeness: for any group  $M$  of agents in the population, the number of  $M$ 's representatives on the panel is a random variable whose distribution is concentrated around  $M$ 's proportional share of the panel. Thus, *any arbitrary group* in the population is likely represented close to proportionally in a (sufficiently large) panel selected through uniform sampling — not only groups that are externally identifiable, but also unobservable groups such as those defined by common beliefs or interests [Stoo8]. In contrast to this stochastic representativeness, stratified sampling satisfies *deterministic* guarantees on representativeness. For example, if half of the panel is drawn uniformly among women and half uniformly among men, each panel will be gender balanced.

But does the deterministic guarantee of stratified sampling come at the cost of worsening the stochastic representation for other groups in the population? We prove that the answer is essentially no: Stratified sampling (with careful treatment of indivisibilities) never increases the variance of any group  $M$ 's representation by more than a factor that is very close to 1.<sup>7</sup> Using data from the General Social Survey, we show that a careful stratification will actually decrease the variance of representation for other groups of interest, because features tend to correlate to some degree. Based on these observations, we recommend that, when selecting a citizens' assembly without self-selection bias, organizers should adopt stratified sampling rather than uniform sampling.

## Chapter 5: Improving Deliberation Groups

Finally, we turn our attention from the selection of the panel to a task that touches on the deliberation happening once the assembly convenes. For this deliberation, assembly members are typically subdivided into groups of 5–10 members each, which we refer to as *tables* since group members typically gather around a table to exchange experiences and arguments under professional moderation. Assemblies last for multiple deliberation sessions, and the composition of the tables is mixed up anew for each session. Our project of optimizing the assignment of tables started from conversations with the *Sortition Foundation*, a nonprofit organization recruiting citizens' assemblies, about a software tool they created for this purpose [Ver22]. According to our contacts, practitioners have two major

[Leio4] Leib (2004): *Deliberative Democracy in America*.

[Mal15] Malkopoulou (2015): *The Paradox of Democratic Selection*.

6: Some practitioners claim far higher opt-in rates for recruitment by phone rather than those typical for random mail. Moreover, should citizens' assemblies gain impact and visibility in the future, participation rates would likely increase [Jac17].

[Stoo8] Stone (2008): *Voting, Lotteries, and Justice*.

7: We also prove that this factor is tight, and that uniform sampling minimizes the maximum representation variance across groups  $M$ . Given that variances for stratified sampling are never much larger, stratified sampling is still near-optimal from this worst-case perspective.

[Ver22] Verpoort (2022): *GroupSelect*.

aims in determining these groups: each small groups should satisfy representativeness quotas like those in [Chapter 2](#), and practitioners want as many assembly members to meet as possible.

Informed by the Sortition Foundation’s existing approach, we capture this task as an optimization problem, in which we must determine a partition of the agents into representative tables, one partition for each of  $T$  many sessions. The objective is to maximize, summed up over all pairs of assembly members, a function of how often the pair meets across all sessions. We call such a function a *saturation function* and assume that it is monotone nondecreasing (“meeting more often does no harm”) and concave (“meetings between the same pair have diminishing returns”). For any given saturation function, a greedy algorithm that uses an ILP solver as a subroutine approximates the optimal objective value within a factor of  $1 - 1/e \approx 0.63$ . As we show on data from real citizens’ assemblies, instantiating the greedy algorithm with an appropriate saturation function yields a group-allocation algorithm that is practically efficient and dramatically outperforms the practitioners’ software in terms of its own metric of success.

To avoid the arbitrary choice of saturation function, we construct an alternative algorithm, which simultaneously approximates the objectives for all possible saturation functions within a factor of  $\Omega(1/\log T)$ . For a natural generalization of the group-allocation problem, we prove that this result is tight up to a  $\log \log T$  factor. Empirically, we show that this simultaneous-approximation approach continues to outcompete the practitioners’ algorithm, even though the greedy algorithms seem somewhat preferable in our data.

## Part II: Impulses for Other Aspects of Democracy: Liquid Delegation and Apportionment

In the second part of the thesis, we explore alternative approaches to democracy in areas other than citizens’ assemblies. First, we extend another democratic innovation, liquid democracy, in a way that shows promise for decreasing the concentration of power observed in past deployments. In the following two chapters, we turn away from democratic innovations proper, and instead investigate new approaches to a well-established element of democracy, namely, the apportionment of legislative seats.

### Chapter 6: Avoiding the Concentration of Power in Liquid Democracy

[Chapter 6](#) studies liquid democracy, another democratic innovation seen as a possible path for giving constituents more say in politics [[BZ16](#)]. As in direct democracy, liquid democracy allows constituents to vote on fine-grained issues such as individual laws. Since constituents have limited expertise on policy and limited time to devote to politics, liquid democracy alternatively allows constituents to *delegate* their vote to any other constituent, whose vote then has the weight of multiple individuals. These delegations are *transitive* (i.e., a delegate can delegate their vote and all votes delegated to them to another constituent) and revocable at any point. The probably largest deployment of liquid democracy took place in the Pirate Party in Germany [[BP14](#); [KKH+15](#)]. A major concern

[[BZ16](#)] Blum and Zuber (2016): *Liquid Democracy*.

[[BP14](#)] Bullwinkel and Probst (2014): *Innerparteiliche Willensbildung und Entscheidungsprozesse durch digitale Partizipation*.  
 [[KKH+15](#)] Kling et al. (2015): *Voting Behaviour and Power in Online Democracy*.

in this deployment was that some individuals, so-called *super-voters*, amassed enormous weight through transitive delegations.

In this chapter, we investigate whether the emergence of super-voters could be prevented if delegators nominated multiple potential delegates that they are indifferent between and if a centralized algorithm could choose the delegate among these options with the goal of reducing the weight of the heaviest super-voter. In a random graph model, we prove that when each delegator nominates two potential delegates rather than one, this approach leads to a doubly exponential reduction in the weight of the largest super-voter, with high probability. This improvement is related to the *power of two choices* phenomenon documented in load balancing [ABKU94]. Furthermore, we find that the clear improvement going from one delegate to two potential delegates persists in random graphs generated with preferential attachment, in which some nodes are much more frequently chosen as delegates than others.

## Chapter 7: Party-List Apportionment with Approval Votes

The following chapter develops procedures for allocating the seats of a legislature to political parties. Specifically, our work applies to electoral systems using *party-list proportional representation*,<sup>8</sup> which means that each party receives a share of seats proportional to its share of the popular vote.

Whereas voters can currently only vote for a single party, we extend party-list proportional representation to *approval ballots*, which means that voters can vote for any subset of the parties. These approval ballots allow voters to communicate their preferences with larger expressiveness, which in turn enables the apportionment procedure to promote parties commonly approved by different voters. In contrast to single-party ballots, how to define proportional representation and how to apportion seats is not well established for approval ballots. We adapt axioms from the related literature on multi-winner elections and design apportionment methods that satisfy combinations of axioms that have remained elusive in that literature. Specifically, we propose an apportionment method satisfying *core stability* [ABC+17] and one that satisfies both *extended justified representation* [ABC+17] and *house monotonicity* [BCo8].

## Chapter 8: Monotone Randomized Apportionment

We then study another extension of apportionment, which is not only applicable to the apportionment of seats to *parties* but also to the apportionment of seats to *states*, as is done in the US for example. This kind of apportionment has a long and fascinating history, which has been marked by the challenge of simultaneously guaranteeing desirable axioms, most prominently *quota*, *house monotonicity*, and *population monotonicity* [BY01]. Grimmett [Grio4] suggests the use of randomness in apportionment, which allows to apportion in a way that is perfectly proportional (ex ante, i.e., in expectation) while satisfying quota.

To help choose between the many randomized apportionment methods satisfying Grimmett's two axioms, we study if apportionment methods can additionally satisfy versions of house monotonicity and population monotonicity that are

[ABKU94] Azar et al. (1994): *Balanced Allocations*.

8: Being used in 85 countries, this electoral system is the most widely used globally [ACE22]. Its most frequent alternative is plurality ("first past the post"), in which candidates compete in districts and, in each district, the candidate winning the largest vote share receives a seat.

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

[BCo8] Barberà and Coelho (2008): *How to Choose a Non-Controversial List with  $k$  Names*.

[BY01] Balinski and Young (2001): *Fair Representation*.

[Grio4] Grimmett (2004): *Stochastic Apportionment*.

suitably generalized to the randomized setting. Indeed, we design a randomized apportionment method that satisfies ex-ante proportionality, quota, and house monotonicity. Satisfying both quota and population monotonicity is not possible, but we give an apportionment method that satisfies ex-ante proportionality and population monotonicity (without quota). Our exploration of these questions leads us to an interesting generalization of randomized dependent rounding [GKPS06], and additionally provides new insights into the mathematical structure of deterministic apportionment methods. Our generalized rounding procedure can be used to resolve multiple shortcomings of a suggested reform of the European Commission proposed by Buchstein and Hein [BH09].

### Part III: Refugee Resettlement

Like the previous two parts, the third part of this thesis is motivated by an urgent need for political action. As of 2022, a record 100 million people are forcibly displaced worldwide, more than twice the number from 2012 [UNH22, p. 7]. Among the 27 million people with refugee status,<sup>9</sup> the *United Nations High Commissioner for Refugees (UNHCR)* identifies 1.5 million as *in need of resettlement* [UNH21]. This designation means that these refugees need to be permanently relocated from their current country of asylum into a third country, for example because their safety is at risk or because of trauma caused by violence or torture. Even for this highly vulnerable population, the response of the international community has fallen far short of the need; even before the pandemic, only around 100 000 refugees were resettled per year [UNHnd].

Despite this similarity in motivation, the third part stands out from the others in two ways: First, the first two parts aim to change democratic structures, hoping that doing so will indirectly enable society to overcome urgent problems. By contrast, the third part engages with a concrete problem, refugee resettlement, and aims for marginal improvements rather than systematic change.<sup>10</sup> Second, whereas the first two parts develop algorithms with theoretical guarantees, the third develops a heuristic algorithm — albeit inspired by theoretical arguments — and studies it empirically.

#### Chapter 9: Online Refugee Placement

In the only chapter of this part, we study a problem faced by resettlement agencies in the US. Each week, such a resettlement agency is assigned a batch of refugees by the federal government, and the agency must place these refugees in its local affiliates, while respecting the affiliates' yearly capacities. The main objective of the resettlement agency is to help many refugees find employment, which depends on which affiliate each refugee is placed in. Building up on a previous tool by Ahani et al. [AAM+21] and its predictions of employment chances, we design heuristic algorithms for making these allocation decisions. In contrast to earlier algorithms, ours explicitly take into account that refugees arrive over time (*online*). Our algorithms achieve over 98 percent of the hindsight-optimal employment, compared to under 90 percent of previous myopic approaches. This improvement persists even when we incorporate a vast array of practical features of the refugee resettlement process including indivisible families, batching, and

[GKPS06] Gandhi et al. (2006): *Dependent Rounding and Its Applications to Approximation Algorithms*.

[BH09] Buchstein and Hein (2009): *Randomizing Europe*.

[UNH22] UNHCR (2022): *Global Trends: Forced Displacement in 2021*.

9: This number excludes people displaced within their home country and asylum seekers.

[UNH21] UNHCR (2021): *Projected Global Resettlement Needs 2022*.

[UNHnd] UNHCR (n.d.): *Refugee Data Finder*.

10: While we wholeheartedly agree that “many of our pressing social problems cannot be solved by better allocating our existing sets of outcomes” [Man19] and that changes to immigration policy would yield much larger positive impact, we still believe that the improvements of this work are worthwhile to pursue.

[AAM+21] Ahani et al. (2021): *Placement Optimization in Refugee Resettlement*.

uncertainty with respect to the number of future arrivals. As we describe below, we incorporated our algorithm into the optimization software used by a leading US resettlement agency.

## 1.2 Real-World Impact

Since the research in this thesis is driven by practical problems, I spent substantial effort to translate our technical results into impact in the real world. In particular, several of the algorithms designed in this thesis have been implemented in software tools that are now being used in practice.

### A Fair Selection Algorithm for Citizens' Assemblies

Over the course of my research on citizens' assemblies, I have built connections with practitioners in more than a dozen organizations. The conversations with these practitioners have sparked research questions, have informed our solution approaches, and have helped practitioners understand and apply the results of our work.

As perhaps the most impactful output of this thesis, I have developed an efficient implementation of the LEXIMIN algorithm described in [Section 2.3](#), which allows practitioners to sample a representative panel in a way that gives pool members maximally fair chances of being selected.<sup>11</sup> Brett Hennig, co-founder and co-director of the Sortition Foundation, which has adopted LEXIMIN, describes the value of our algorithm to his organization as follows:

“Putting the algorithm we use to select representative samples of people for citizens' assemblies on a solid theoretical basis, and proving that these algorithms are the fairest possible algorithms we can use, has been invaluable to the Sortition Foundation as we strive to increase the transparency and legitimacy of deliberative democracy processes all over the world. It is absolutely essential to get this crucial step in the recruitment process right as we move towards giving such assemblies real political power.”

**StratifySelect.** My implementation of LEXIMIN is now the default algorithm in the open-source software *StratifySelect* [[HG21](#)], which has been used to select the assemblies of multiple NGOs since before our work. Though we do not have systematic insight into who uses this software, the Sortition Foundation has used StratifySelect with our algorithm since March 2020 and selects dozens of citizens' assemblies per year. Among the assemblies selected by the Sortition Foundation with our algorithm are Scotland's Climate Assembly [[SCO21](#)], the NHS Test and Trace Public Advisory Group in the United Kingdom [[UKH22](#)], and the Citizens' Council in East Belgium, which is the oldest permanent citizens' assembly in the world [[NR19](#)].

<sup>11</sup>: I also created implementations for other algorithms in this framework, maximizing fairness measures based on egalitarian welfare and Nash welfare ([Section 2.2](#)).

[[HG21](#)] Hennig and Gölz (2021): *StratifySelect*.

[[SCO21](#)] Scotland's Climate Assembly (2021): *Recommendations for Action*.

[[UKH22](#)] UKHSA (2022): *Test and Trace Public Advisory Group Report*.

[[NR19](#)] Niessen and Reuchamps (2019): *Designing a Permanent Deliberative Citizens' Assembly*.

**Panelot.** To circumvent the technical hurdle of installing StratifySelect, Gili Rusak and I developed and maintain the website Panelot ([panelot.org](https://panelot.org)) [GR20]. On this website, anybody can select citizens’ assemblies through LEXIMIN for free.

For privacy reasons, we originally did not retain any data about access to this website. Nevertheless, we know that the site is actively used: we have anecdotally heard about the website being used to select citizens’ assemblies, and, in the 30 days after we enabled basic logging in June 2022, users computed 10 LEXIMIN lotteries.<sup>12</sup>



[GR20] Gözl and Rusak (2020): *Panelot*.

12: Based on pool and panel sizes, we believe that these lotteries selected 5 distinct citizens’ assemblies, and that the remaining 5 lotteries were used to replace members of an existing panel who dropped out. The website was used an additional 5 times with infeasible quotas; it seems that practitioners use Panelot to find possible relaxations of infeasible quotas.

**Figure 1.2:** Screenshot of the web interface during the democratic lottery for the Michigan assembly. For an explanation of the interface and process, see [Figure 2.3](#).

**Democratic Lottery with Physical Randomness.** As described in [Section 2.5](#), I actively supported the nonprofit organization *of by for \** in selecting the panel for their citizens’ assembly on COVID-19 in Michigan [Cit20]. To make the selection process more trustworthy and to allow observers to see the fairness of selection probabilities, the panel was selected using lottery balls on live stream [Ofb20]. I helped *of by for \** design this process, optimized a “rounded” lottery over panels, and together with Stephen Braitsch, created the website that allowed pool members to watch the live stream and track their chances of being selected in real time ([Figure 1.2](#)). We received very positive feedback from the organizers, and pool members found the lottery exciting. One participant recounted her experience of being selected as follows (edited for readability):

“I was very surprised when I made the lotto. I was showing my son the democratic drawing. He was sitting next to me in the chair when the balls were going up. ‘Mom you won!’ He was really happy.”

I similarly supported the selection of the Washington Climate Assembly [Was21].

### Improved Mixing of Deliberation Groups

In [Chapter 5](#), we find that certain greedy algorithms can schedule deliberation groups in a way that allows many more pairs of panel members to interact than the algorithm currently used in practice. With Philipp Verpoort from the Sortition Foundation, Rose Hong, and Jake Barrett, I am currently integrating this algorithm into the tool GROUPSELECT [Ver22], which previously hosted the baseline algorithm. This implementation will include a novel user interface that supports practitioners in trading off group representativeness and breadth of pairwise interactions, and is based on the experience of using the tool in practice.

[Cit20] Citizens’ Panel on COVID-19 (2020): *Final report*.

[Ofb20] Of By For \* (2020): *Democratic lottery — the citizens’ panel on COVID-19*.

[Was21] Washington Climate Assembly (2021): *Washingtonians Finding Solutions Together*.

[Ver22] Verpoort (2022): *GroupSelect*.

## Online Refugee Placement

In [Chapter 9](#), we design a data-based online algorithm for matching resettled refugees to the local affiliates of a resettlement agency. In the Summer of 2021, Narges Ahani and I implemented this algorithm in the software tool Annie™ MOORE used by the US resettlement agency HIAS. As we describe in [Chapter 9](#), the algorithm was designed with close attention to the practical problems encountered by HIAS. Moreover, our user interface augments the interface of the original version of Annie™ in a way that exposes key tradeoffs in case HIAS staff need to manually override the matching ([Section 9.8](#)). Our improvements to the software have been well-received by HIAS leadership:

“Annie™ 2.0 is a game-changer for our pre-arrivals processes, allowing us to plan and optimize our pre-arrival strategy a year rather than a week ahead.”

## 1.3 Bibliographic Notes

The research contained in this thesis is based on joint work with different co-authors, as described below. In all works included, I was the primary contributor or one of multiple primary contributors. On all publications, authors appear in alphabetical order.

[Chapter 2](#) is based on joint work with Bailey Flanigan, Anupam Gupta, Brett Hennig, and Ariel Procaccia [[FGG+21](#)]. [Chapter 3](#) is based on joint work with Bailey Flanigan, Anupam Gupta, and Ariel Procaccia [[FGGP20](#)]. [Chapter 4](#) is based on joint work with Gerdus Benadè and Ariel Procaccia [[BGP19](#)]. [Chapter 5](#) is based on joint work with Jake Barrett, Kobi Gal, Rose Hong, and Ariel Procaccia [[BGG+22](#)]. [Chapter 6](#) is based on joint work with Anson Kahng, Simon Mackenzie, and Ariel Procaccia [[GKMP18](#)]. [Chapter 7](#) is based on joint work with Markus Brill, Dominik Peters, Ulrike Schmidt-Kraepelin, and Kai Wilker [[BGP+22](#)]. [Chapter 8](#) is based on joint work with Dominik Peters and Ariel Procaccia [[GPP22](#)]. [Chapter 9](#) is based on joint work with Narges Ahani, Ariel Procaccia, Alexander Teytelboym, and Andrew Trapp [[AGP+21](#)].

To keep this thesis more thematically coherent, a substantial part of my doctoral research is omitted:

- ▶ Work on refugee matching from a more theoretical angle [[GP19](#)].
- ▶ Work on mechanism design for kidney exchange using smoothed analysis [[BG21](#)].
- ▶ Work on fair item allocation, in a random model [[BG22](#)] and using smoothed analysis [[BFGP22](#)].
- ▶ Work combining fair machine learning with axioms from fair division [[GKP19](#)].

[[FGG+21](#)] Flanigan et al. (2021): *Fair Algorithms for Selecting Citizens’ Assemblies*.

[[FGGP20](#)] Flanigan et al. (2020): *Neutralizing Self-Selection Bias in Sampling for Sortition*.

[[BGP19](#)] Benadè et al. (2019): *No Stratification Without Representation*.

[[BGG+22](#)] Barrett et al. (2022): *Now we’re talking*.

[[GKMP18](#)] Gözl et al. (2018): *The Fluid Mechanics of Liquid Democracy*.

[[BGP+22](#)] Brill et al. (2022): *Approval-Based Apportionment*.

[[GPP22](#)] Gözl et al. (2022): *In This Apportionment Lottery, the House Always Wins*.

[[AGP+21](#)] Ahani et al. (2021): *Dynamic Placement in Refugee Resettlement*.

[[GP19](#)] Gözl and Procaccia (2019): *Migration as Submodular Optimization*.

[[BG21](#)] Blum and Gözl (2021): *Incentive-Compatible Kidney Exchange in a Slightly Semi-Random Model*.

[[BG22](#)] Bai and Gözl (2022): *Envy-Free and Pareto-Optimal Allocations for Agents with Asymmetric Random Valuations*.

[[BFGP22](#)] Bai et al. (2022): *Fair Allocations for Smoothed Utilities*.

[[GKP19](#)] Gözl et al. (2019): *Paradoxes in Fair Machine Learning*.

# PROPOSALS FOR CITIZENS' ASSEMBLIES

# Fair Algorithms for Selecting Citizens' Assemblies

# 2

## 2.1 Introduction

In representative democracies, political representatives are usually selected by election. However, over the last 35 years, an alternative selection method has been gaining traction among political scientists [CM99; Dah90; DD10] and practitioners [FL05; MM18; OEC20; Par21]: *sortition*, the random selection of representatives from the population. The chosen representatives form a panel, commonly called a *citizens' assembly*, which convenes to deliberate on a policy question. Citizens' assemblies are now being administered by around 50 organizations in over 25 countries [Dem19], and just one of these organizations, the Sortition Foundation in the UK, recruited 29 panels in 2020. While many citizens' assemblies are initiated by civil-society organizations, they are also increasingly being commissioned by public authorities on municipal, regional, national, and supranational levels [OEC20]. In fact, since 2019, multiple regional parliaments in Belgium and the Council of Paris have internally established permanent sortition bodies [NR19; OEC21]. Citizens' assemblies' growing utilization by governments is giving their decisions a more direct path to policy impact. For example, two recent citizens' assemblies commissioned by Ireland's national legislature led to the legalization of same-sex marriage and abortion [Iri19a].

Ideally, a citizens' assembly selected via sortition acts as a microcosm of society: its participants are representative of the population, and thus its deliberation simulates the entire population convening "under conditions where it can really consider competing arguments and get its questions answered from different points of view" [Fis18]. Whether this goal is realized in practice, however, depends on exactly how assembly members are chosen.

As we sketched in the introduction, panel selection is generally done in three stages: first, thousands of randomly chosen constituents are invited to participate. Second, a subset of the invited constituents opt into a *pool* of volunteers. Third, a panel of pre-specified size is randomly chosen from the pool via some fixed procedure, which we call a *selection algorithm*. As the final and most complex component of the selection process, the selection algorithm has great power in deciding who will be chosen to represent the population. In this chapter, we introduce selection algorithms that preserve the key desirable property of existing algorithms, while also more fairly distributing the sought-after opportunity [Cit20; Jac19; NEK+10; Ofb20] of being a representative.

Of the several selection algorithms used in practice prior to this work,<sup>1</sup> all aim to satisfy one particular property: *descriptive representation*, the idea that the panel should reflect the composition of the population [Fis18]. Unfortunately, the pool from which the panel is chosen tends to be far from representative. Specifically, it tends to overrepresent groups whose members are more likely to accept an invitation to participate, such as high educational attainment. To ensure descriptive representation despite the biases of the pool, selection algorithms

2.1	Introduction	13
2.1.1	Related Work	15
2.2	Algorithmic Framework	15
2.2.1	Definitions	15
2.2.2	Formulating the Optimization Task	15
2.2.3	Solving the Optimization Task	16
2.3	Deployable Algorithm	18
2.4	Effect of Adopting LEXIMIN over LEGACY	19
2.5	Discussion	20

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Dah90] Dahl (1990): *After the Revolution*.

[DD10] Delannoi and Dowlen (2010): *Sortition*.

[FL05] Fishkin and Luskin (2005): *Experimenting with a Democratic Ideal*.

[MM18] McKay and MacLeod (2018): *MASS LBP and Long-Form Deliberation in Canada*.

[OEC20] OECD (2020): *Innovative Citizen Participation and New Democratic Institutions*.

[Par21] Various Authors (2021): *Participedia*.

[Dem19] Democracy R&D (2019): *About the network*.

[NR19] Niessen and Reuchamps (2019): *Designing a Permanent Deliberative Citizens' Assembly*.

[OEC21] OECD (2021): *Eight Ways to Institutionalise Deliberative Democracy*.

[Iri19a] Irish Citizens' Assembly Project (2019): *Irish citizens' assembly (2019–2020)*.

[Fis18] Fishkin (2018): *Democracy When the People Are Thinking*.

[Cit20] Citizens' Panel on COVID-19 (2020): *Final report*.

[Jac19] Jacquet (2019): *The Role and the Future of Deliberative Mini-Publics*.

[NEK+10] Neblo et al. (2010): *Who Wants to Deliberate—and Why?*

[Ofb20] Of By For \* (2020): *Democratic lottery — the citizens' panel on COVID-19*.

1: See [supplementary information 12](#) of the full version.

[Fis18] Fishkin (2018): *Democracy When the People Are Thinking*.

require that the panels they output satisfy upper and lower *quotas* on a set of specified features, which are roughly proportional to each feature's population rate (e.g. quotas might require that a 40-person panel contain between 20 and 21 women). These quotas are generally imposed on feature categories delineated by gender, age, education level, and other attributes relevant to the policy issue at hand. We note that quota constraints of this form are more general than those achievable via *stratified sampling*, a common technique for drawing representative samples.<sup>2</sup>

The selection algorithms pre-dating this work focused solely on satisfying quotas, leaving unaddressed a second property that is also central to sortition: that all individuals should have an *equal chance* of being chosen for the panel. Several political theorists present equality of selection probabilities as a central advantage of sortition, stressing its role in promoting the ideals such as *equality of opportunity* [CM99; Par11], *democratic equality* [Fis09; Fis18; Par11; Sto16], and *allocative justice* [Sto11; Sto16]. In fact, Engelstad, who introduced an influential model of sortition's benefits, argues that this form of equality constitutes "The strongest normative argument in favour of sortition" [Eng89]. (In [Appendix A.1](#), we explore desiderata from political theory in more detail.) In addition to political theorists, major practitioner groups have also advocated for equal selection probabilities [AB18; MAS17]. However, they face the fundamental hurdle that, in practice, the quotas almost always necessitate selecting people with somewhat unequal probabilities, as individuals from groups that are underrepresented in the pool must be chosen with disproportionately high probabilities to satisfy the quotas.

Though it is generally impossible to achieve *perfectly* equal probabilities, the reasons to strive for equality also motivate a more gradual version of this goal: making probabilities as equal as possible, subject to the quotas. We refer to this goal as *maximal fairness*. We find that our benchmark, a selection algorithm representing the previous state of the art, falls far short of this goal, giving volunteers drastically unequal probabilities across several real-world instances. This algorithm even consistently selects certain types of volunteers with *near-zero* probability, thereby excluding them in practice from the chance to serve. We further show that, in these instances, it is possible to give all volunteers probability well above zero while still satisfying the quotas, demonstrating that the level of inequality produced by the benchmark is avoidable.

In this chapter, we close the gaps we have identified, both in theory and in practice. We first introduce not just one selection algorithm that achieves maximal fairness, but a more general (I) [algorithmic framework](#) for producing such algorithms. Motivated by the multitude of possible ways to quantify the fairness of an allocation of selection probabilities, our framework gives a maximally fair selection algorithm for any measure of fairness with a certain functional form. Notably, such measures include the most prominent from the literature on *fair division* [BT96; Mou04], and we show that these well-established metrics can be applied to our setting by casting the problem of assigning selection probabilities as one of fair resource allocation.<sup>3</sup> Then, to bring this innovation into practice, we implement a (II) [deployable selection algorithm](#), which is maximally fair according to one specific measure of fairness. We evaluate this algorithm and find that it is substantially fairer than the benchmark on several real-world datasets and by multiple fairness measures. Our algorithm is now in use by a growing

2: See [supplementary information 3](#) of the full version and [Chapter 4](#).

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Par11] Parker (2011): *Randomness and Legitimacy in Selecting Democratic Representatives*.

[Fis09] Fishkin (2009): *When the People Speak*.

[Fis18] Fishkin (2018): *Democracy When the People Are Thinking*.

[Sto16] Stone (2016): *Sortition, Voting, and Democratic Equality*.

[Sto11] Stone (2011): *The Luck of the Draw*.

[Eng89] Engelstad (1989): *The Assignment of Political Office by Lot*.

[AB18] Allianz Vielfältige Demokratie and Bertelsmann Foundation (2018): *Citizens' participation using sortition*.

[MAS17] MASS LBP (2017): *How to Run a Civic Lottery*.

[BT96] Brams and Taylor (1996): *Fair Division*.

[Mou04] Moulin (2004): *Fair Division and Collective Welfare*.

3: See [supplementary information 9](#) of the full version.

number of sortition organizations around the world, making it one of only a few [BCKO17; Faind; GP14; Sun14] deployed applications of fair division.

### 2.1.1 Related Work

Besides the work in this thesis, three computer science papers study selection algorithms for citizens’ assemblies. The first paper, by Do et al. [DALU21], investigates selection algorithms in an “online” model. In this model, the panel is not selected from a pool of volunteers; instead, volunteers present themselves to the algorithm sequentially, and the algorithm must immediately and irrevocably decide whether the current volunteer should be included in the panel.<sup>4</sup> In this model, Do et al. study three selection algorithms and characterize the representativeness achieved by the algorithms as a function of the number of volunteers (equality is not considered). In the second paper, Flanigan et al. [FKP21] design postprocessing algorithms that allow to apply the selection algorithms from this chapter in lotteries with physical randomness. We describe these lotteries in Sections 1.2 and 2.5, and discuss their work in Section 2.5. Finally, a third paper by Meir et al. [MST21] studies a social choice model in which society must make a sequence of binary decisions, while only eliciting the preferences of a fixed number  $k$  of agents. In this context, Meir et al. find that “ $k$ -sortition”, i.e., uniformly selecting  $k$  agents from the population and holding majority votes among the  $k$  agents, is a good approximation of a majority vote among the entire population, and is optimal when many decisions are made with worst-case preferences.

[BCKO17] Budish et al. (2017): *Course Match*.

[Faind] Fair Outcomes (n.d.): *Fair Proposals*.

[GP14] Goldman and Procaccia (2014): *Spliddit*.

[Sun14] Sun (2014): *Divide your rent fairly*.

[DALU21] Do et al. (2021): *Online Selection of Diverse Committees*.

4: This model does not apply when constituents are simultaneously invited by mail, which is the case in most organizations we talked to. Instead, it seems like an accurate model for recruitment by phone, which is also used for citizens’ assemblies.

[FKP21] Flanigan et al. (2021): *Fair Sortition Made Transparent*.

[MST21] Meir et al. (2021): *Representative Committees of Peers*.

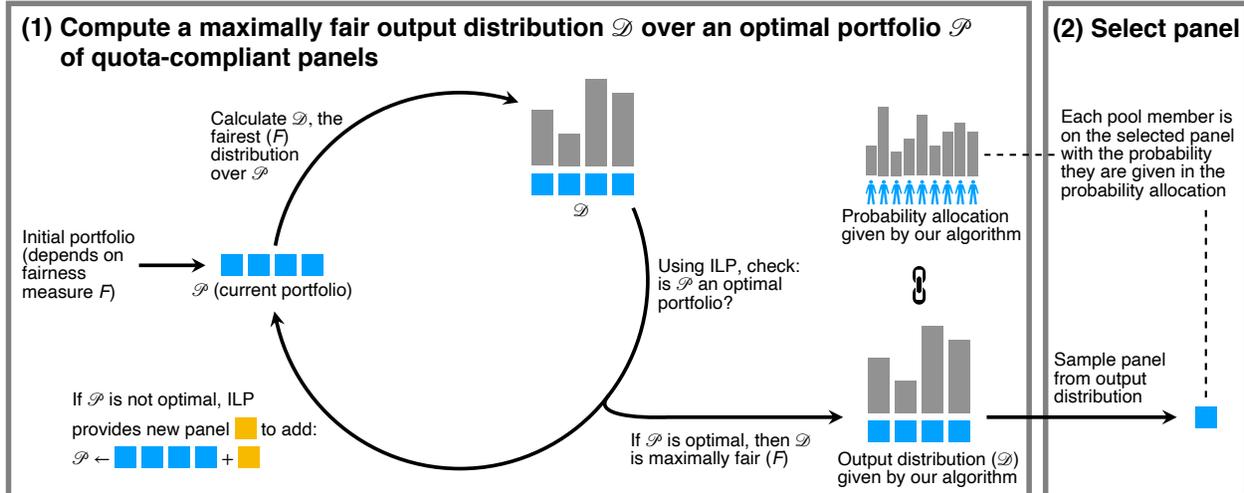
## 2.2 Contribution I: Algorithmic Framework

### 2.2.1 Definitions

We begin by introducing necessary terminology. We refer to the input to a selection algorithm — a pool of size  $n$ , a set of quotas, and the desired panel size  $k$  — as an *instance* of the panel selection problem. Given an instance, a selection algorithm randomly selects a *panel*, which is a quota-compliant set of  $k$  pool members. We define the algorithm’s *output distribution* on an instance as the distribution specifying the probabilities with which the algorithm outputs each possible panel. Then, a pool member’s *selection probability* is the probability that they are on a panel randomly drawn from the output distribution. We refer to the mapping from pool members to their selection probabilities as the *probability allocation*, which we aim to make as fair as possible. Finally, a *fairness measure* is a function that maps a probability allocation to a fairness “score” (e.g. the geometric mean of probabilities, where higher is fairer). An algorithm is called *optimal* with respect to a fairness measure if, on any instance, the fairness of the algorithm’s probability allocation is at least as high as that of any other algorithm.

### 2.2.2 Formulating the Optimization Task

To inform our approach, we first analyze the algorithms pre-dating ours. Those we have seen in use all have the same high-level structure: they select individuals



**Figure 2.1:** The steps of the algorithm optimizing the fairness measure  $F$ . The left-hand panel shows the implementation of step (1): constructing a maximally fair output distribution over panels (denoted by white boxes), which is done by iteratively building an optimal portfolio of panels and computing the fairest distribution over that portfolio. The right-hand panel shows step (2): sampling the distribution to select a final panel.

for the panel one-by-one, in each step randomly choosing whom to add next from among those who, according to a myopic heuristic, seem unlikely to produce a quota violation later. Since finding a quota-compliant panel is an algorithmically hard problem,<sup>5</sup> it is already an achievement that such simple algorithms find *any* panel in most practical instances. Due to their focus on finding any panel at all, however, these algorithms do not tightly control *which* panel they output, or more precisely, their output distribution (the probabilities with which they output different panels). Since an algorithm's output distribution directly determines its probability allocation, existing algorithms' probability allocations are also uncontrolled, leaving room for them to be highly unfair.

<sup>5</sup>: See [supplementary information 6](#) of the full version.

While existing algorithms allow their output distribution to arise *implicitly* from a sequence of myopic steps, we design the algorithms in our framework to (1) *explicitly* compute their own output distribution, and then (2) sample from that distribution to select the final panel (Figure 2.1). Crucially, the output distribution found in the first step is *maximally fair* to pool members, making our algorithms optimal. To see why, note that the behavior of *any* selection algorithm on a given instance is described by some output distribution; thus, since our algorithm finds the *fairest possible* output distribution, it is always at least as fair as any other algorithm.

Since step (2) of our selection algorithm is simply a random draw, we have reduced the problem of finding an optimal selection algorithm to the optimization problem in step (1) — finding a maximally fair distribution over panels. Now, to fully specify our algorithm, it remains only to solve this optimization problem.

### 2.2.3 Solving the Optimization Task

A priori, it would seem that computing a maximally fair distribution might require constructing *all possible* panels, since achieving optimal fairness might necessitate assigning non-zero probability to all of them. Such an approach

would be impracticable, however, as the number of panels in most instances is intractably large. Fortunately, since we measure fairness according to only individual selection probabilities, there must exist an *optimal portfolio* — a set of panels over which there exists a maximally fair distribution — containing few panels by Carathéodory's theorem:

**Proposition 2.1** *Fix an arbitrary instance and a fairness measure  $F$  for this instance. If there exists any maximally fair distribution over panels for  $F$ , there exists a maximally fair output distribution whose support includes at most  $n + 1$  panels.*

*Proof.* Consider the hypercube  $[0,1]^n$ , and associate each dimension with one pool member. A panel  $P$  can be embedded into this space by its characteristic vector  $\vec{v}_P \in \{0,1\}^n$ , whose  $i$ th component is one exactly if pool member  $i$  is contained in  $P$ .

Fix a maximally fair output distribution, let  $\mathcal{P}$  denote its support, and let  $\{\lambda_P\}_{P \in \mathcal{P}}$  denote its probability mass function. Note that

$$\vec{p} := \sum_{P \in \mathcal{P}} \lambda_P \vec{v}_P$$

is a probability allocation maximizing  $F$ , and that it is a convex combination of the  $\{\vec{v}_P\}_{P \in \mathcal{P}}$ . By Carathéodory's theorem, there is a subset  $\mathcal{P}' \subseteq \mathcal{P}$  of size at most  $n + 1$  such that  $\vec{p}$  still lies in the convex hull of this smaller set. Thus, there are nonnegative real numbers  $\{\lambda'_P\}_{P \in \mathcal{P}'}$  adding up to one such that

$$\vec{p} = \sum_{P \in \mathcal{P}'} \lambda'_P \vec{v}_P.$$

These  $\lambda'_P$  form the probability mass function of a distribution over at most  $n + 1$  panels, which has the same probability allocation  $\vec{p}$  as the original maximally fair distribution, which implies that the new distribution is also maximally fair for  $F$ .  $\square$

This result brings a practical algorithm within reach, and shapes the goal of our algorithm: to find an optimal portfolio while constructing as few panels as possible.

We accomplish this goal using an algorithmic technique called *column generation*, where, in our case, the “columns” being generated correspond to panels. A more in-depth discussion and formal description of this algorithm, as well as proofs of correctness, can be found in [supplementary information 8](#) of the full version. As shown in [Figure 2.1](#), our algorithms find an optimal portfolio by iteratively adding panels to a portfolio  $\mathcal{P}$ , in each iteration alternating between two subtasks: (i) finding the optimal distribution  $\mathcal{D}$  over only the panels currently in  $\mathcal{P}$  and (ii) adding a panel to  $\mathcal{P}$  that, based on the gradient of the fairness measure, will move the portfolio furthest towards optimality. This second subtask makes use of *integer linear programming*, which we use to generate quota-compliant panels despite the theoretical hardness of the problem. Eventually, the panel with the most promising gradient will already be in  $\mathcal{P}$ , in which case  $\mathcal{P}$  is provably

optimal and  $\mathcal{S}$  must be a maximally fair distribution. In practice, we observe that this procedure terminates after few iterations.

Our techniques extend column generation methods that are typically applied to linear programs, allowing them to be used to solve a large set of convex programs.<sup>6</sup> This extension allows our framework to be used with a wide range of fairness measures — essentially any for which the fairest distribution over a portfolio can be found via convex programming. Supported measures include those most prominent in the fair division literature: egalitarian welfare [End10], Nash welfare [Mou04], Gini inequality [End13; LP10], and the Atkinson indices [End13; SDE17]. Our algorithmic approach also has the benefit of easily extending to organization-specific constraints beyond quotas; for example, practitioners can prevent multiple members of the same household from appearing on the same panel. Due to its generality, our framework even applies to domains outside of sortition, including the allocation of classrooms to charter schools [KPS18] and kidney exchange [RSÜ05].<sup>7</sup>

## 2.3 Contribution II: Deployable Selection Algorithm

To bring fair panel selection into practice, we develop an efficient implementation of one specific selection algorithm, which we call LEXIMIN (formally defined in [supplementary information 10](#) of the full version). LEXIMIN optimizes the well-established fairness measure *leximin* [BM04; KPS18; Mou04], a fairness measure that is sensitive to the very lowest selection probabilities. In particular, *leximin* is optimized by maximizing the lowest selection probability, then breaking ties between solutions in favor of probability allocations with highest second-lowest probability, and so on. This choice of fairness measure is motivated by the fact that, as we show in this section and in [supplementary information 13](#) of the full version, LEGACY gives some pool members a near-zero probability when much more equal probabilities are possible. This type of unfairness is especially pressing because, if it consistently impacted pool members with certain combinations of features, these individuals and their distinct perspectives would be “systematically excluded from participation” [Smio9], which runs counter to a key promise of random selection.

To increase the accessibility of LEXIMIN, we made its implementation available through an existing open-source panel selection tool [HG21] and on [Panelot](#) [GR20], a website where anyone can run the algorithm without installation. LEXIMIN has since been deployed by several organizations, including *Cascadia* (US), the *Danish Board of Technology* (Denmark), *Nexus* (Germany), *of by for* \* (US), *Particitiz* (Belgium), and the *Sortition Foundation* (UK). As of July 2021, the Sortition Foundation alone had already used LEXIMIN to select more than 40 panels.

We measure the impact of adopting LEXIMIN over pre-existing algorithms by comparing its fairness to that of a benchmark, LEGACY (described in [supplementary information 11](#) of the full version), the algorithm used by the Sortition Foundation prior to their adoption of LEXIMIN. We choose LEGACY as a benchmark because it was widely used prior to this work, it is similar to several other selection algorithms used in practice,<sup>8</sup> and it is the only existing algorithm we

6: See [supplementary information 8.1](#) of the full version.

[End10] Endriss (2010): *Lecture Notes on Fair Division*.

[Mou04] Moulin (2004): *Fair Division and Collective Welfare*.

[End13] Endriss (2013): *Reduction of Economic Inequality in Combinatorial Domains*.

[LP10] Lesca and Perny (2010): *LP Solvable Models for Multiagent Fair Allocation Problems*.

[SDE17] Schneckeburger et al. (2017): *The Atkinson Inequality Index in Multiagent Resource Allocation*.

[KPS18] Kurokawa et al. (2018): *Leximin Allocations in the Real World*.

[RSÜ05] Roth et al. (2005): *Pairwise Kidney Exchange*.

7: See [supplementary information 8.2](#) of the full version.

[BM04] Bogomolnaia and Moulin (2004): *Random Matching under Dichotomous Preferences*.

[Smio9] Smith (2009): *Democratic Innovations*.

[HG21] Hennig and Gözl (2021): *StratifySelect*.

[GR20] Gözl and Rusak (2020): *Panelot*.

8: See [supplementary information 13](#) of the full version.

Table 2.1: List of instances used in our experiments. At the request of practitioners, topics, dates, and locations of the panels are not identified.

instance <sup>a</sup>	pool size $n$	panel size $k$	number of quota categories	Mean selection probability $k/n$	LEGACY min. probability (sampled) <sup>b</sup>	LEXIMIN min. probability (exact)	LEXIMIN running time
sf(a)	312	35	6	11.2%	$\leq 0.32\%$	6.7%	20 s
sf(b)	250	20	6	8.0%	$\leq 0.17\%$	4.0%	9 s
sf(c)	161	44	7	27.3%	$\leq 0.15\%$	8.6%	6 s
sf(d)	404	40	6	9.9%	$\leq 0.11\%$	4.7%	46 s
sf(e)	1727	110	7	6.4%	$\leq 0.03\%$	2.6%	67 min
cca	825	75	4	9.1%	$\leq 0.03\%$	2.4%	7 min
hd	239	30	7	12.6%	$\leq 0.09\%$	5.1%	37 s
mass	70	24	5	34.3%	$\leq 14.9\%$	20.0%	1 s
nexus	342	170	5	49.7%	$\leq 2.24\%$	32.5%	1 min
obf	321	30	8	9.3%	$\leq 0.03\%$	4.7%	3 min

<sup>a</sup> For the instances we study, panels were recruited by the following organisations. sf(a–e): Sortition Foundation; cca: Center for Climate Assemblies; hd: Healthy Democracy; mass: MASS LBP; nexus: Nexus; obf: of by for \*.

<sup>b</sup> 99% confidence bound, see methods section “Statistics” of the full version.

found that was fully specified by an official implementation. We compare the LEXIMIN and LEGACY on ten datasets from real-world panels, with respect to several fairness measures including the minimum probability (Table 2.1), the Gini coefficient, and the geometric mean. In this analysis, we find that LEXIMIN is fairer on all instances we examine, and substantially so in nine out of ten.

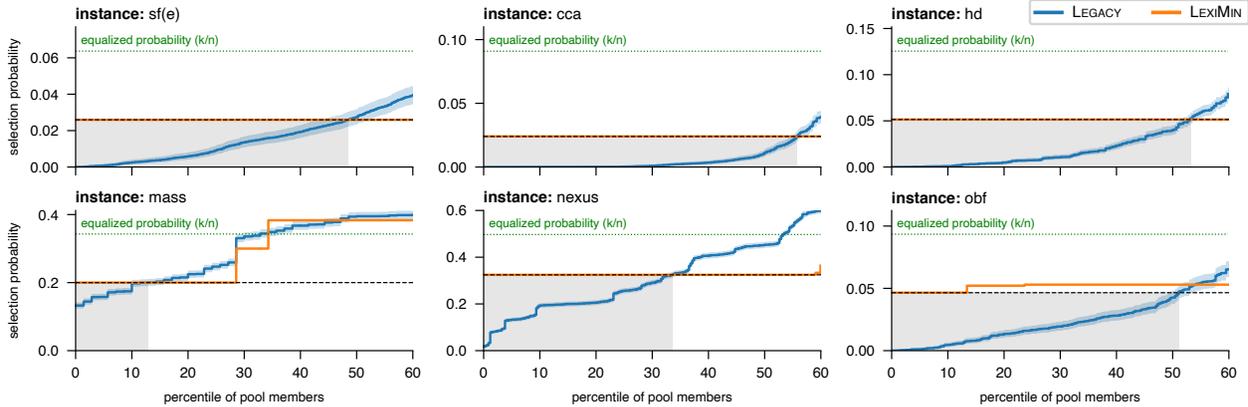
## 2.4 Effect of Adopting LEXIMIN over LEGACY

We study datasets from ten sortition panels, organized by six different sortition organizations in Europe and North America. As Table 2.1 shows, our instances are diverse in panel size (range: 20–170, median: 37.5) and number of quota categories (range: 4–8). On consumer hardware, the run-time of our algorithm is well within the time available in practice.

Out of concern about low selection probabilities, we first compare the minimum selection probabilities given by LEGACY and LEXIMIN, summarized in the second and third columns from the right in Table 2.1. Strikingly, in all instances except mass (an outlier in that its quotas only mildly restrict the fraction of panels that are feasible), LEGACY chooses some pool members with probability close to zero. In fact, we can identify combinations of features that lead to low selection probabilities *across all instances*,<sup>9</sup> raising the concern that LEGACY may in fact systematically exclude some groups from participation. By contrast, LEXIMIN selects no one nearly so infrequently, with minimum selection probabilities ranging from 26% to 65% (median: 49%) of  $k/n$ , the “ideal” probability individuals would receive in the absence of quotas.

<sup>9</sup>: See methods section “Individuals rarely selected by LEGACY” of the full version.

One might wonder whether this increased minimum probability achieved by LEXIMIN affects only a few pool members most disadvantaged by LEGACY. This is not the case: As shown in Figure 2.2 by the shaded boxes, between 13% and 56% of pool members (median 46%) across instances receive probability from LEGACY lower than the *minimum given to anyone* by LEXIMIN (Table A.2 in



**Figure 2.2:** Selection probabilities given by LEGACY and LEXIMIN to the bottom 60% of pool members on six representative instances, where pool members are ordered in order of increasing probability given by the respective algorithms. Shaded boxes denote the range of pool members whose selection probability given by LEGACY is lower than the minimum probability given by LEXIMIN. LEGACY probabilities are estimated over 10 000 random panels and are indicated with 99% confidence intervals (see methods section “Statistics” of the full version). For corresponding graphs for all other instances and up to the 100th percentile, see Figures A.1 and A.2 in Appendix A.2.

Appendix A.2). Thus, even just the first stage of LEXIMIN, i.e., maximizing the *minimum* probability, provides a sizable section of the pool with more equitable access to the panel.

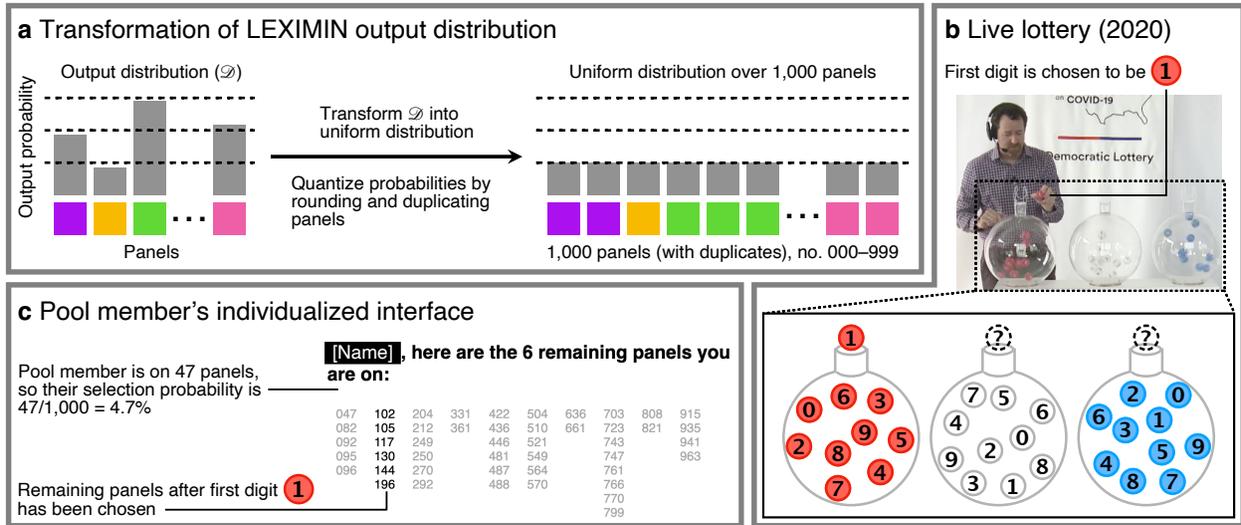
We have so far compared LEGACY and LEXIMIN over only the lower end of selection probabilities, as this is the range in which LEXIMIN prioritizes being fair. However, even considering the *entire* range of selection probabilities, we find that LEXIMIN is quantifiably fairer than LEGACY on all instances by two established metrics of fairness, the Gini Coefficient and the geometric mean (Table A.1 in Appendix A.2). For example, across instances excluding mass, LEXIMIN decreases the Gini coefficient, a standard measure of inequality, by between 5 and 16 percentage points (median: 12; negligible improvement on mass). Strikingly, the 16-point improvement in the Gini coefficient achieved by LEXIMIN on the instance obf (from 59% to 43%) approximately reflects the gap between relative income inequality in Namibia (59% in 2015) and the United States (42% in 2019) [Wor22].

[Wor22] World Bank Group (2022): *GINI index*.

## 2.5 Discussion

As the recommendations made by citizens’ assemblies increasingly impact public decision-making, so grows the urgency that selection algorithms distribute this power fairly across constituents. We have made substantial progress on this front: the optimality of our algorithmic framework conclusively resolves the search for fair algorithms for a broad class of fairness measures, and the deployment of LEXIMIN puts an end to some pool members being virtually never selected in practice.

Beyond these immediate benefits to fairness, the exchange of ideas we have initiated between practitioners and theorists presents continuing opportunities to improve panel selection in areas such as transparency. For example, for an assembly in Michigan, we assisted *of by for* \* in selecting their panel via a live lottery in which participants could easily observe the probabilities with which



**Figure 2.3:** How LEXIMIN’s output was used to select a panel via a live uniform lottery. (a) First, the output distribution was transformed into a uniform distribution over 1 000 panels, numbered 000–999. (b) The three digits determining the final panel were drawn from lottery machines, making each panel observably selected with equal probability. (c) The personalized interface (screen-captured with (b)) shows each pool member the number of panels out of 1 000 they are on, allowing them to verify their own and others’ selection probabilities. Screenshots credit: *of by for* \*.

each pool member was selected. This is an advance over the transparency possible with previous selection algorithms. We found that, in this instance, the output distribution of LEXIMIN could be transformed into a simple lottery without meaningful loss of fairness (Figure 2.3). Subsequent work by Flanigan et al. [FKP21] developed general procedures and bounds for this transformation.

The Organisation for Economic Co-operation and Development (OECD) describes citizens’ assemblies as part of a broader democratic movement to “give citizens a more direct role in [...] shaping the public decisions that affect them” [OEC20]. By bringing mathematical structure, increased fairness, and greater transparency to the practice of sortition, research in this area promises to put practical sortition on firmer foundations, and to promote citizens’ assemblies’ mission to give everyday people a greater voice.

[FKP21] Flanigan et al. (2021): *Fair Sortition Made Transparent*.

[OEC20] OECD (2020): *Innovative Citizen Participation and New Democratic Institutions*.

## 3.1 Introduction

Owing to its focus on practical applicability, the previous section approached the problem of panel-selection from a narrow perspective: out of the three stages of sortition (invitation, self-selection, and selection algorithm, Figure 1.1), our analysis considered a setting in which the former two had already occurred and in which the quotas were already set by practitioners. We pursued the most natural objective from this perspective, namely, selecting the *pool members* with equal probability (or, at least, as equal as possible). Fundamentally, however, we primarily care about equality between all members of the population rather than between the members of a pool. From a member of the population, the guarantees of the previous chapter are somewhat unwieldy: if a constituent randomly receives an invitation (and opts into the pool), then the remainder of the random process (i.e., the selection algorithm) will treat the constituent as equally as possible to their fellow pool members.

In this chapter, we aim to design a selection process with a more direct guarantee for constituents: constituents should have equal *end-to-end probabilities*, by which we mean their probabilities of going from *population* to *panel*, across the intermediate steps of being invited, opting into the pool, and being selected for the panel. At the same time, panels should still be representative in the sense that they satisfy quotas such as those in the previous chapter.

The main hurdle to both equalizing end-to-end probabilities and representativeness is the second step of the sortition process, *self-selection*. Not only is nonresponse prevalent — practitioners tell us that response rates between 2 and 5% are typical — but, more importantly, different groups participate at different rates. These disparate response rates produce *self-selection bias*, which means that the pool composition is not representative of the population, but rather skews toward groups with certain features. Our selection algorithm will counteract self-selection both on the level of groups and of individuals: to obtain representative panels, our selection algorithm must select groups that are underrepresented in the pool at higher rates, and, to equalize end-to-end probabilities, individuals who are unlikely to respond must be selected with higher probability if they respond against the odds.

End-to-end probabilities are not straight-forward to define because they do not only depend on the selection procedure, but also on whether and when the agent decides to opt into the pool. For this reason, we will design our selection process for equalizing end-to-end probabilities in a mathematical model in which constituents' decision of participating is also random, and we will assume that the selection algorithm knows each individual's probability of responding to the invitation. We will then propose a way of applying this process in practice by learning from data; we leave the question of whether the resulting process is desirable in practice for interdisciplinary discussion.

3.1	Introduction	22
3.1.1	Related Work	24
3.2	Model	24
3.3	Selection algorithm	25
3.3.1	Part I: Assigning Marginals	26
3.3.2	Part II: Rounding Marginals	27
3.4	Learning Participation Probabilities	28
3.5	Experiments	29
3.6	Discussion	32

The main contribution of this chapter is a selection algorithm that, in the random model, equalizes end-to-end probabilities while also allowing the deterministic satisfaction of quotas. In particular, our algorithm satisfies the following desiderata:

- ▶ *End-to-End Fairness*: The algorithm selects the panel via a process such that all members of the population appear on the panel with probability asymptotically close to  $k/n$ . By linearity of expectation, this also implies that all groups in the population, including those defined by intersections of arbitrarily many features, will be near-proportionally represented in expectation.
- ▶ *Deterministic Quota Satisfaction*: The selected panel satisfies certain upper and lower quotas enforcing approximate representation for a set of specified features.
- ▶ *Computational Efficiency*: The algorithm returns a valid panel (or fails) in polynomial time.

As the last point indicates, the process might not return a representative panel with some small positive probability, which is unavoidable in our model. Indeed, since agents independently decide whether to respond to the invitation, the pool might end up so skewed that it contains no representative panel — in fact, the pool might even contain fewer members than the panel size. Nonetheless, we are able to give an algorithm that succeeds with probability converging to one, under weak assumptions mainly relating the number of invitation letters sent out to  $k$  and the minimum participation probability over all agents.

The main guiding principle underlying our selection algorithm is that it should reverse the self-selection bias occurring in the formation of the pool. We formalize this self-selection bias by assuming that each agent  $i$  in the population agrees to join the pool with some positive participation probability  $q_i$  when invited. If these  $q_i$  values are known for all members of the pool, our selection algorithm can use them to neutralize self-selection bias. To do so, our algorithm selects agent  $i$  for the panel with a probability (close to) proportional to  $1/q_i$ , conditioned on  $i$  being in the pool. This compensates for agents' differing likelihoods of entering the pool, thereby giving all agents an equal end-to-end probability. On a given pool, the algorithm assigns marginal selection probabilities to every agent in the pool. Then, to find a distribution over valid panels that implements these marginals, the algorithm randomly rounds a linear program using techniques based on discrepancy theory. Since our approach aims for a fair *distribution* of valid panels rather than just a single panel, we can give probabilistic fairness guarantees.

As we mentioned, our theoretical and algorithmic results, presented in [Section 3.3](#), take the probabilities  $q_i$  of all pool members  $i$  as given in the input. While these values are not observed in practice, we show in [Section 3.4](#) that they can be estimated from available data. We cannot directly train a classifier predicting participation, however, because practitioners collect data only on those who *do* join the pool, yielding only positively labeled data. In place of a negatively labeled control group, we use publicly available survey data, which is unlabeled (i.e., includes no information on whether its members would have joined the pool). To learn in this more challenging setting, we use techniques from *contaminated controls*, which combine the pool data with the unlabeled

sample of the population to learn a predictive model for agents' participation probabilities. In [Section 3.5](#), we use data from a real-world citizens' assembly to show that plausible participation probabilities can be learned and that the algorithm produces panels that are close to proportional across features. For a synthetic population produced by extrapolating the real data, we show that our algorithm obtains fair end-to-end probabilities.

### 3.1.1 Related Work

Our work is broadly related to existing literature on fairness in the areas of *machine learning*, *statistics*, and *social choice*. Through the lens of fair machine learning, our quotas can be seen as enforcing approximate statistical fairness for protected groups, and our near-equal selection probability as a guarantee on individual fairness. Achieving simultaneous group- and individual-level fairness is a commonly discussed goal in fair machine learning [[Bin20](#); [GKP19](#); [HC20](#)], but one that has proven somewhat elusive. To satisfy fairness constraints on orthogonal protected groups, we draw upon techniques from discrepancy theory [[Ban19](#); [BF81](#)], which we hope to be more widely applicable in this area.

This chapter addresses self-selection bias, which is routinely faced in statistics and usually addressed by sample reweighting. Indeed, our selection algorithm can be seen as a way of reweighting the pool members under the constraint that weights must correspond to the marginal probabilities of a random distribution. While reweighting is typically done by the simpler methods of post-stratification, calibration [[HE91](#)], and sometimes regression [[RBPW09](#)], we use the more powerful tool of learning with contaminated controls [[LI96](#); [WHB+09](#)] to determine weights on a more fine-grained level.

## 3.2 Model

**Agents.** Let  $N$  be a set of  $n$  agents, constituting the underlying population. Let  $F$  be a set of *features*, where feature  $f \in F$  is a function  $f : N \rightarrow V_f$ , mapping the agents to a set  $V_f$  of possible values of feature  $f$ . For example, for the feature *gender*, we could have  $V_{\text{gender}} = \{\text{male}, \text{female}, \text{non-binary}\}$ . Let the *feature-value pairs* be  $\bigcup_{f \in F} \{(f, v) \mid v \in V_f\}$ . In our example, the feature-value pairs are  $(\text{gender}, \text{male})$ ,  $(\text{gender}, \text{female})$ , and  $(\text{gender}, \text{non-binary})$ . Denote the number of agents with a particular feature-value pair  $(f, v)$  by  $n_{f,v}$ .

Each agent  $i \in N$  is described by their *feature vector*  $F(i) := \{(f, f(i)) \mid f \in F\}$ , the set of all feature-value pairs pertaining to this agent. Building on the example instance, suppose we add the feature *educationlevel*, so  $F = \{\text{gender}, \text{educationlevel}\}$ . If *educationlevel* can take on the values *college* and *no college*, a college-educated woman would have the feature-vector  $\{(\text{gender}, \text{female}), (\text{educationlevel}, \text{college})\}$ .

**Panel selection process.** Before starting the selection process, organizers of a citizens' assembly must commit to the panel's parameters. First, they must choose the number of *recipients*  $r$  who will be invited to potentially join the panel, and the required *panel size*  $k$ . Moreover, they must choose a set of features  $F$  and

[[Bin20](#)] Binns (2020): *On the Apparent Conflict between Individual and Group Fairness*.

[[GKP19](#)] Gözl et al. (2019): *Paradoxes in Fair Machine Learning*.

[[HC20](#)] Hu and Chen (2020): *Fair Classification and Social Welfare*.

[[Ban19](#)] Bansal (2019): *On a Generalization of Iterated and Randomized Rounding*.

[[BF81](#)] Beck and Fiala (1981): "Integer-making" Theorems.

[[HE91](#)] Holt and Elliot (1991): *Methods of Weighting for Unit Non-Response*.

[[RBPW09](#)] Raab et al. (2009): *Adjusting for Non-Response by Weighting*.

[[LI96](#)] Lancaster and Imbens (1996): *Case-Control Studies with Contaminated Controls*.

[[WHB+09](#)] Ward et al. (2009): *Presence-Only Data and the EM Algorithm*.

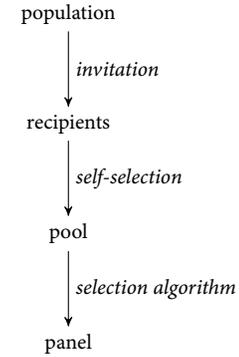
values  $\{V_f\}_{f \in F}$  over which quotas will be imposed. Finally, for all feature-value pairs  $(f, v)$ , they must choose a *lower quota*  $\ell_{f,v}$  and an *upper quota*  $u_{f,v}$ , implying that the eventual panel of  $k$  agents must contain *at least*  $\ell_{f,v}$  and *at most*  $u_{f,v}$  agents with value  $v$  for feature  $f$ . Once these parameters are fixed, the panel selection process proceeds in the familiar three stages, reproduced on the side from Figure 1.1.

In the first stage (“invitation”), the organizer of the panel sends out  $r$  letters, inviting a subset of the population — sampled with equal probability and without replacement — to volunteer for serving on the panel. We refer to the random set of agents who receive these letters as *Recipients*. Only the agents in *Recipients* will have the opportunity to advance in the process toward being on the panel.

In the third stage (“self-selection”), each letter recipient may respond affirmatively to the invitation, thereby opting into the pool of agents from which the panel will be chosen. These agents form the random set *Pool*, defined as the set of agents who received a letter and agreed to serve on the panel if ultimately chosen. We assume that each agent  $i$  joins the pool with some *participation probability*  $q_i > 0$ . Let  $q^*$  be the lowest value of  $q_i$  across all agents  $i \in N$ . A key parameter of an instance is  $\alpha := q^* r/k$ , which measures how large the number of recipients is relative to the other parameters. Larger values of  $\alpha$  will allow us the flexibility to satisfy stricter quotas.

In the third stage (“selection algorithm”), the panel organizer runs a *selection algorithm*, which selects the panel from the pool. This panel, denoted as the set *Panel*, must be of size  $k$  and satisfy the predetermined quotas for all feature-value pairs. The selection algorithm may also fail without producing a panel.

We consider the first two steps of the process to be fully prescribed. The focus of this chapter is to develop a selection algorithm for the third step that satisfies the three desiderata listed in the introduction: end-to-end fairness, deterministic quota satisfaction, and computational efficiency.



### 3.3 Selection algorithm

In this section, we give an algorithm which ensures, under natural assumptions, that every agent ends up on the panel with probability at least  $(1 - o(1)) k/n$  as  $n$  goes to infinity.<sup>1</sup> Furthermore, the panels produced by this algorithm satisfy non-trivial quotas, which ensure that the ex-post representation of each feature-value pair cannot be too far from being proportional.

Our algorithm proceeds in two phases: (I) *assignment of marginals*, during which the algorithm assigns a marginal selection probability to every agent in the pool, and (II) *rounding of marginals*, in which the marginals are dependently rounded to 0 or 1, the agents’ indicators of being chosen for the panel. As we discussed previously, our algorithm succeeds only with high probability, rather than deterministically; it may fail in phase I if the desired marginals do not satisfy certain conditions. We refer to pools on which our algorithm succeeds as *good pools*. A pool is good, to be defined precisely later, if its size and the prevalence of all feature values within it are close to their respective expected values. We leave the behavior of our algorithm on bad pools unspecified: while the algorithm

1: We allow  $k \geq 1$  and  $r \geq 1$  to vary arbitrarily in  $n$  and assume that the feature-value pairs are fixed.

may try its utmost on these pools, we give no guarantees in these cases, so the probability of representation guaranteed to each agent must come only from good pools and valid panels. Fortunately, under reasonable conditions, we show that the pool will be good with high probability, i.e., with probability converging to one as  $\alpha \rightarrow \infty$ . When the pool is good, our algorithm always succeeds, meaning that our algorithm is successful overall with high probability.

Our algorithm satisfies the following theorem, guaranteeing close-to-equal end-to-end selection probabilities for all members of the population as well as the satisfaction of quotas.

**Theorem 3.1** *Suppose that  $\alpha \rightarrow \infty$  and  $n_{f,v} \geq n/k$  for all feature-value pairs  $f, v$ . Consider a selection algorithm that, on a good pool, selects a random panel, Panel, via the randomized version of Lemma 3.3, and else does not return a panel. This process satisfies, for all  $i$  in the population, that*

$$\mathbb{P}[i \in \text{Panel}] \geq (1 - o(1)) k/n.$$

*All panels produced by this process satisfy the quotas  $\ell_{f,v} := (1 - \alpha^{-49}) k n_{f,v}/n - |F|$  and  $u_{f,v} := (1 + \alpha^{-49}) k n_{f,v}/n + |F|$  for all feature-value pairs  $f, v$ .*

The guarantees of the theorem grow stronger as the parameter  $\alpha = q^* r/k$  tends toward infinity, i.e., as the number  $r$  of invitations grows. Note that, since  $r \leq n$ , this assumption requires that  $q^* \gg k/n$ . We defer all proofs to Appendix B of the full version and discuss the preconditions in Appendix B.1.

### 3.3.1 Algorithm Part I: Assignment of Marginals

To afford equal probability of panel membership to each agent  $i$ , we would like to select agent  $i$  with probability inversely proportional to their probability  $q_i$  of being in the pool. For ease of notation, let  $a_i := 1/q_i$  for all  $i$ . Specifically, for agent  $i$ , we want  $\mathbb{P}[i \in \text{Panel} \mid i \in \text{Pool}]$  to be proportional to  $a_i$ . Achieving this exactly is tricky, however, because each agent's *selection probability* from pool  $P$ , call it  $\pi_{i,P}$ , must depend on those of all other agents in the pool, since their marginals must add to the panel size  $k$ . Thus, instead of reasoning about an agent's probability across all possible pools at once, we take the simpler route of setting agents' selection probabilities for each pool separately, guaranteeing that  $\mathbb{P}[i \in \text{Panel} \mid i \in P]$  is proportional to  $a_i$  across all members  $i$  of a good pool  $P$ . For any good pool  $P$ , we select each agent  $i \in P$  for the panel with probability

$$\pi_{i,P} := k \frac{a_i}{\sum_{j \in P} a_j}.$$

Note that this choice ensures that the marginals always sum up to  $k$ .

**Definition of good pools.** For this choice of marginals to be reasonable and useful for giving end-to-end guarantees, the pool  $P$  must satisfy three conditions, whose satisfaction defines a *good pool*  $P$ . First, the marginals do not make much

sense unless all  $\pi_{i,P}$  lie in  $[0, 1]$ :

$$0 \leq \pi_{i,P} \leq 1 \quad \forall i \in P. \quad (3.1)$$

Second, the marginals summed up over all pool members of a feature-value pair  $f, v$  should not deviate too far from the proportional share of the pair:

$$(1 - \alpha^{-.49}) k n_{f,v}/n \leq \sum_{i \in P: f(i)=v} \pi_{i,P} \leq (1 + \alpha^{-.49}) k n_{f,v}/n \quad \forall f, v. \quad (3.2)$$

Third, we also require that the term  $\sum_{i \in P} a_i$  is not much larger than  $\mathbb{E}[\sum_{i \in \text{Pool}} a_i] = \sum_{i \in N} (q_i \frac{r}{n}) \cdot a_i = r$ , which ensures that the  $\pi_{i,P}$  do not become too small:

$$\sum_{i \in P} a_i \leq r/(1 - \alpha^{-.49}). \quad (3.3)$$

Under the assumptions of our theorem, pools are good with high probability, even if we condition on a given agent  $i$  being in the pool:

**Lemma 3.2** *Suppose that  $\alpha \rightarrow \infty$  and  $n_{f,v} \geq n/k$  for all  $f, v$ . Then, for all agents  $i \in \text{Population}$ ,  $\mathbb{P}[\text{Pool is good} \mid i \in \text{Pool}] \rightarrow 1$ .*

Note that only constraint (3.1) prevents Phase II of the algorithm from running; the other two constraints just make the resulting distribution less useful for our proofs. In practice, if it is possible to rescale the  $\pi_{i,P}$  and cap them at 1 such that their sum is  $k$ , running phase II on these marginals seems reasonable.

### 3.3.2 Algorithm Part II: Rounding of Marginals

The proof of [Theorem 3.1](#) now hinges on our ability to implement the chosen  $\pi_{i,P}$  for a good pool  $P$  as marginals of a distribution over panels. This phase can be expressed in the language of randomized dependent rounding: we need to define random variables  $X_i = \mathbb{1}\{i \in \text{Panel}\}$  for each  $i \in \text{Pool}$  such that  $\mathbb{E}[X_i] = \pi_{i,P}$ . This difficulty of this task stems from the ex-post requirements on the pool, which require that  $\sum_i X_i = k$  and that  $\sum_{i: f(i)=v} X_i$  is close to  $k n_{f,v}/n$  for all feature-value pairs  $f, v$ . While off-the-shelf dependent rounding [[CVZ10](#)] can guarantee the marginals and the sum-to- $k$  constraint, it cannot simultaneously ensure small deviations in terms of the representation of all  $f, v$ .

Our algorithm uses an iterative rounding procedure based on a celebrated theorem by Beck and Fiala [[BF81](#)]. We sketch here how to obtain a deterministic rounding satisfying the ex-post constraints; the argument can be randomized using results by Bansal [[Ban19](#)] or via column generation.<sup>2</sup> The iterated rounding procedure manages a variable  $x_i \in [0, 1]$  for each  $i \in \text{Pool}$ , which is initialized as  $\pi_{i,P}$ . As the  $x_i$  are repeatedly updated, more of them are fixed as either 0 or 1 until the  $x_i$  ultimately correspond to indicator variables of a panel. Throughout the rounding procedure, it is preserved that  $\sum_i x_i = \sum_i \pi_{i,P} = k$ , and the equalities  $\sum_{i: f(i)=v} x_i = \sum_{i: f(i)=v} \pi_{i,P}$  are preserved until at most  $|F|$  variables  $x_i$  in the sum are yet to be fixed. As a result, the final panel has exactly  $k$  members, and the number of members from a feature-value pair  $f, v$  is at least  $\sum_{i: f(i)=v} \pi_{i,P} - |F| \geq (1 - \alpha^{-.49}) k n_{f,v}/n - |F|$  (symmetrically for the upper bound). Observe that our Beck-Fiala-based rounding procedure only increases

[[CVZ10](#)] Chekuri et al. (2010): *Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures*.

[[BF81](#)] Beck and Fiala (1981): *“Integer-making” Theorems*.

[[Ban19](#)] Bansal (2019): *On a Generalization of Iterated and Randomized Rounding*.

2: We describe the column-generation algorithm in [Appendix B.4.2](#) of the full version. Bansal’s randomized rounding procedure runs in polynomial time, but we found it to be very slow in practice. Our column generation procedure is faster in practice (but not formally polynomial time), and provides the same guarantees on representativeness.

the looseness of the quotas by a constant additive term beyond the losses to concentration. The concentration properties of standard dependent randomized rounding do not guarantee such a small gap with high probability. Moreover, our bound does not directly depend on the number of quotas (i.e., twice the number of feature-value pairs) but only depends on the number of features, which are often much fewer.

As we show in [Appendix B.4](#) of the full version,

**Lemma 3.3** *There is a polynomial-time selection algorithm that, given a good pool  $P$ , produces a random panel  $\text{Panel}$  such that (1)  $\mathbb{P}[i \in \text{Panel}] = \pi_{i,P}$  for all  $i \in P$ , (2)  $|\text{Panel}| = k$ , and (3)  $\sum_{i:f(i)=v} \pi_{i,P} - |F| \leq |\{i \in \text{Panel} \mid f(i) = v\}| \leq \sum_{i:f(i)=v} \pi_{i,P} + |F|$ .*

Our main theorem ([Theorem 3.1](#)) follows from a simple argument combining [Lemmas 3.2](#) and [3.3](#) ([Appendix B.5](#) of the full version).

While this theorem is asymptotic in the growth of  $\alpha$ , the same proof gives bounds on the end-to-end probabilities for finite values of  $\alpha$ . If one wants bounds for a specific instance, however, bounds uniquely in terms of  $\alpha$  tend to be loose, and one might want to relax [Condition \(3.2\)](#) of a good pool in exchange for more equal end-to-end probabilities. In this case, plugging the specific values of  $n, r, k, q^*, n_{f,v}$  into the proof allows to make better trade-offs and to extract sharper bounds.

### 3.4 Learning Participation Probabilities

The algorithm presented in the previous section relies on knowing  $q_i$  for all agents  $i$  in the pool. While these  $q_i$  are not directly observed, we can estimate them from data available to practitioners.

First, we assume that an agent  $i$ 's participation probability  $q_i$  is a function of their feature vector  $F(i)$ . Furthermore, we assume that  $i$  makes their decision to participate through a specific generative model known as *simple independent action* ([\[Fin71\]](#), as cited by [\[Wei86\]](#)). First,  $i$  flips a coin with probability  $\beta_0$  of landing on heads. Then,  $i$  flips a coin for each feature  $f \in F$ , where the coin pertaining to  $f$  lands on heads with probability  $\beta_{f,f(i)}$ . They participate in the pool if and only if all coins they flip land on heads, leading to the following functional dependency:

$$q_i = \beta_0 \cdot \prod_{f \in F} \beta_{f,f(i)}.$$

We think of  $1 - \beta_{f,v}$  as the probability that a reason specific to the feature-value pair  $f, v$  prevents the agent from participating, and of  $1 - \beta_0$  as the baseline probability of them not participating for reasons independent of their features. The simple independent action model assumes that these reasons occur independently between features, and that the agent participates iff none of the reasons occur.

If we had a representative sample of agents — say, the recipients of the invitation letters — labeled according to whether they decided participate (“positive”) or not (“negative”), learning the parameters  $\beta$  would be straightforward. However, the organizers of a citizens’ assembly only have access to the features of those

[\[Fin71\]](#) Finney (1971): *Probit Analysis*.

[\[Wei86\]](#) Weinberg (1986): *Applicability of the Simple Independent Action Model to Epidemiologic Studies Involving Two Factors and a Dichotomous Outcome*.

who enter the pool, and not of those who never respond. Without a control group, it is impossible to distinguish a feature that is prevalent in the population and associated with low participation rate from a rare feature associated with a high participation rate. Thankfully, we can use additional information: in place of a negatively-labeled control group, we use a *background sample* — a dataset containing the features for a uniform sample of agents, but without labels indicating whether they would participate. Since this control group contains both positives and negatives, this setting is known as *contaminated controls*. A final piece of information we use for learning is the fraction  $\bar{q} := |Pool|/r$ , which estimates the mean participation probability across the population. In other applications with contaminated controls, including  $\bar{q}$  in the estimation increased model identifiability [WHB+09].

To learn our model, we apply methods for maximum likelihood estimation (MLE) with contaminated controls introduced by Lancaster and Imbens [LI96]. By reformulating the simple independent action model in terms of the logarithms of the  $\beta$  parameters, their estimation (with a fixed value of  $\bar{q}$ ) reduces to maximizing a concave function.

**Theorem 3.4** *The log-likelihood function for the simple independent action model under contaminated controls is concave in the model parameters.*

By this theorem, proved in Appendix C of the full version, we can directly and efficiently estimate  $\beta$ . Logistic models, by contrast, require more involved techniques for efficient estimation [WHB+09].

### 3.5 Experiments

**Data.** We validate our  $q_i$  estimation and selection algorithm on pool data from *Climate Assembly UK* [Cliz0], a national-level citizens’ assembly co-organized by the Sortition Foundation in 2020. The panel consisted of  $k = 110$  many UK residents aged 16 and above. The Sortition Foundation invited all members of 30 000 randomly selected households, which reached an estimated  $r = 60\,000$  eligible participants.<sup>3</sup> Of these letter recipients, 1 715 participated in the pool,<sup>4</sup> corresponding to a mean participation probability of  $\bar{q} \approx 2.9\%$ . The feature-value pairs used for this panel can be read off the axis of Figure 3.1. We omit an additional feature *climate concern level* in our main analysis because only 4 members of the pool have the value *not at all concerned*, whereas this feature-value pair’s proportional number of panel seats is 6.5. To allow for proportional representation of groups with such low participation rates,  $r$  should have been chosen to be much larger. We believe that the merits of our algorithm can be better observed in parameter ranges in which proportionality can be achieved. For the background sample, we used the 2016 European Social Survey [NSD16], which contains 1 915 eligible individuals, all with features and values matching those from the panel. Our implementation is based on PyTorch and Gurobi, runs on consumer hardware, and its code is available at <https://github.com/pgoelz/entoend>. Appendix D of the full version contains details on Climate Assembly UK, data processing, the implementation, and further experiments (including the climate concern feature).

[WHB+09] Ward et al. (2009): *Presence-Only Data and the EM Algorithm*.

[LI96] Lancaster and Imbens (1996): *Case-Control Studies with Contaminated Controls*.

[Cliz0] Climate Assembly UK (2020): *The path to net zero: Full report*.

3: Note that every person in the population has equal probability (30 000/#households) of being invited. We ignore correlations between members of the same household.

4: Excluding 12 participants with gender “other” as no equivalent value is present in the background data.

[NSD16] NSD (2016): *European Social Survey Round 8 Data*.

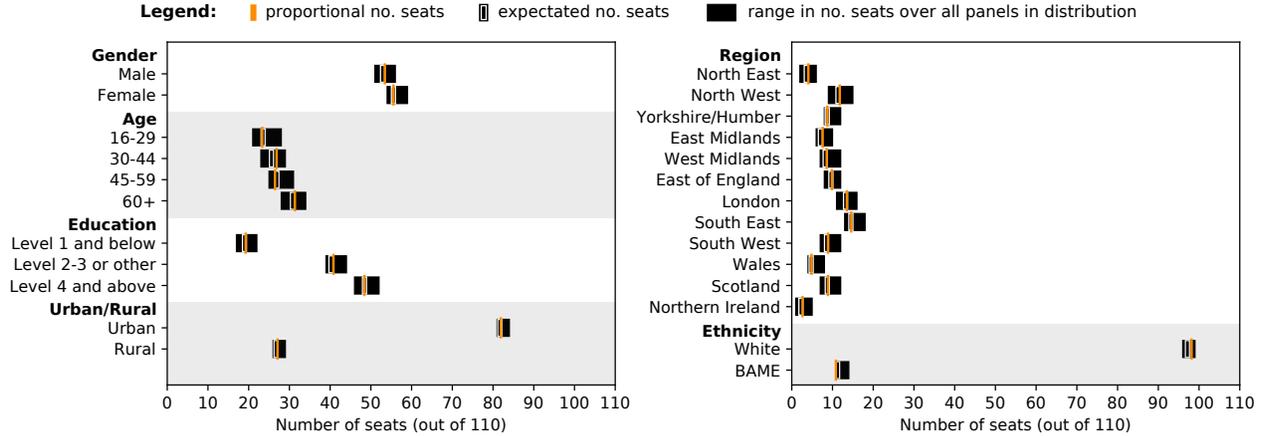
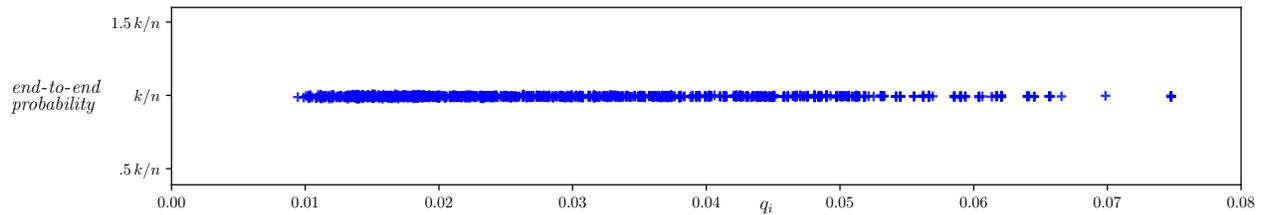


Figure 3.1: Expected and realized numbers of panel seats our algorithm gives each feature-value pair in the Climate Assembly pool.

**Estimation of  $\beta$  parameters.** We find that the baseline probability of participation is  $\beta_0 = 8.8\%$ . Our  $\beta_{f,v}$  estimates suggest that (from strongest to weakest effect) highly educated, older, urban, male, and non-white agents participate at higher rates. These trends reflect these groups' respective levels of representation in the pool compared to the underlying population, suggesting that our estimated  $\beta$  values fit our data well. Different values of the remaining feature, region of residence, seem to have heterogeneous effects on participation, where being a resident of the South West gives substantially increased likelihood of participation compared to other areas. The lowest participation probability of any agent in the pool, according to these estimates, is  $q^* = 0.78\%$ , implying that  $\alpha \approx 4.25$ . See [Appendix D.4](#) of the full version for detailed estimation results and validation.

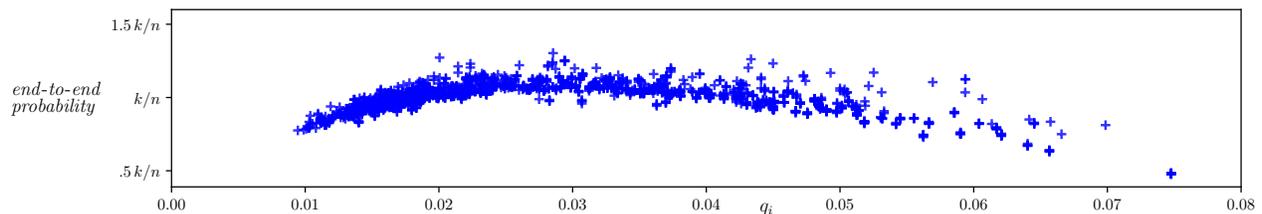
**Running the selection algorithm on the pool.** The estimated  $q_i$  allow us to run our algorithm on the Climate Assembly pool and thereby study its fairness properties for non-asymptotic input sizes. We find that the Climate Assembly pool is good relative to our  $q_i$  estimates, i.e., that it satisfies [Equations \(3.1\) to \(3.3\)](#). As displayed in [Figure 3.1](#), the marginals produced by Phase I of our algorithm give each feature-value pair  $f, v$  an expected number of seats,  $\sum_{i \in P, f(i)=v} \pi_{i,P}$ , within *one seat* of its proportional share of the panel,  $k n_{f,v}/n$ . By [Lemma 3.3](#), Phase II of our algorithm then may produce panels from these marginals in which  $f, v$  receives up to  $|F| = 6$  fewer or more seats than its expected number. However, as the black bars in [Figure 3.1](#) show, the actual number of seats received by any  $f, v$  across *any panel* produced by our algorithm on this input never deviates from its expectation by more than 4 seats. As a result, while [Theorem 3.1](#) only implies lower quotas of  $0.51 k n_{f,v}/n - |F|$  and upper quotas of  $1.49 k n_{f,v}/n + |F|$  for this instance, the shares of seats our algorithm produces lie in the much narrower range  $k n_{f,v}/n \pm 5$  (and even  $k n_{f,v}/n \pm 3$  for 18 out of 25 feature-value pairs). This suggests that, while the quotas guaranteed by our theoretical results are looser than the quotas typically set by practitioners, our algorithm will often produce substantially better ex-post representation than required by the quotas.

**End-to-end probabilities.** In the previous experiments, we were only able to argue about the algorithm’s behavior on a single pool. To validate our guarantees on individual end-to-end probabilities, we construct a synthetic population of size 60 million by duplicating the ESS participants, assuming our estimated  $q_i$  as their true participation probabilities. Then, for various values of  $r$ , we sample a large number of pools. By computing  $\pi_{i,p}$  values for all agents  $i$  in each pool, we can estimate each agent’s end-to-end probability of ending up on the panel. Crucially, we assume that our algorithm does not produce any panel for bad pools, analogously to [Theorem 3.1](#). As shown in the following graph, if  $r$  is 60 000 (as was used in Climate Assembly UK), all agents in our synthetic population, across the full range of  $q_i$ , receive probability that are barely distinguishable from  $k/n$  (averaged over 100 000 random pools):



That these end-to-end probabilities are so close to  $k/n$  also implies that bad pools are exceedingly rare for this value of  $r$ . As we show in [Appendix D.6](#) of the full version, we see essentially the same behavior for values of  $r$  down to roughly 15 000, when  $\alpha \approx 1$ . For even lower  $r$ , most pools are bad, so end-to-end probabilities are close to zero under our premise that no panels are produced from bad pools.

As a baseline, we re-run the experiment above, this time using the LEGACY algorithm to select a panel from each generated pool. Since their algorithm requires explicit quotas as input, we set the lower and upper quotas for each feature-value group to be the floor and ceiling of that group’s proportional share of seats. This is a popular way of setting quotas in current practice.



The results of this experiment show that the individual end-to-end probabilities generated by LEGACY range from below  $0.5k/n$  up to  $1.3k/n$ . In comparison to the end-to-end probabilities generated by our algorithm, those generated by LEGACY are substantially skewed, and tend to disadvantage individuals with either low or high participation probabilities. One might argue that the comparison between our algorithm and LEGACY is not quite fair, since the latter is required to satisfy stronger quotas. However, looser quotas do not improve the behavior of LEGACY; they simply make it behave more similarly to uniform sampling from the pool, which further disadvantages agents with low participation probability (for details, see [Appendix D.5](#) of the full version).

Taken together, these results illustrate that, although greedy algorithms like the one we examined achieve proportional representation of a few pre-specified groups via quotas, they do not achieve end-to-end fairness. Compared to the naive solution of uniform sampling from the pool, greedily striving for quota satisfaction does lead to more equal end-to-end probabilities, as pool members with underrepresented features are more likely to be selected for the panel than pool members with overrepresented features. However, this effect does not neutralize self-selection bias when there are multiple features, even when selection bias acts through the independent-action model as in our simulated population. Indeed, in this experiment, the greedy algorithm insufficiently boosts the probabilities of agents in the intersection of multiple low-participation groups (the agents with lowest  $q_i$ ), while also too heavily dampening the selection probability of those in the intersection of multiple high-participation groups (with highest  $q_i$ ). These observations illustrate the need for panel selection algorithms that explicitly control individual probabilities.

We cannot repeat the same experiment for LEXIMIN since its running time is prohibitively large for this setup. Based on our experiments in the previous chapter, we would expect the corresponding graph to slope less downward for agents with high participation probability (i.e., those with many overrepresented features). We would still expect to see some form of upward slope on the left, since LEXIMIN does not intentionally select underrepresented pool members with higher probability. But the left side of the plot is harder to predict because the highest selection probabilities from LEXIMIN are less structured than the smallest one, and since it is uncertain how these effects aggregate over different random pools.

### 3.6 Discussion

**Deterministic model.** In a model in which agents *stochastically* decide whether to participate, our algorithm guarantees similar end-to-end probabilities to all members of the population. In reality, however, some might argue that an agent's decision to participate when invited might not be random, but rather *deterministically* predetermined.

From the point of view of such an agent  $i$ , does our algorithm, based on a model that doesn't accurately describe their (and their peers') behavior, still grant them individual fairness? If  $i$  deterministically *participates*, the answer is yes (if not, of course they cannot be guaranteed anything). To see why, first observe that, insofar as it concerns  $i$ 's chance of ending up on the panel, all other agents might as well participate randomly.<sup>5</sup> Indeed, from agent  $i$ 's perspective, the process looks like the stochastic process where every other agent  $j$  participates with probability  $q_j$ , where  $i$  herself always participates, and where the algorithm erroneously assumes that  $i$  joins only with some probability  $q_i$ . Therefore, the pool is still good with high probability conditioned on  $i$  being in it, as argued in [Lemma 3.2](#). Even if the algorithm knew that  $q_i = 1$ ,  $i$ 's end-to-end probability would be at least  $(1 - o(1)) k/n$ , and the fact that the algorithm underestimates their  $q_i$  only increases their probability of being selected from the pool. It follows that  $i$ 's end-to-end probability in this setting still must be at least around  $k/n$ .

5: Fix a group of agents who, assuming the stochastic model, will participate if invited with probability  $q$ . Then, sampling letter recipients from this set of agents in the stochastic model is practically equivalent to sampling recipients from this group in the deterministic model, if a  $q$  fraction of the group deterministically participate.

Thus, in a deterministic model of participation, our individual guarantees are reminiscent of the axiom of population monotonicity in fair division: *If the whole population always participated when invited, every agent would reach the panel with probability  $k/n$ . The fact that some agents do not participate cannot (up to lower-order terms) decrease the selection probabilities for those who do.*

**Desirability of algorithm in practice.** If we model the agents as random, and if the machine learning can recover the  $q_i$ , should the algorithm we developed in this chapter be used in practice? One big advantage is the representation of groups that are not protected by quotas, such as for example intersectional groups: Since the agents' end-to-end probabilities are approximate equal, any subset of the population will, in expectation, be represented by a number of assembly members proportional to its size (and, we conjecture, this number will be concentrated around its expectation). Since other algorithms, including LEXIMIN, do not provide such guarantees, citizens assemblies selected via this chapter's algorithm might be more richly representative. In general, using panel selection algorithms that come with mathematical fairness guarantees can also give added legitimacy to the recommendations made by citizens' assemblies.

Our main point of uncertainty is about the transparency of the sampling process. Since an individual's probability of selection from the pool depends on the estimated  $q_i$ , the fairness of the process hinges on the entire machine-learning pipeline — data used, choice of model, and estimation methods — which is opaque to most of the population. More discussion, which ought to include practitioners and social scientists, seems necessary to tell whether these concerns can be overcome, and whether the benefits outweigh the drawbacks.

# Benefits of Stratified Sampling

# 4

## 4.1 Introduction

In the last two chapters, our selection procedures were heavily marked by self selection. This is perhaps more obvious in [Chapter 3](#), where the main principle underlying the selection algorithm was to counter self-selection bias. But self selection was also the cause for much of the complexity in [Chapter 2](#): self-selection bias caused pools to be highly unrepresentative of the population, which gave rise to the tensions between representativeness and equality that earlier algorithms struggled to navigate. Since self selection plays an outsize role in the recruitment of citizens' assemblies, our focus on self selection allowed us to address and contribute to citizens' assemblies in practice. But this focus also caused us to develop selection processes that were far more complex than those envisaged in much of the literature justifying citizens' assemblies.

In this chapter, we study panel selection in a much simpler model without self selection, which we refer to as *idealized sortition*. In this model, the familiar three stages of sortition are replaced by a single stage, in which the selection algorithm chooses panel members directly from the population. As we discuss in [Appendix A.1](#), idealized sortition is the foundation of much of the political theory justifying the merits of citizens' assemblies [[CM99](#); [Sto11](#)], and it will allow us to study much simpler selection algorithms than those in previous sections. The most natural selection algorithm is *uniform sampling*, i.e., selecting  $k$  members from the population uniformly and without replacement.

A simple but powerful property of uniform sampling is that it affords fair representation to *all possible groups in the population*, in expectation. Since every agent has the same probability of being selected, any subpopulation  $M$  is expected to send  $\frac{|M|}{n} k$  representatives to a panel of size  $k$ , where  $n$  is the total number of agents. This is an advantage over elected legislatures, in which women or racial minorities are often underrepresented, and over panels such as those generated by LEGACY and LEXIMIN, in which no such guarantee holds for any subgroups that is not protected by quotas.

Still, these representation guarantees for arbitrary groups only hold *ex ante*, i.e., in expectation over the random selection. By concentration of measure, any given group should be unlikely to be grossly over- or underrepresented if the panel is large enough, but realistic panel sizes still allow for nonnegligible deviations from the proportional share.<sup>1</sup> To mitigate these deviations, a second selection algorithm is *stratified sampling* — for example, one might choose half of the representatives among women and half among men. Assuming that this reflects the composition of the population, each individual's probability of being selected remains equal under stratified sampling, and therefore the ex-ante representation of groups continues to be proportional. In addition, stratified sampling guarantees *ex-post* fair representation to the strata, i.e., to women and men.

4.1	Introduction . . . . .	34
4.1.1	Approach and Results . . . . .	35
4.1.2	Background on Stratified Sampling . . . . .	36
4.2	Model . . . . .	37
4.3	Continuous Setting . . . . .	38
4.4	Stratified Sampling Variance . . . . .	39
4.4.1	Block Rounding . . . . .	39
4.4.2	Variance Upper Bound . . . . .	40
4.5	General Selection Algorithms . . . . .	44
4.6	Experiments . . . . .	46
4.6.1	Random Stratification . . . . .	47
4.6.2	Case Study . . . . .	47
4.7	Discussion . . . . .	50

[[CM99](#)] Carson and Martin (1999): *Random Selection in Politics*.

[[Sto11](#)] Stone (2011): *The Luck of the Draw*.

1: For example, when uniformly sampling 30 panel members from a large population of half women, half men, there is a 20% probability that one gender has at most 11 seats rather than the proportional 15 seats.

But what if the stratified aims to protect the representation of a group whose composition they do not know? In particular, we would argue that the representation of groups defined by common opinions, values, or experiences are particularly relevant to the outcome of the deliberation. Often, we cannot directly stratify for such groups since they might only emerge in the course of deliberation. Furthermore, we might not even know which of these groups will be relevant, and the groups could be too numerous to stratify for all of them.

Nonetheless, we show that stratification is an effective tool for promoting fair representation of such groups, by reducing the *variance* of their representation. Taking the perspective of the organizer of a citizens’ panel, our aim is to characterize the effect of stratification on the variance of the number of representatives from *unknown* groups, and to demonstrate that this knowledge can help more accurately represent opinion holders.

#### 4.1.1 Our Approach and Results

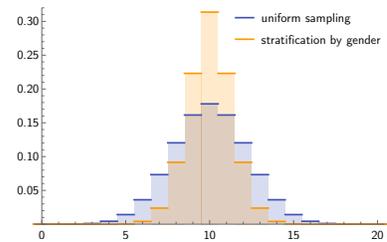
The key insight underlying our work is that the benefits of stratification extend, beyond the strata themselves, to all groups which correlate with the strata. Informally, this observation has been made as early as 1972 [MTW72]. Consider our earlier example of stratification by gender. Let  $M$  capture a certain political opinion held by, say, half of the population, and let a random variable  $A$  denote the number of agents from  $M$  in the panel. If the opinion is highly prevalent among women and rare among men, the distribution of  $A$  is more concentrated with stratification than without, as can be seen in Figure 4.1a. Surprisingly, there is virtually nothing to be lost in stratifying: If, as in Figure 4.1b,  $M$  is equally split between the strata — which is the worst case — stratification only increases the variance by a minuscule amount.

Transcending our toy example, these observations suggest a strategy of more elaborate stratification than is common today: When planning a citizens’ panel, an organizer can use public data to partition the population into many small strata, each corresponding to a small number of seats in the panel. The goal is to group together citizens who are as similar as possible. Since demographic data is predictive of political views [Pew14, p. 92], most  $M$  of interest should “polarize” the strata, i.e., most strata should either have a very high or very low prevalence of  $M$ , in which case we profit from the reduction in variance observed above. To do so, a panel organizer needs to reason about the variance of  $A$ , which can be difficult.

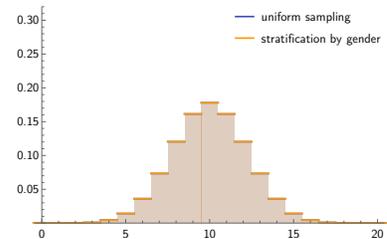
Our main contribution is a tight bound showing that — up to a factor very close to one — stratification cannot increase variance. Crucially, we propose a way of rounding seat allocations for the strata, which maintains equal probability of participation for all agents, and still satisfies the above guarantee. To gauge the benefits of stratification, we give a second upper bound which characterizes the reduction in variance due to stratification in terms of the concentration of  $M$  in every stratum.

Next, we explore the space of selection algorithms that uphold expected representation for all groups, but are not necessarily based on stratification. We show that no such algorithm dominates any other algorithm; thus, we need to assume

[MTW72] Mueller et al. (1972): *Representative Democracy via Random Selection*.



(a)  $M$  contains 90% of women, 10% of men.



(b)  $M$  contains 50% of women, 50% of men.

**Figure 4.1:** Probabilities of different numbers of agents from  $M$  in the panel. The population consists of 500 women and 500 men; the panel has size 20; 500 agents are in  $M$ , distributed among genders as indicated.

[Pew14] Pew Research Center (2014): *Beyond Red vs. Blue: The Political Typology*.

that relevant  $M$  will correlate with visible features. Finally, we show that uniform sampling is optimal from a worst-case perspective. Since stratified sampling can *never* increase variance by more than a minuscule amount, all our selection algorithms provide close-to-optimal worst-case guarantees.

Finally, we investigate the effect of stratification on variance using a large dataset containing information about demographic features and political attitudes. We find that random stratifications are helpful overall, but that only a few of them lead to significant reductions. In a case study simulating the situation of a panel organizer, we find that, using insights from our theoretical analysis, a human stratifier can simultaneously reduce the variance with respect to multiple unknown, attitude-based groups. Compared to uniform sampling, these decreases in variances correspond to an increase in panel size by multiple seats. Manual stratification also clearly outperforms a simple stratification by gender and race. Stratifications automatically generated via  $k$ -means clustering fall short of the manual stratification, but show promise.

#### 4.1.2 Background on Stratified Sampling

It is worth pointing out that stratified sampling is less flexible in expressing ex-post representativeness constraints than the quotas we considered in the previous two sections. To understand why such quotas are more general than those implied by stratified sampling, we first note that the constraints expressed by a stratification can directly be expressed as a system of quotas. This is done by turning the strata into the values of a single feature, and then setting each value's lower and upper quota to the floor and ceiling of the stratum's proportional share of panel seats. By contrast, not every system of quotas can be expressed as a stratification. This is for two reasons: first, whereas the quotas imposed by practitioners can permit an arbitrary tolerance between a feature's upper and lower quota, stratified sampling requires specifying the *exact number* of people to be chosen from each stratum. Second, and more fundamentally, quotas are often imposed on overlapping groups (e.g., the groups women and young people, where individuals can belong to both groups at once), whereas all strata must be disjoint.

To see why this restriction limits the generality of stratified sampling, consider an example in which we have overlapping categories gender and age, and want to impose quotas on women, men, people of non-binary gender, young people, and old people. In stratified sampling, one would define six disjoint strata: young women, young men, young people of nonbinary gender, old women, old men, and old people of nonbinary gender. One would then have to specify some exact number of people from each stratum; by contrast, the constraints expressed by quotas on the feature can be much more flexible since they, for example, do not directly constrain the age composition within the group of women.

As illustrated in the above example, one can implement quotas in practical settings by defining the strata to be all intersectional groups. However, this strategy does not extend practicably to the number of feature categories on which quotas are imposed in practice (in our instances, between 4 and 8). This is because imposing quotas on many orthogonal features (e.g. gender, age, region, and education level) would require setting aside a number of seats for exponentially

many combinations of these features (e.g., “female, 18–25 years old, London, no diploma”), which would quickly exceed the number of panel seats.

Our work is connected to questions of (stratified) sampling in statistics [Dal50; HT52]. Whereas we want a proportional representation of a certain feature in our sample, pollsters use the sample to obtain an accurate estimate of the prevalence of the feature. In this framework, our requirement of fair expected representation translates into requiring an unbiased estimator. In both cases, mechanisms leading to lower variance are preferred, and it is known that sampling from a continuous pool cannot increase variance, which we will show again in Section 4.3. Despite these similarities, our setting is more restrictive, since we cannot weight agents differently, which is an important technique in polling. While, in final votes, weighting representatives might be defensible, a representative’s influence on a debate cannot be weighted.

[Dal50] Dalenius (1950): *The Problem of Optimum Stratification*.

[HT52] Horvitz and Thompson (1952): *A Generalization of Sampling Without Replacement From a Finite Universe*.

## 4.2 Model

Denote the population by  $N := [n]$ , by which we mean the set  $\{1, \dots, n\}$  for some  $n \in \mathbb{N}_{\geq 1}$ . Out of these  $n$  agents, we will randomly sample a panel of size  $k$ . An unknown set of agents  $M \subseteq N$  with  $|M| = m$  share a hidden feature, which we hope to represent as fairly as possible.

Let  $\mathcal{U}(N, k)$  be the *uniform sampling algorithm*, which returns a uniformly random subset of  $k$  agents. Notice that  $\mathbb{P}[x \in \mathcal{U}(N, k)] = k/n$  for all  $x \in N$ . We denote the *representation* of  $M$  under uniform sampling, i.e., the number of agents with the hidden feature selected by  $\mathcal{U}$ , by  $U_M^{n,k} := |\mathcal{U}(N, k) \cap M|$ . Since  $\mathcal{U}_M^{n,k}$  follows a hypergeometric distribution,

$$\text{Var}(U_M^{n,k}) = k \frac{m}{n} \left(1 - \frac{m}{n}\right) \frac{n-k}{n-1}.$$

In general,  $\mathcal{A}(N, k)$  will denote a random process which generates subsets of  $N$  of size  $k$ . As above, we set  $A_M^{n,k}$  for the representation  $|\mathcal{A}(N, k) \cap M|$  of  $M$  in the panel. To ensure fair representation to every possible subset  $M \subseteq N$  of agents, we constrain such processes to satisfy  $\mathbb{E}[A_M^{n,k}] = \frac{m}{n} k$  for all  $M \subseteq N$ . By linearity of expectation, this is equivalent to every agent  $x \in N$  being selected with equal probability

$$\mathbb{P}[i \in \mathcal{A}(N, k)] = k/n. \quad (4.1)$$

Our goal is to find selection algorithms  $\mathcal{A}$  which, for unknown and arbitrary  $M$ , reduce the variance of  $A_M^{n,k}$ , compared to  $U_M^{n,k}$ .

We will be particularly interested in stratified sampling algorithms. Such an algorithm defines a partition of  $N$  into  $\ell$  strata, i.e.,  $N = N_1 \dot{\cup} N_2 \dot{\cup} \dots \dot{\cup} N_\ell$ . Let  $n_i := |N_i|$  and  $m_i := |N_i \cap M|$  for all  $i \in [\ell]$ . If the agents with the hidden feature were proportionally distributed across strata, we would expect stratum  $i$  to contain  $\frac{n_i}{n} m$  of them. Define  $\epsilon_i$  as the difference between the actual and expected number of agents with the hidden feature in stratum  $i$ , i.e.,  $m_i = \frac{n_i}{n} m + \epsilon_i$ . Note that  $\sum_{i \in [\ell]} \epsilon_i = 0$ .

After stratification, we assume that  $\mathcal{A}$  will select  $\tilde{k}_i$  agents out of each stratum  $i$  uniformly at random. Conceptually,  $\tilde{k}_i$  should be  $k_i := \frac{n_i}{n} k$ , which would guarantee that every agent is selected with the appropriate probability. However,  $k_i$  may not be integer, in which case it is necessary to randomly “round”<sup>2</sup>  $k_i$  to get  $\tilde{k}_i$ . Because the sampling happens independently in each stratum, the variance of  $A_M^{n,k}$  is

$$\sum_{i=1}^{\ell} \tilde{k}_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \frac{n_i - \tilde{k}_i}{n_i - 1},$$

conditioned on the choice of the  $\tilde{k}_i$ .

### 4.3 Warming Up in a Continuous Setting

We hope to employ stratification to decrease the variance of the number of selected agents possessing a hidden feature. There are two ways in which the discrete nature of our setting complicates studying this variance, however.

First, the  $\tilde{k}_i$  draws from a single stratum (or the  $k$  draws for uniform sampling) are not independent, since the same agent cannot be chosen multiple times. This leads to variance terms of the form  $\tilde{k}_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \frac{n_i - \tilde{k}_i}{n_i - 1}$  rather than  $\tilde{k}_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right)$ . Since the  $\tilde{k}_i$  are typically much smaller than the  $n_i$ , these correction factors have a modest impact, but they suffice to make the variances less well-behaved.

Second, the indivisibility of agents forces us to round the  $k_i$ . Again, this rounding should not change the big picture, but showing this requires careful analysis.

To understand the high-level impact of rounding on variance, we will for now ignore these complications. We study a setting in which the agents form a continuum. To make the connections between the models more suggestive, we reuse the notation from our standard model, but would like to point out where the continuous setting differs: Algorithms can now return any multiset with cardinality  $k$  and support  $N$ . Accordingly, the uniform mechanism now returns the collection of  $k$  independent uniform draws<sup>3</sup> from  $N$  and has a variance of  $k \frac{m}{n} \left(1 - \frac{m}{n}\right)$ . Each agent  $x$  is expected to appear  $k/n$  times in the panel, and this is what we require of all algorithms in this setting. All stratified-sampling algorithms will simply have  $\tilde{k}_i = k_i$  (i.e., we ignore that this number might not be integer) and sample from every stratum as the uniform algorithm samples from the whole population. Thus, the variance of stratified sampling is  $\sum_i k_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right)$ .

Surprisingly, in this setting, the benefits of stratification come entirely for free! More precisely, stratification never increases the variance over uniform sampling — but can bring it all the way down to zero if every stratum contains either only members of  $M$ , or only members of its complement. The same argument can be applied to any preexisting stratum to argue for further subdivision, down to the level of individual agents. Notably, this observation does not extend to the discrete case, where we will see that such extreme stratification leads to a potentially large increase in variance due to rounding.

**Proposition 4.1** *In the continuous setting, for any stratified-sampling algorithm  $\mathcal{A}$  and any  $M$ , it holds that  $\text{Var}(A_M^{n,k}) \leq \text{Var}(U_M^{n,k})$ .*

2: In the rounding scheme proposed in Section 4.4,  $\tilde{k}_i$  can take on integer values between  $\lfloor k_i \rfloor - 1$  and  $\lceil k_i \rceil + 1$ , so our use of the term is slightly non-standard.

3: This is not equivalent to returning a uniformly chosen multiset. For example, the multiset  $\{1, \dots, k\}$  is  $k!$  times more likely than the multiset containing the first agent  $k$  times.

*Proof.* We bound

$$\begin{aligned}\text{Var}(A_M^{n,k}) &= \sum_{i=1}^{\ell} k_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) = \sum_{i=1}^{\ell} \frac{n_i}{n} k \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \\ &= \frac{k}{n} \sum_{i=1}^{\ell} m_i \left(1 - \frac{m_i}{n_i}\right) \leq \frac{k}{n} \mu,\end{aligned}$$

where  $\mu$  is the optimal value of

$$\max_{m_i} \left\{ \sum_{i=1}^{\ell} \left( m_i - \frac{m_i^2}{n_i} \right) \mid \sum_{i=1}^{\ell} m_i = m \right\}.$$

In our setting, the  $m_i$  must be integers between 0 and  $n_i$ , but relaxing this requirement to  $m_i \in \mathbb{R}$  can only make the term larger, so the above inequality continues to hold.

Since the maximized function is concave (and smooth), and since the equality constraint is affine, the KKT conditions are sufficient for a global maximum. For this, we need a real constant  $\lambda$  which equals  $-1 + 2 \frac{m_i}{n_i}$  for all  $i$ , which means that  $\frac{m_i}{n_i} = \frac{m_j}{n_j}$  for all  $i, j \in [\ell]$ . This can easily be reconciled with the constraint  $\sum_i m_i = m$  by choosing the common value of the  $\frac{m_i}{n_i}$  as  $\frac{m}{n}$ . In other words, the worst case variance occurs when the hidden feature appears in every stratum with the same density as in the overall population. It follows that

$$\begin{aligned}\text{Var}(A_M^{n,k}) &\leq \frac{k}{n} \sum_{i=1}^{\ell} m_i \left(1 - \frac{m}{n}\right) = \frac{k}{n} \left(1 - \frac{m}{n}\right) \sum_{i=1}^{\ell} m_i \\ &= \frac{k}{n} \left(1 - \frac{m}{n}\right) m = \text{Var}(U_M^{m,k}).\end{aligned}\quad \square$$

For the rest of the chapter, we return to our standard discrete setting.

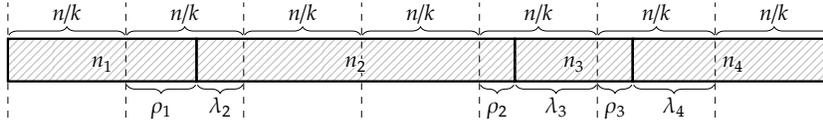
## 4.4 Main Result: The Variance of Stratified Sampling

In this section, we transport the result of [Proposition 4.1](#) from the continuous utopia to our discrete reality. We first discuss rounding, then turn to establishing an upper bound on the variance of stratified sampling.

### 4.4.1 Block Rounding

Arguably the most significant obstacle to studying the variance of stratified sampling is the construction of a rounding mechanism. How should we round  $k_i$  to get  $\tilde{k}_i$ ? To satisfy [Equation \(4.1\)](#), every stratum  $i$  must satisfy  $\mathbb{E}[\tilde{k}_i] = k_i$ . Simultaneously, the rounding must be dependent to ensure that  $\sum_i \tilde{k}_i = k$ .

How we do this can seriously affect the variance. Consider the case where half of the strata are full of agents in  $M$  and the other half contain none of them. Then, after rounding,  $A_M^{n,k}$  is deterministic and has zero variance. All variance comes from the rounding process, which directly assigns between 0 and  $O(\ell)$  additional



**Figure 4.2:** Example illustrating block rounding, with  $\ell = 4$  strata (boxes) and  $k = 7$  many blocks (areas between dashed lines). The indicated  $\lambda_i$  and  $\rho_i$  should be divided by  $n/k$  such that, for example,  $\rho_1 + \lambda_2 = 1$ . Both  $\lambda_1$  and  $\rho_4$  are zero.

seats to  $M$ . Since the rounding decisions are not independent, the introduced variance may be in  $O(\ell^2)$ , which can drastically exceed the variance of uniform sampling. Ideally, our rounding mechanism would never add more variance than what lies between the real variance and the variance of uniform sampling on the same  $M$ .

Since we were not able to make such an argument for off-the-shelf dependent randomized rounding mechanisms as defined by Chekuri et al. [CVZ09], we propose our own rounding mechanism, which we refer to as *block rounding*. Imagine lining up the agents, stratum by stratum, in the order of strata. As shown in Figure 4.2, draw a line every  $n/k$  agents to obtain a block for each of the  $k$  open seats.

Some of these blocks only contain agents from a single stratum (for example, the first, third, fourth, and seventh blocks in Figure 4.2). Such a seat will certainly be filled with an agent from that stratum. For each stratum  $i$ , denote the number of seats guaranteed to be filled by one of its members by  $g_i$ .

We make the innocuous assumption (as explicitly done in Theorem 4.2) that  $n_i \geq n/k$ , i.e., each stratum is large enough that it is entitled to at least one seat in expectation. It follows that blocks of size  $n/k$  can intersect at most two strata. Consider a block that intersects two strata  $i$  and  $i + 1$ . Let the fraction of the block intersecting stratum  $i$  be  $\rho_i$  and the fraction intersecting stratum  $i + 1$ ,  $\lambda_{i+1}$ . The seat corresponding to this block's seat will be drawn from stratum  $i$  with probability  $\rho_i$  and else (with probability  $\lambda_{i+1}$ ) from stratum  $i + 1$ , and this decision is made independently for each block. We will record these options through binary random variables  $R_i$  and  $L_{i+1}$ , specifically,  $R_i = 1$  with probability  $\rho_i$  and  $L_{i+1} = 1 - R_i$ . When the end of a stratum lines up with the border of a block, the corresponding variables are zero.

Thus, we choose  $\tilde{k}_i := g_i + L_i + R_i$  many agents from stratum  $i$ .<sup>4</sup> It is easy to see that  $\mathbb{E}[\tilde{k}_i] = k_i$  and that  $\sum_i \tilde{k}_i = k$ . Furthermore, all  $L_i$  and  $R_j$  belonging to different blocks are independent, whereas for all  $i$ ,

$$\text{CoVar}(R_i, L_{i+1}) = \mathbb{E}[R_i L_{i+1}] - \mathbb{E}[R_i] \mathbb{E}[L_{i+1}] = \mathbb{E}[0] - \rho_i \lambda_{i+1} = -\rho_i (1 - \rho_i).$$

#### 4.4.2 Variance Upper Bound

Using the foregoing rounding scheme construction, we are able to formulate and prove our main result.

**Theorem 4.2** For some  $n$  and  $k$ , let  $\mathcal{S}(N, k)$  be a stratifying algorithm based on block rounding. Suppose that every stratum  $i$  has size  $n_i \geq n/k$ , i.e., that the expected number  $k_i$  of selected representatives from  $i$  is at least 1. Then, for any

[CVZ09] Chekuri et al. (2009): *Dependent Randomized Rounding for Matroid Polytopes and Applications*.

4: Going forward, we assume that this is indeed feasible, i.e., that  $\tilde{k}_i$  never exceeds  $n_i$ . This can be enforced through extremely mild assumptions, e.g., by requiring  $k \leq n/3$  for all  $i$ .

$M \subseteq N$ ,

$$\text{Var}(A_M^{n,k}) \leq \frac{n-1}{n-k} \text{Var}(U_M^{n,k}).$$

If we assume that  $k$  grows much slower than  $n$ ,  $\frac{n-1}{n-k}$  is essentially 1 for large  $n$ . For example, for the UK Climate Assembly [Clizo], this number is around 1.000 002. Thus, we nearly recover what we found in the continuous case: that stratification can only decrease variance.

[Clizo] Climate Assembly UK (2020): *The path to net zero: Full report.*

Turning to the theorem's proof, we advise the interested reader to pay attention to the  $\epsilon_i$  terms, as defined in Section 4.2. We find their appearance, and subsequent disappearance, particularly satisfying.

*Proof of Theorem 4.2.* Since we fix the algorithm and  $M$ , we will simplify the notation by setting  $A := A_M^{n,k}$  and  $U := U_M^{n,k}$ . By the law of total variance,

$$\text{Var}(A) = \mathbb{E}[\text{Var}(A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell)] + \text{Var}(\mathbb{E}[A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell]), \quad (4.2)$$

where the outer expectation and variance range over the choices of the  $L_i$  and  $R_i$ .

Recall that  $\tilde{k}_i = g_i + L_i + R_i \in \mathbb{N}$  is the (rounded) number of agents sampled from stratum  $i$ . We bound the first summand:

$$\begin{aligned} & \mathbb{E}[\text{Var}(A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell)] \\ & \mathbb{E}\left[\sum_i \tilde{k}_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \frac{n_i - \tilde{k}_i}{n_i - 1}\right] \\ & \leq \mathbb{E}\left[\sum_i \tilde{k}_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right)\right] \quad (\tilde{k}_i \in \mathbb{N}, \text{ thus } \tilde{k}_i = 0 \text{ or } \tilde{k}_i \geq 1) \\ & = \mathbb{E}\left[\sum_i (g_i + L_i + R_i) \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right)\right] \\ & = \sum_i k_i \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \quad (\text{linearity of expectation}) \\ & = \sum_i \frac{n_i}{n} k \frac{m_i}{n_i} \left(1 - \frac{m_i}{n_i}\right) \\ & = \frac{k}{n} \sum_i m_i \left(1 - \frac{m_i}{n_i}\right) \\ & = \frac{k}{n} \sum_i \left(\frac{n_i}{n} m + \epsilon_i\right) \left(1 - \frac{\frac{n_i}{n} m + \epsilon_i}{n_i}\right) \\ & = \frac{k}{n} \sum_i \frac{m}{n} n_i + \epsilon_i - \frac{m^2}{n^2} n_i - 2 \frac{m}{n} \epsilon_i - \frac{\epsilon_i^2}{n_i} \\ & = \frac{k}{n} \left( \left(\frac{m}{n} \sum_i n_i\right) + \left(\sum_i \epsilon_i\right) - \left(\frac{m^2}{n^2} \sum_i n_i\right) - \left(2 \frac{m}{n} \sum_i \epsilon_i\right) - \left(\sum_i \frac{\epsilon_i^2}{n_i}\right) \right) \\ & = \frac{k}{n} \left( m + 0 - \frac{m^2}{n} - 0 - \sum_i \frac{\epsilon_i^2}{n_i} \right) \\ & = k \frac{m}{n} \left(1 - \frac{m}{n}\right) - \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}. \end{aligned}$$

Recall that the uniform algorithm had a variance of

$$\text{Var}(U) = k \frac{m}{n} \left(1 - \frac{m}{n}\right) \frac{n-k}{n-1}.$$

Thus, the first summand of Equation (4.2) can be bounded as

$$\mathbb{E}[\text{Var}(A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell)] \leq \frac{n-1}{n-k} \text{Var}(U) - \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}. \quad (4.3)$$

Now, consider the second summand of Equation (4.2), which was  $\text{Var}(\mathbb{E}[A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell])$ .

$$\begin{aligned} & \text{Var}(\mathbb{E}[A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell]) \\ &= \text{Var}\left(\sum_i \frac{m_i}{n_i} \tilde{k}_i\right) \\ &= \text{Var}\left(\sum_i \left(\frac{m}{n} + \frac{\epsilon_i}{n_i}\right) (g_i + L_i + R_i)\right) \\ &= \text{Var}\left(\underbrace{\sum_i \left(\frac{m}{n} + \frac{\epsilon_i}{n_i}\right) g_i}_{\text{constant}} + \underbrace{\frac{m}{n} \sum_i (L_i + R_i)}_{\text{constant}} + \sum_i \frac{\epsilon_i}{n_i} L_i + \sum_i \frac{\epsilon_i}{n_i} R_i\right) \\ &= \text{Var}\left(\sum_i \frac{\epsilon_i}{n_i} L_i + \sum_i \frac{\epsilon_i}{n_i} R_i\right). \end{aligned}$$

We can rewrite the latter variance of a linear combination as

$$\begin{aligned} &= \sum_i \left(\frac{\epsilon_i}{n_i}\right)^2 \text{Var}(L_i) + \sum_i \left(\frac{\epsilon_i}{n_i}\right)^2 \text{Var}(R_i) + 2 \sum_{i < j} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(L_i, R_j) \\ &+ 2 \sum_{i < j} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(L_i, L_j) + 2 \sum_{i < j} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(R_i, R_j) \\ &\leq \frac{1}{2} \sum_i \left(\frac{\epsilon_i}{n_i}\right)^2 + 2 \sum_{ij} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(L_i, R_j) + 2 \sum_{i < j} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(L_i, L_j) \\ &+ 2 \sum_{i < j} \frac{\epsilon_i}{n_i} \frac{\epsilon_j}{n_j} \text{CoVar}(R_i, R_j), \end{aligned}$$

where the last step used the fact that binary variables have variance of at most  $\frac{1}{4}$ . As noted, nearly all of these  $L_i$  and  $R_i$  are independent, and thus, their covariance is zero. This allows us to drop most covariance terms, with only  $\ell - 1$  many remaining. All other rounding methods we considered correlate more pairs of variables, leading to bounds on the second term that might far exceed the first term.<sup>5</sup> Block rounding allows us to continue

$$\begin{aligned} & \text{Var}(\mathbb{E}[A \mid L_1, \dots, L_\ell, R_1, \dots, R_\ell]) \\ &= \frac{1}{2} \sum_i \left(\frac{\epsilon_i}{n_i}\right)^2 + 2 \sum_{i=1}^{\ell-1} \frac{\epsilon_i}{n_i} \frac{\epsilon_{i+1}}{n_{i+1}} \text{CoVar}(R_i, L_{i+1}) \\ &= \frac{1}{2} \sum_i \left(\frac{\epsilon_i}{n_i}\right)^2 - 2 \sum_{i=1}^{\ell-1} \frac{\epsilon_i}{n_i} \frac{\epsilon_{i+1}}{n_{i+1}} \rho_i (1 - \rho_i) \end{aligned} \quad (4.4)$$

<sup>5</sup>: While dependent rounding [CVZ09] guarantees negative correlation, this is not helpful when two strata have different  $\text{sgn}(\epsilon_i)$ .

$$\begin{aligned} &\leq \frac{1}{2} \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 + 2 \sum_{i=1}^{\ell-1} \frac{|\epsilon_i|}{n_i} \frac{|\epsilon_{i+1}|}{n_{i+1}} \rho_i (1 - \rho_i) \\ &\leq \frac{1}{2} \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 + \frac{1}{2} \sum_{i=1}^{\ell-1} \frac{|\epsilon_i|}{n_i} \frac{|\epsilon_{i+1}|}{n_{i+1}}. \end{aligned}$$

For all real numbers  $a$  and  $b$ ,  $a^2 + b^2 - 2ab = (a-b)^2 \geq 0$ , and thus,  $\frac{1}{2}(a^2 + b^2) \geq ab$ . Accordingly,

$$\begin{aligned} &\leq \frac{1}{2} \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 + \frac{1}{4} \sum_{i=1}^{\ell-1} \left( \left( \frac{|\epsilon_i|}{n_i} \right)^2 + \left( \frac{|\epsilon_{i+1}|}{n_{i+1}} \right)^2 \right) \\ &= \frac{1}{2} \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 + \frac{1}{4} \sum_{i=1}^{\ell-1} \left( \left( \frac{\epsilon_i}{n_i} \right)^2 + \left( \frac{\epsilon_{i+1}}{n_{i+1}} \right)^2 \right) \\ &= \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 - \frac{1}{4} \left( \left( \frac{\epsilon_1}{n_1} \right)^2 + \left( \frac{\epsilon_\ell}{n_\ell} \right)^2 \right) \\ &\leq \sum_i \left( \frac{\epsilon_i}{n_i} \right)^2 \\ &\leq \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}, \end{aligned} \tag{4.5}$$

where the final step uses the assumption  $n_i \geq n/k$ . Inserting this bound and Equation (4.3) into Equation (4.2), we obtain the desired bound of

$$\text{Var}(A) \leq \frac{n-1}{n-k} \text{Var}(U). \quad \square$$

As we mentioned above, the theorem shows only a small multiplicative loss. Can we make it even smaller? No: The following example shows that the bound given by Theorem 4.2 is tight.

**Example 4.1** Let  $k$  divide  $n$ , and let all  $\ell := k$  strata have equal size  $n/k$ . Note that this setting will not lead to rounding. Furthermore, let all strata have equal  $m_i =: m_0$ . Under uniform sampling, the variance is  $k \frac{m_0 k}{n} \left(1 - \frac{m_0 k}{n}\right) \frac{n-k}{n-1}$ . Under stratification, the variance is

$$\ell \left( 1 - \frac{m_0}{n/k} \right) \left( 1 - \frac{m_0}{n/k} \right) \frac{n/k - 1}{n/k - 1} = k \frac{m_0 k}{n} \left( 1 - \frac{m_0 k}{n} \right).$$

We conclude that there are instances where Theorem 4.2 is tight.

If we ignore the  $\frac{n-1}{n-k}$  factor, stratification does not increase the variance. This is reassuring, but we consider stratification in the hope that it will *reduce* variance. If we stratified well and managed to concentrate the hidden feature in some strata and not in others, we would like a guaranteed improvement in variance over uniform sampling. As it turns out, the proof of Theorem 4.2 immediately gives us such a bound, just by assuming slightly larger strata.

**Corollary 4.3** *In the setting of Theorem 4.2, assume furthermore that every*

stratum has size  $n_i \geq \alpha \frac{n}{k}$  for some constant  $\alpha \geq 1$ . Then, for any  $M \subseteq N$ ,

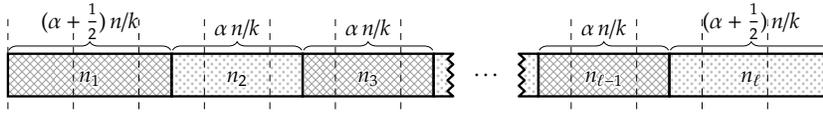
$$\text{Var}(A_M^{n,k}) \leq \frac{n-1}{n-k} \text{Var}(U_M^{n,k}) - \left(1 - \frac{1}{\alpha}\right) \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}.$$

*Proof.* In Equation (4.5) in the proof of Theorem 4.2, use  $n_i \geq \alpha \frac{n}{k}$  instead of  $n_i \geq \frac{n}{k}$ .  $\square$

This bound generalizes the previous one and, as all  $\epsilon_i$  are 0 in Example 4.1, it is also tight. But it is more interesting to look at this bound in settings with rounding. In the proof of Theorem 4.2, we saw that the expectation of the variance was essentially<sup>6</sup>

$$\frac{n-1}{n-k} \text{Var}(U_M^{n,k}) - \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}.$$

The additional  $\frac{1}{\alpha} \frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}$  in the bound of Corollary 4.3 accounts for the variance of expectation, i.e., for the increase in variance caused by rounding. The following example demonstrates that this treatment is nearly tight.



6: Up to factors of  $\frac{n_i - \tilde{k}_i}{n_i - 1}$ , which are essentially 1 for large  $n/k$ .

Figure 4.3: Example 4.2 for  $\alpha = 2$ . All agents in hatched strata are in  $M$ ; none of those in dotted strata are.

**Example 4.2** For a given integer  $\alpha \geq 1$ , let  $\ell := \frac{k-1}{\alpha}$  be an even integer. Set  $n_1 = n_\ell = (\alpha + \frac{1}{2})\frac{n}{k}$  and all other  $n_i$  to  $\alpha \frac{n}{k}$ , and assume that these numbers are integers as well. Let  $M$  be exactly all strata with odd indices. This implies that  $m = \frac{n}{2}$ , thus  $\frac{n-1}{n-k} \text{Var}(U_M^{n,k}) = \frac{k}{4}$ . The strata are laid out as in Figure 4.3:  $\ell - 1$  of the blocks of Section 4.4.1 are split half-half between an all-in- $M$  and a none-in- $M$  stratum while all other blocks are fully in  $M$  or not in  $M$ . The variance is the same as that of a sum of  $\ell - 1$  independent Bernoulli trials with probability  $1/2$ , which is  $\frac{\ell-1}{4} = \frac{k}{4\alpha} - \frac{1+\alpha}{4\alpha}$ .

Since the  $\frac{\epsilon_i}{n_i}$  alternate between  $\frac{1}{2}$  and  $-\frac{1}{2}$ , the bound of Corollary 4.3 reduces to

$$\frac{k}{4} - \left(1 - \frac{1}{\alpha}\right) \frac{k}{n} \sum_i n_i \left(\frac{\epsilon_i}{n_i}\right)^2 = \frac{k}{4} - \left(1 - \frac{1}{\alpha}\right) \frac{k}{4n} \sum_i n_i = \frac{k}{4\alpha},$$

which only leaves an additive gap of  $\frac{1+\alpha}{4\alpha} \in (1/4, 1/2]$  to the actual variance of  $A_M^{n,k}$ . This gap is independent of the instance size, so if we fix  $\alpha$  and let  $k$  go to infinity, the ratio between our bound and the actual variance converges to 1.

## 4.5 General Selection Algorithms

So far, we have compared stratified sampling to the baseline of uniform sampling. Since these are the options widely discussed in the literature, this was a natural

choice. Nevertheless, we should reassure ourselves that we do not overlook novel selection algorithms that might surpass both in terms of variance.

In this section, we consider general algorithms  $\mathcal{A}(N, k)$ , not necessarily based on sampling, for some  $k \geq 2$ . Every such algorithm is uniquely defined by probabilities  $p_C$  for all  $C \in \binom{N}{k}$ , where each  $p_C$  gives the probability of choosing the panel  $C$ .<sup>7</sup> In the following,  $C$  always ranges over  $\binom{N}{k}$ . We continue to require that the algorithm preserve expectations, which is equivalent to selecting each agent  $x \in N$  with probability  $\sum_{C \ni x} p_C = k/n$ .

<sup>7</sup>: For a set  $N$  and an integer  $k$ ,  $\binom{N}{k}$  denotes the set of all subsets of  $N$  with cardinality  $k$ .

After fixing  $\mathcal{A}$ , fix a subset  $M$  of  $N$ . Since  $N$  and  $k$  are clear, we drop the superscript from  $A_M^{n,k}$  and extend the notation  $A_{M'} := |\mathcal{A} \cap M'|$  to other subsets  $M'$  of  $N$ . Furthermore, we set  $A_x := A_{\{x\}}$  for all  $x \in N$ . Then, we can write the variance of  $A_M$  as

$$\begin{aligned} \text{Var}(A_M) &= \text{Var}\left(\sum_{x \in M} A_x\right) = \sum_{x \in M} \underbrace{\text{Var}(A_x)}_{=\frac{k}{n}\left(1-\frac{k}{n}\right)} + 2 \sum_{x < y \in M} \underbrace{\text{CoVar}(A_x, A_y)}_{=\mathbb{P}[A_x \wedge A_y] - \left(\frac{k}{n}\right)^2} \\ &= |M| \frac{k}{n} - |M|^2 \left(\frac{k}{n}\right)^2 + 2 \sum_{x < y \in M} \mathbb{P}[A_x \wedge A_y]. \end{aligned} \quad (4.6)$$

For specific priors on  $M$ , some complicated  $\mathcal{A}$  might reduce the expected variance better than any algorithm based on stratification. In general, however, we show that no such algorithm is a clearly superior choice. For this, we show that no algorithm  $\mathcal{A}$  dominates another algorithm  $\mathcal{A}'$ , i.e., strictly decreases the variance on some  $M$  without increasing the variance on any  $M$ ; the proof is relegated to [Appendix A.1](#) of the full version.

**Proposition 4.4** *Let there be two algorithms  $\mathcal{A}(N, k)$  and  $\mathcal{A}'(N, k)$ , and let their corresponding random variables be called  $A_M$  and  $A'_{M'}$ , respectively. If  $\text{Var}(A_M) < \text{Var}(A'_{M'})$  for some  $M$ , there is an  $M'$  such that  $\text{Var}(A_{M'}) > \text{Var}(A'_{M'})$ .*

Thus, no matter how we choose to stratify, we have the peace of mind that no other selection algorithm is universally preferable.

Another reason for selecting an algorithm outside of stratification might be if it had a much better worst-case guarantee. In other words, such a hypothetical algorithm would guarantee a low variance to all groups  $M$  in the population, which might be attractive in the absence of much information about  $M$ . As we show in [Proposition 4.5](#) (whose proof is relegated to [Appendix A.2](#) of the full version), uniform sampling turns out to be optimal from this worst-case perspective. By extension, since [Theorem 4.2](#) guarantees that stratified sampling can only be marginally worse on any  $M$ , stratified sampling must also be competitive.

**Proposition 4.5** *Uniform sampling minimizes  $\max_{M \subseteq N} \text{Var}(A_M)$  among all algorithms.*

Interestingly, uniform sampling is not the only algorithm with this property: As shown by the following example, there are non-uniform selection algorithms

that have the same variance as uniform sampling on every  $M \subseteq N$ , at least for some values of  $n$  and  $k$ .

**Example 4.3** Let  $n := 6$  and  $k := 3$ . Let  $\mathcal{A}(N, k)$  be the algorithm which picks one of the ten triangles displayed in Figure 4.4, uniformly at random, and returns its vertices. In this representation, we can see that it is obtained from the uniform mechanism by dropping every second rotation of the sets, for instance, the set  $\{1, 2, 3\}$ .

One can verify that every  $1 \leq x \leq 6$  appears in the result with probability  $k/n = 1/2$ . Furthermore, every set  $x, y$  is jointly selected with probability  $\binom{k}{2} / \binom{n}{2} = 1/5$ , just as by the uniform algorithm. Thus, by Equation (4.6), the variance equals the variance of the uniform algorithm for every  $M$ .

## 4.6 Experiments

As we showed in the previous sections, there is practically no harm in stratifying to a very fine granularity. If the strata correlate well with all relevant  $M$ , the variance can be reduced considerably.

In this section, we investigate how such stratifications can be found without knowing  $M$ . Any such argument need rely on the correlation between visible features and  $M$  in a population. We explore these relationships using data from the *General Social Survey* (GSS) [SDFH18] of the years 2014 and 2016.<sup>8</sup> The subpopulations that we use of each year contain 1 714 and 1 956 agents, respectively. In all of our experiments, we set  $k$  to 50. This is a compromise between having a reasonably large panel without the panel being too large of a fraction of the population.

Since ratios of variances are hard to interpret, we benchmark stratifications by their *equivalent panel size*. For a given stratification and group  $\emptyset \subsetneq M \subseteq N$ , we can compute its normalized variance  $\text{Var}(A_M^{n,k}/k)$ . Its equivalent panel size  $k'$  is the panel size such that the uniform mechanism has the same normalized variance  $\text{Var}(U_M^{n,k'}/k')$ .<sup>9</sup>

Figure 4.5 shows the conversion between ratios of variances and equivalent panel sizes for our choice of  $n$  and  $k$ . As shown in Appendix A.3 of the full version, the equivalent panel size can be directly computed as

$$\frac{nk}{(n-k) \text{Var}(A_M^{n,k}) / \text{Var}(U_M^{n,k}) + k}.$$

If a stratification has an equivalent panel size of  $50 + x$  on a relevant  $M$ , the stratification saved  $x$  seats without reducing the accuracy of  $M$ 's representation. Based upon the Irish Citizens' Assembly's cost of roughly €10 000 per participant,<sup>10</sup> high equivalent panel sizes imply a significant reduction in the cost of a citizens' panel due to stratification.

Our simulation code, along with the exact experimental setup as an IPython notebook, are available at <https://github.com/pgoelz/sortition>.

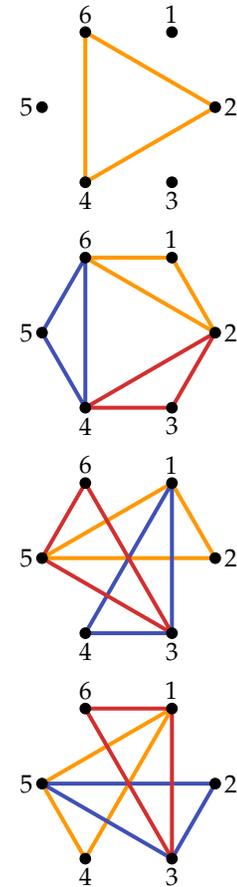


Figure 4.4: Specification of the selection algorithm in Example 4.3.

[SDFH18] Smith et al. (2018): *General Social Surveys, 2014 and 2016*.

8: The GSS is representative of the adult population living in US households (subject to being able to complete the survey in English or Spanish). Because we do not weight participants, we cannot claim representativeness. Our sample likely underrepresents individuals living in large households and individuals who initially refuse to participate in the survey.

9: We allow equivalent panel sizes to be fractional, interpolating the uniform variance using the formula  $k m/n (1 - m/n) (n - k)/(n - 1)$ .

10: The Assembly spent around €1 million in the categories “conference/catering and accommodation”, “reimbursement of travel and other expenses”, and “recruitment of members/facilitation and notetaking services” [Iri19b].

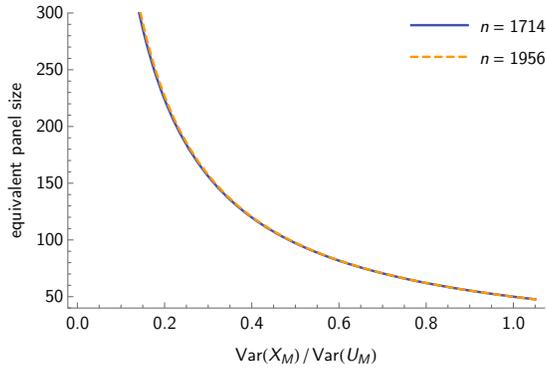


Figure 4.5: Conversion between ratio of variances and equivalent panel size for  $k = 50$ , independent of  $m$ .

### 4.6.1 Random Stratification

We begin with a quantitative overview over the effect of fine-grained stratification on variance. For this, we benchmark random stratifications based on demographic features with respect to groups  $M$  induced by random other features of the GSS dataset.

We identify 14 “demographic” features, capturing age, gender, race, social class, education, region of living, general happiness, religion, party affiliation, number of children, marital state, veteran status, urban-rural divide, and whether the individual was born abroad.<sup>11</sup> Our stratifications are induced by a random permutation of these features. We traverse the features in order, in every step subdividing every stratum greedily by the current feature, subject to the constraint that no stratum be smaller than  $n/k$ . The groups  $M$  are determined by a random column other than the demographic features. We pick a random individual, and define  $M$  to either be all agents with lower or larger value in this column.<sup>12</sup> Since similar features are usually encoded with sequential numbers, we hope that this will lead to relatively coherent groups.

As guaranteed by [Theorem 4.2](#), no stratification increases variance by more than a factor of  $\frac{n-1}{n-k}$ , which is approximately 1.03 for both years. Expressed in equivalent panel sizes, this guarantees sizes of at least 48.6 in 2014 and of 48.8 in 2016. As shown in [Figure 4.6](#), we did not observe any panel sizes close to this theoretical lower bound, with all values staying above 49.7. A large fraction of equivalent panel sizes are above 50, implying that even random stratification is more likely to be beneficial than harmful. The mean equivalent panel size is about 51.5 (2014) and 54.0 (2016), but the latter is skewed by a few large outliers. Still, a majority of stratifications give modest improvements — around one seat-equivalent — over uniform distributions.

Clearly, it is possible to increase the equivalent panel size further. It remains to be seen, however, whether targeted stratification can achieve this simultaneously for a broad set of unknown features.

### 4.6.2 Case Study: Comparison of Stratification Methods

To show this, we recreate the situation of a panel organizer. Hopefully, using insights from our theoretical analysis, a human stratifier can decrease the variance for unknown groups  $M$ , and go beyond uninformed stratification, say by the

11: See [Appendix D.1](#) of the full version.

12: We do not consider columns with more than 10% missing entries, agents who do not have a value in the given column, and groups  $M$  containing no or all agents.

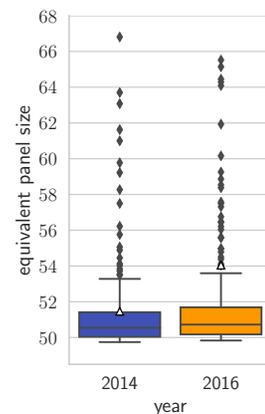


Figure 4.6: Equivalent panel size for 250 random stratifications per year. Omitting three outliers per year above 68. Triangles mark means.

intersections of gender and race. We also experiment with automatic stratification based on  $k$ -means clustering (where the number of clusters is  $\ell$ , not  $k$ ).

One collaborator took the role of the stratifier. Until the stratification was complete, we isolated them from information about the dataset (including the analyses of the previous section). Our pre-committed experimental setup can be found in [Appendix C](#) of the full version.

We want to benchmark the different approaches on groups  $M$  that might be relevant in a political context. Thus, we identified 10 “attitude” features for each of the years and projected them into binary features.<sup>13</sup> These attitude features reflect opinions touching a wide range of issues, including social liberties, economic policy, penal law, and trust in institutions. No pair of attitude features has a higher (anti-)correlation than 37%, with most pairs being well below that. For each year, a random subset of five attitude features was made available to the stratifier and the clustering algorithm. Their stratifications should have low variance for the remaining attitude features, of which the stratifier does not even know the category name. The revealed attitudes allow the stratifier to build an intuition for the political topology and to get an impression of the kinds of groups they will have to accommodate. In practice, similar information (and likely more) is available from experience, polling, and knowledge about upcoming issues. The rationale for the manual stratification is documented in [Appendix E](#) of the full version. Important elements of the high-level approach — such as the granularity of stratification, the goal of polarizing strata, and the sequential ordering of the strata — were directly motivated by our technical analysis.

We also consider a stratification based on  $k$ -means clustering. Since the demographic and revealed features form a high-dimensional space, a clustering algorithm might be better at identifying coherent subgroups than a human. We request  $\ell := 48$  clusters, the maximum number for which each stratum can still have a size of at least  $n/k$ . As documented in the experimental setup, we translate the features into real vectors in a relatively naïve way. We do not attempt to scale features differently and simply optimize the squared Euclidean distance. A more principled approach to clustering might lead to better results, but is outside of the scope of this work. We slightly deviated from our experimental setup by using *constrained*  $k$ -means clustering [[BBDo0](#)] instead of *balanced*  $k$ -means clustering. While balanced clustering constrains all clusters to have size between  $\lfloor n/\ell \rfloor$  and  $\lceil n/\ell \rceil$ , we only require the lower bound. We decided to do so after observing strongly non-contiguous clusters formed by balanced clustering even on two-dimensional toy examples.

Finally, the gender-race stratification gives one stratum to each combination of white/black/other and male/female. Stratifying by these categories can be easily implemented, and the features of race and gender are often controlled by quotas in practice.

As displayed in [Figure 4.7](#), we find that manual stratification clearly outperforms all other stratification approaches. For all tested groups, variance of distribution decreases over the baseline of uniform sampling. In four out of the six cases, this increase corresponds to an increase in panel size of more than four seats, i.e., by 8%. Clustering performs second best and marginally beats manual stratification for two attitude groups. For several groups  $M$  however, clustering performs

<sup>13</sup>: See [Appendix D.2](#) of the full version.

[[BBDo0](#)] Bradley et al. (2000): *Constrained K-Means Clustering*.

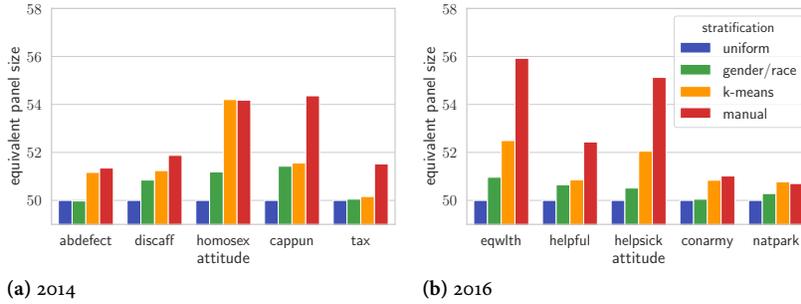


Figure 4.7: Equivalent panel size for different attitude groups and different stratification mechanisms.

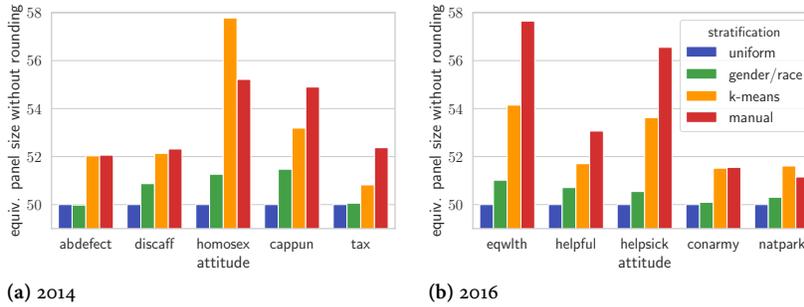
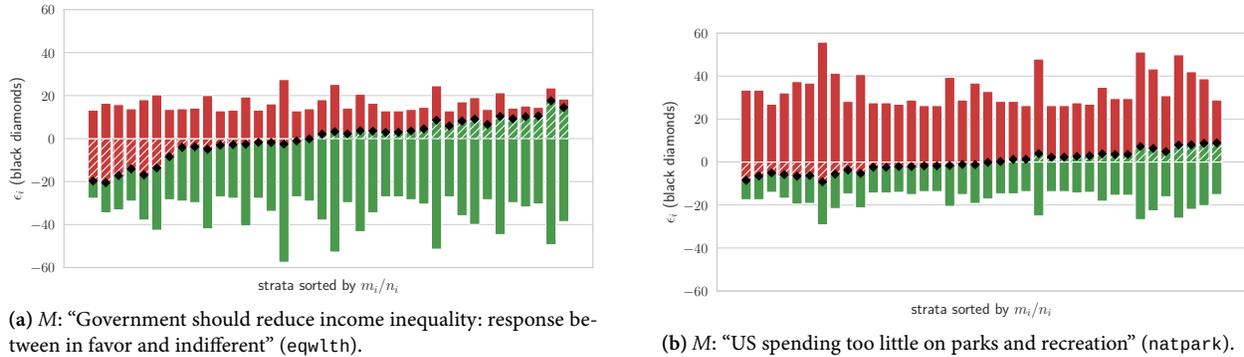


Figure 4.8: Equivalent panel size for different attitude groups and different stratification mechanisms, based only on the expectation of variance, ignoring the variance of expectation.

comparably to simple gender-race stratification, even though the manual stratification was able to make larger gains. Stratifying by race and gender leads to a slight increase in variance for one group, but still looks worthwhile given the improvements on other groups. Nevertheless, both manual or clustering-based stratification are clearly preferable when possible.

How would these results improve in absence of rounding? Depending on how much the equivalent panel size increases without rounding, it might be more or less worthwhile to optimize future stratifications in this direction. To examine this, we decompose the variance as in Equation (4.2) and completely ignore the variance of expectation. As can be seen in Figure 4.8, this leads to significant gains for both manual stratification and clustering. While the overall ordering of approaches remains the same, the clustering approach clearly profits more, with an average gain of about 1.3 seat-equivalents as compared to the 0.8 seat-equivalents gained by manual stratification. At first glance, this might be surprising because  $k$ -means leads to near-equal strata sizes. However, this does not imply that the strata boundaries line up nicely with the blocks since (due to divisibility issues) there can be at most 48 strata with size at least  $n/k$ . The variance of expectations is driven by terms of the shape  $\rho_i(1 - \rho_i)$  and of the shape  $(m_i/n_i - m_{i+1}/n_{i+1})^2$ . The distribution of these terms for both approaches is displayed in Appendix B.2 of the full version. The lining up of strata with blocks is reflected in the first kind of terms; neither kind of stratification has a clear edge there. Instead, the large rounding losses of  $k$ -means match its significantly higher terms of the second kind, which implies that it could profit from optimizing the order of strata. Adjacent strata should be similar because rounding between strata with similar concentrations of  $M$  adds less to the variance of expectation. Our manual stratifier explicitly tried to do this, and seems to have been successful. In Appendix B.3 of the full version, we illustrate this difference for the feature *homosex*. In general, we see that rounding should be considered when stratifying, both with respect to block alignment and ordering.



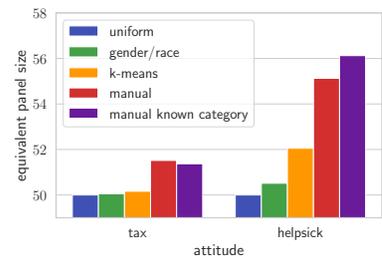
**Figure 4.9:** Distribution of different  $M$  over manual 2016 stratification. Overall height of bars is strata size  $n_i$ , height of green part (lower end to diamond) is  $m_i$ . Position of black diamonds corresponds to  $\epsilon_i$ .

We predicted that the key to low variance would be that relevant  $M$  polarize as many strata as possible. We inspect the manual stratification of the 2016 dataset to see whether different levels of polarization explain the difference in success between attitudes like `eqwlth` (high equivalent panel size) and `natpark` (low equivalent panel size). Looking at Figure 4.9, we indeed see a pronounced difference. For the feature `eqwlth` (Figure 4.9a), a substantial number of strata are polarized in each direction. By contrast, the feature `natpark` (Figure 4.9b) appears in many strata in a concentration  $m_i/n_i$  similar to the global concentration  $m/n$ . In the continuous setting, the relative reduction in variance is given by the ratio between  $\frac{k}{n} \sum_i \frac{\epsilon_i^2}{n_i}$  and  $k \frac{m}{n} (1 - \frac{m}{n})$ . This indicator for the polarization in both stratifications captures the superior performance on `eqwlth` (14.9%) over that on `natpark` (3.7%). Corresponding values for all stratifications and features are available in Appendix B.1 of the full version.

Finally, we wanted to see whether our human stratifier can stratify better when given the question defining a group  $M$ , without having information about which individuals belong to this group. The previous experiment models the situation of a general-purpose panel whose topics of discussion are not fixed at selection time. Revealing the question defining  $M$  could then reflect the situation of a panel convening to debate a fixed issue without specialized polling information on this issue. For each year, we revealed one random category description: `tax` in 2014 and `helpsick` in 2016. In both cases, our stratifier adapted the general-purpose stratification created earlier, and attempted to stratify on available features that seemed most relevant to the revealed category. While the specialized stratification added around one seat-equivalent for the `helpsick` group, variance slightly increased on the `tax` group. Further experimentation will be needed to see whether knowing the category consistently helps in manual stratification.

## 4.7 Discussion

In this chapter, we examined the effect of stratification in the idealized sortition setting. We formulated the goal of minimizing the variance for opinion groups, proposed a low-variance rounding scheme, and characterized the variance under stratification. We have applied these contributions in a case study, and our results suggest that stratification indeed has positive effects on representation



**Figure 4.10:** Equivalent panel sizes for different stratifications, including manual stratification when knowing category description.

in a realistic setting. It seems to us that there is more potential for the fields of artificial intelligence and computational social choice to contribute to sortition, starting with the following issues:

**Stratification by machine learning.** While our clustering approach fell short of manual stratifications, there is ample room for improvement. A principled distance measure in the space of demographics and attitudes and a better treatment of rounding may allow automated clustering to surpass manual approaches. Beyond that, a better solution to the machine learning problem in [Section 4.6.2](#) should distinguish between the demographic and revealed attitude features and leverage the different types of information they offer.

**Optimal stratification and selection.** Say that our belief about  $M$  is given as an independent probability of membership for every agent. Which stratification has the lowest expected variance over  $M$ ? To decrease variance, it may be useful not only to negatively correlate similar agents — as we do by placing them in the same stratum — but also to positively correlate agents at opposite ends of the opinion spectrum. Even if this is optimal, will the fairness of such an approach still be convincing? Otherwise stratification, which is already in wide use, might be the better choice.

## 5.1 Introduction

Having studied the random selection of citizens’ assemblies in depth in the preceding chapters, we now turn our focus to what happens once a citizens assembly convenes. The most important aspect of a citizens assembly’s work is *deliberation* — the extensive discussion between assembly members, which ultimately allows an assembly composed of laypeople to reach judicious recommendations on a complex issue. Though deliberation lies at the heart of a citizens’ assembly’s purpose, almost no work so far supports deliberation through computer science. Deliberation in a citizens’ assembly takes place over a number of sessions, where in each session, participants are divided into discussion groups, which we refer to as *tables*. For example, the Citizens’ Assembly of Scotland, which was convened by the Scottish Government in 2019–2020, ran over 16 sessions spread across 8 weekends; in each session, the 104 participants were divided across 12 tables.

The work underlying this chapter arises out of a collaboration with the Sortition Foundation on the design and implementation of algorithms for managing deliberation. One problem that our contacts brought up is scheduling the assignment of participants to tables, which we address in this chapter. The practitioners’ primary goal is find a schedule that, over the course of the process, allows participants to exchange ideas with as many other participants as possible. In addition, tables must be demographically diverse; in the Citizens’ Assembly of Scotland, they were diversified based on political view, age, and gender.

Currently, the Sortition Foundation as well as other nonprofits use a heuristic algorithm called `GROUPSELECT` [Ver22] to allocate tables, which the Sortition Foundation developed. Internally, this algorithm optimizes the objective of maximizing the number of pairs of participants who meet at least once, assigning no value to subsequent meetings. We see two shortcomings with this current approach: First, as we show in Section 5.6, `GROUPSELECT` performs quite poorly in terms of its chosen objective. Second, the objective itself often fails to encourage good schedules. We elaborate on this problem in Section 5.3, but an example of such a problematic situation is when all participants have met each other. At that point, the objective is indifferent between all possible assignments, and thus even a schedule repeating the same table assignment across all remaining sessions would be optimal.

To overcome these shortcomings, we must address several challenges. On a conceptual level, we need a principled measure of interaction between participants, which we seek to maximize. If interaction is measured as a function of the number of times each pair of participants meets, how much value should the first meeting between Alice and Bob have relative to the second, third, or fourth? On a technical level, we aim to develop a theoretically sound and practical algorithmic framework for optimizing our measure of interaction, with an eye towards real-world deployment.

5.1 Introduction . . . . .	52
5.1.1 Approach and Results . . . . .	53
5.1.2 Related Work . . . . .	53
5.2 Model . . . . .	54
5.3 Table Allocation and Concave Balancing . . . . .	55
5.4 Optimizing a Specific Saturation Function . . . . .	56
5.5 Simultaneous Approximation . . . . .	58
5.6 Empirical Results . . . . .	60
5.6.1 Single Objective . . . . .	61
5.6.2 Comparison of Algorithms . . . . .	62
5.7 Discussion . . . . .	63

[Ver22] Verpoort (2022): *GroupSelect*.

### 5.1.1 Our Approach and Results

It is intuitive that meetings between the same participants have diminishing marginal returns, e.g., the third meeting does not carry as much value as the second. We express this idea through what we call a *saturation function*  $f$ : for a monotone nondecreasing and concave function  $f : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ , we model the goal of maximizing interaction between participants through the submodular objective  $\hat{f} = \sum_{(i,j)} f(m_{i,j})$ , where the sum ranges over all pairs of agents  $i, j$  and  $m_{i,j}$  is the number of sessions in which  $i$  and  $j$  are assigned to the same table. For any choice of saturation function  $f$ , we obtain a practical algorithm that maximizes the corresponding objective  $\hat{f}$  within an approximation factor of  $1 - 1/e \approx 63\%$ , building on a classical result in submodular maximization [NWF78] and using ILP solver as a subroutine.

But which saturation function  $f$  should we use? Given that no objective seems universally better than the others, we pursue an approach of simultaneous approximation [SW97]. Specifically, we design an algorithm that produces schedules that  $\Omega(1/\log T)$ -approximate the objectives  $\hat{f}$  for all saturation functions  $f$  at once, where  $T$  is the number of sessions.

We also compare our different optimization algorithms with GROUPSELECT on data from seven real citizens' assemblies. We find that all our algorithms outperform GROUPSELECT by a wide margin, including when measured by its own objective. Two saturation functions, based on the harmonic and geometric series, seem promising options for optimizing schedules in practice.

### 5.1.2 Related Work

To our knowledge, one other paper seeks to support deliberation in citizens' assemblies through a practical, computational approach: Fishkin et al. [FGG+18] develop a system that automatically manages speaking times and speaker order in online deliberation. Clearly, this work addresses an orthogonal aspect of the deliberation process, and their platform could be combined with our algorithms for table allocation.

From a more theoretical perspective, an extensive line of work [CD20; FGMS17; GL16; PP15; ZLT21] proposes and analyzes mathematical models for deliberation, which we see as complementary to our approach. Whereas these papers capture the dynamics of deliberation with much more nuance than us, this chapter approaches deliberation through the lens of a practical problem, table allocation, and its interaction with deliberation.

Our use of submodular objectives follows a long tradition in AI of maximizing submodular functions to obtain diverse solutions. For example, this methodology encourages different parts of a multi-document summary to refer to different sources [CG98; LB11], sensors to be placed where they can collect complementary information [KSGo8], or papers to be assigned to reviewers with different expertise [ADF17]. In our application, the submodular objective encourages schedules to vary which pairs of assembly members meet across the sessions.

Finally, our setup resembles a classic problem in combinatoric optimization known as the social golfer problem:

[NWF78] Nemhauser et al. (1978): *An Analysis of Approximations for Maximizing Submodular Set Functions—I*.

[SW97] Stein and Wein (1997): *On the Existence of Schedules That Are Near-Optimal for Both Makespan and Total Weighted Completion Time*.

[FGG+18] Fishkin et al. (2018): *Deliberative Democracy with the Online Deliberation Platform*.

[CD20] Chung and Duggan (2020): *A Formal Theory of Democratic Deliberation*.

[FGMS17] Fain et al. (2017): *Sequential Deliberation for Social Choice*.

[GL16] Goel and Lee (2016): *Towards Large-Scale Deliberative Decision-Making*.

[PP15] Perote-Peña and Piggins (2015): *A Model of Deliberative and Aggregative Democracy*.

[ZLT21] Zvi et al. (2021): *Iterative Deliberation via Metric Aggregation*.

[CG98] Carbonell and Goldstein (1998): *The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries*.

[LB11] Lin and Bilmes (2011): *A Class of Submodular Functions for Document Summarization*.

[KSGo8] Krause et al. (2008): *Near-Optimal Sensor Placements in Gaussian Processes*.

[ADF17] Ahmed et al. (2017): *Diverse Weighted Bipartite B-Matching*.

$n$  golfers must be repeatedly partitioned into  $k$  groups, each of equal size  $s$ . Find a schedule of maximum length given that no two golfers may be placed in the same group twice.

Much work in this space has either analyzed the solutions for specific  $n$  and  $k$ , or optimized Boolean satisfiability formulations to find long schedules (e.g., [LMCS15; SSV22; TM12]). Our problem differs from the social golfer problem in two ways. First, the social golfer problem maximizes the number of sessions subject to a hard constraint on repeated meetings, whereas we minimize repeated meetings subject to a fixed schedule length. Second, whereas the social golfer problem allows to group any  $s$  golfers together, our representativeness constraints make the problem no longer symmetric and even less combinatorically tractable than the social golfer problem.

## 5.2 Model

**Table allocation problem.** An instance of the table allocation problem is a tuple consisting of a set of *agents*  $N = [n]$ , a number  $k$  of *tables*, a number of *sessions*  $T \geq 2$ , and a set of *representativeness constraints*. These representativeness constraints are given as a set  $F$  of *features*, where each feature  $\varphi \in F$  is defined by a set of agents  $A_\varphi \subseteq N$  possessing this feature, a lower quota  $\ell_\varphi$ , and an upper quota  $u_\varphi$  such that  $0 \leq \ell_\varphi \leq u_\varphi \leq \lceil n/k \rceil$ .

A *partition* for this instance partitions the agents into  $k$  disjoint tables  $N = \Delta_1 \dot{\cup} \dots \dot{\cup} \Delta_k$ , subject to two constraints: (1) each table  $\Delta_i$  has size either  $\lfloor n/k \rfloor$  or  $\lceil n/k \rceil$ , and (2) each table  $\Delta_i$  satisfies all representativeness constraints, in the sense that  $\ell_\varphi \leq |\Delta_i \cap A_\varphi| \leq u_\varphi$  for all features  $\varphi$ .<sup>1</sup> Throughout this chapter, we will assume that a given instance allows for at least one partition; whether this is the case can be easily checked using an ILP solver. Given a table allocation instance, our aim is to construct a *schedule*, which is a multiset  $Z$  over partitions containing  $T$  elements.<sup>2</sup>

**Concave balancing problem.** In Section 5.3 below, we will cast table allocation in terms of a more abstract problem, which we call the *concave balancing problem*. This framing will make explicit that, in much of our analysis, the meetings between pairs of agents — not the agents and tables themselves — are the primary object of study. This setup also highlights parts of our analysis that are not specifically tied to table allocation and might be of use in other contexts.

An instance of the concave balancing problem consists of a finite *ground set*  $G$ , a collection  $\mathcal{Z}$  of sets  $S \subseteq G$  of ground elements, a number of sessions  $T \geq 2$ , and a *saturation function*  $f : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$  that is monotone nondecreasing, concave, and satisfies  $f(0) = 0$ . For a given instance of the concave balancing problem, a *selection* is a multiset over  $\mathcal{Z}$ , and a *solution* is a selection of cardinality  $T$ . The goal of the concave balancing problem is to find a solution  $Z$  that maximizes the *objective*  $\hat{f}$ , which is a function mapping selections  $Z$  to  $\mathbb{R}_{\geq 0}$  defined in terms of  $f$  such that

$$\hat{f}(Z) := \sum_{g \in G} f(\text{number of sets in } Z \text{ that contain } g)$$

[LMCS15] Lardeux et al. (2015): *Set Constraint Model and Automated Encoding into SAT*.

[SSV22] Schmand et al. (2022): *A Greedy Algorithm for the Social Golfer and the Oberwolfach Problem*.

[TM12] Triska and Musliu (2012): *An Effective Greedy Heuristic for the Social Golfer Problem*.

1: In some assemblies in practice, partitions are further constrained by *clustering constraints*, which require that some participants be grouped together due to logistical reasons. Examples of such constraints include participants who are unwilling to be photographed, those who need translation services, and those who require disability accommodations. Given that our approach in Section 5.4 generalizes to clustering in a straight-forward way, we omit these constraints for ease of exposition.

2: A multiset over a finite support  $X$  is a function  $ms : X \rightarrow \mathbb{N}$ , where  $ms(x)$  indicates how many copies of some  $x \in X$  are contained in the multiset. We write  $|ms| = \sum_{x \in X} ms(x)$  for the cardinality of a multiset and denote multiset addition by  $+$ , multiset difference by  $-$ , and multiset inclusion by  $\sqsubseteq$ .

$$= \sum_{g \in G} f \left( \sum_{S \in \mathcal{Z}: g \in S} Z(S) \right).$$

For the saturation function  $f^1(x) := \mathbb{1}\{x \geq 1\} = \min\{x, 1\}$ , the concave balancing problem coincides with the classic maximum coverage problem [HP98] of selecting  $T$  sets such that maximally many ground elements  $g$  appear in at least one set. The saturation function  $f$  adds expressivity beyond maximizing coverage; e.g., the objective assigns value to second appearances of  $g$  if  $f(2) > f(1)$ . Generally, the concavity of  $f$  promotes schedules that contain ground elements similar numbers of times.

A function  $s$  that maps selections to  $\mathbb{R}_{\geq 0}$  satisfies *diminishing returns* if, for any two selections  $Z_1 \sqsubseteq Z_2$  and for any  $S \in \mathcal{Z}$ ,  $s(Z_1 + \{S\}) - s(Z_1) \geq s(Z_2 + \{S\}) - s(Z_2)$ . We call  $s$  *monotone* if, for all selections  $Z_1 \sqsubseteq Z_2$ ,  $s(Z_1) \leq s(Z_2)$ . One easily verifies that all objectives  $\hat{f}$  have diminishing returns and are monotone. Finally, for some  $\alpha \in (0, 1)$ , a solution  $Z$   $\alpha$ -approximates an objective  $\hat{f}$  if

$$\hat{f}(Z) \geq \alpha \cdot \max_{\text{solution } Z'} \hat{f}(Z').$$

A solution  $Z$  is a *simultaneous  $\alpha$ -approximation* if it  $\alpha$ -approximates the objectives  $\hat{f}$  for all saturation functions  $f$  at once.

[HP98] Hochbaum and Pathria (1998): *Analysis of the Greedy Approach in Problems of Maximum  $K$ -Coverage*.

### 5.3 Expressing the Table Allocation Problem as Concave Balancing

Looking at the table allocation problem by itself, it is not obvious what makes one schedule more conducive to deliberation than another, other than an intuition that it is desirable to “mix up” discussion groups between sessions. The Sortition Foundation’s work on GROUPSELECT makes an important contribution by highlighting a single, mathematically precise objective: maximizing how many pairs of assembly members meet at least once. This objective is an incomplete perspective on what makes a schedule conducive to deliberation, but it is distinguished by virtue of coming from an organization with first-hand experience and in close contact with other nonprofits organizing citizens’ assemblies.

We can express the objective optimized by GROUPSELECT by casting a given table allocation instance as a concave balancing problem: Let the ground set  $G$  be the set  $\binom{N}{2}$  of all unordered pairs of agents, and let the collection  $\mathcal{Z}$  contain, for each partition  $S = \Delta_1 \dot{\cup} \dots \dot{\cup} \Delta_k$  of the table allocation instance, the set  $\bigcup_{1 \leq i < j \leq k} \binom{\Delta_i}{2}$  of all pairs sitting at the same table in  $S$ . We will use this reduction to optimize different objectives  $\hat{f}$  for the table allocation problem throughout the chapter. GROUPSELECT’s objective reduces to the concave balancing problem if we choose the saturation function  $f^1(x) = \min\{x, 1\}$  since each pair  $i, j$  that does not meet contributes 0 to the objective  $\hat{f}^1$  and all other pairs contribute 1, whether they meet once or more often.

As mentioned in the introduction, optimizing GROUPSELECT’s objective  $\hat{f}^1$  does raise some concerns. One obvious pitfall is that  $\hat{f}^1$  does not express any preference between schedules in which all pairs meet. This is an obstacle in realistic

instances, and we will indeed see this occurring in our empirical evaluation (see the appendix of the full version).

A more subtle issue with optimizing  $\hat{f}^1$  is that prioritizing first meetings might not be worth an arbitrarily high cost in terms of which other pairs meet. In appendix of the full version, we present a table allocation problem in which it is difficult to arrange meetings for a subset  $P$  of the pairs, in the sense that (1) at most one pair in  $P$  can meet per partition without violating representativeness, (2) whenever a pair in  $P$  meets, representativeness implies that the other pairs meeting each other are essentially always the same ones, and (3) if no pair in  $P$  meets, there is a lot of freedom in who meets whom. In this instance, an algorithm optimizing  $\hat{f}^1$  will expend most sessions to make pairs in  $P$  meet one by one, even if this means that the overwhelming majority of pairs meet either excessively often or just once. In this instance, it seems preferable to forgo some first meetings in  $P$  in order to make most pairs meet a more balanced number of times.

Motivated by the above limitations of  $\hat{f}^1$ , we generalize the optimization problem proposed by the Sortition Foundation by considering saturation functions other than  $f^1$ . Each saturation function has its distinct advantages and disadvantages, which might matter to different degrees depending on the instance. For example, for any  $r \geq 2$ , consider the saturation function  $f^r(x) := \min\{x, r\}$ , for which each pair's contribution to  $\hat{f}^r$  increases by 1 per meeting up to the  $r$ 'th meeting, and does not increase beyond that. On the one hand, the objective  $\hat{f}^r$  pushes the schedule towards an ideal point in which each pair meets  $r$  times with maximum vigor. On the other hand, if the representativeness constraints force some pairs to meet fewer than  $r$  times (or more than  $r$  times),  $\hat{f}^r$  is indifferent between how equally the number of meetings below  $r$  (or above  $r$ , respectively) are spread.

In search of saturation functions whose marginal returns diminish more smoothly, two kinds of saturation functions strike us as promising. The first are the *geometric* saturation functions  $g^\beta$  (for some  $0 < \beta < 1$ ), where  $g^\beta(x) := \sum_{i=1}^x \beta^i$ . Given that the marginals  $\beta^{m_{i,j}}$  decay exponentially in the number  $m_{i,j}$  of previous meetings of the pair, the geometric objectives  $\hat{g}^\beta$  should still put much weight on the first meeting. Geometric objectives possess the intuitively appealing “self-similarity” property that, if we fix a partial schedule in which all pairs appear equally often, the problem of optimizing the remaining partitions looks just like optimizing a shorter schedule, with the objective multiplied by a constant. A final example is the *harmonic* saturation function  $h(x) := \sum_{i=1}^x 1/i$ . Since this function's marginals decrease more slowly, we would expect the objective  $\hat{h}$  to prioritize earlier meetings less radically. Note that the “self-similarity” property is not satisfied by this objective.<sup>3</sup>

## 5.4 Optimizing a Specific Saturation Function

Having built some intuition about the preference over allocation trade-offs expressed by a saturation function, we now investigate how an objective  $\hat{f}$  can be approximately optimized.

One immediate obstacle is that already the problem of simply deciding whether any partition exists for the given representativeness constraints is NP-hard (see

3: Consider a setting where we can achieve all pairs meeting exactly ten times. In an ideal setting, the problem would “reset” as if nobody had met: the trade-off between participants meeting for the  $n$ th and  $(n + 1)$ th times should be constant regardless of  $n$ . This property is satisfied by the geometric utility function, but for a harmonic utility function we will have a decreasing difference in utilities as  $n$  increases. For example, three third meetings have the same marginal utility as two second meetings ( $3/3 = 2/2$ ), but once every pair has met ten times, three twelfth meetings have a higher marginal utility than two eleventh meetings ( $3/12 > 2/11$ ).

the appendix of the full version). From a theoretical angle, this shuts the door to any prospect of developing polynomial-time algorithms, which is why we will search for algorithms that run sufficiently fast on inputs encountered in practice. Fortunately, state-of-the-art ILP solvers can reliably find a representative partition in little time. Though ILP solvers are powerful, formulating the entire maximization of the objective over schedules as an ILP would require a vast number of variables and constraints, and thus seems hopeless to solve. Therefore, our algorithmic approach will use ILP as a powerful subroutine for finding partitions, but our approach will handle in outside logic how the contributions of different partitions interact in the objective.

What will enable us to break down the optimization into generating partitions one at a time are the properties of the objectives  $\hat{f}$  we consider, namely diminishing returns, monotonicity, and that  $\hat{f}(\emptyset) = 0$ . These properties are useful since, for any multiset function over  $\mathcal{Z}$  satisfying them, Nemhauser et al. [NWF78] showed that a simple greedy algorithm returns a multiset of cardinality  $T$  whose objective value is at least a  $1 - 1/e$  fraction of the optimal objective value across all multisets of cardinality  $T$ .<sup>4</sup> This greedy algorithm iteratively constructs a multiset  $Z$  by starting from the empty multiset and  $T$  times adding the set  $S \in \mathcal{Z}$  with largest marginal increase  $\hat{f}(Z + \{S\}) - \hat{f}(Z)$ . In most cases where this greedy algorithm is run, the collection of sets  $\mathcal{Z}$  is not too large and explicitly given, which allows to identify  $S$  by enumerating  $\mathcal{Z}$ . By contrast, the set of all partitions might be exponentially large, so enumerating all of them is not an option.

Instead, we implement each step of the greedy algorithm by solving an ILP that will yield the partition with largest marginal increase. This ILP formulation makes use of the specific shape of our objectives, which decompose into a sum over pairs of agents, and which have the property that any partition's marginal contribution to a pair  $\{i, j\}$ 's summand is either zero (if  $i$  and  $j$  do not meet) or a constant value  $f(m_{i,j} + 1) - f(m_{i,j})$  (if  $i$  and  $j$  meet), where  $m_{i,j}$  denotes the number of times  $i$  and  $j$  have met before. Below we describe the ILP, whose variables are  $x_{i,\tau}$  ("agent  $i$  is allocated to table  $\tau$ ") and  $y_{\{i,j\},\tau}$  ("agents  $i$  and  $j$  are both allocated to table  $\tau$ "), for all  $i \neq j \in N$  and  $1 \leq \tau \leq k$ :

$$\begin{aligned}
& \text{maximize} && \sum_{\substack{(i,j) \in \binom{N}{2} \\ 1 \leq \tau \leq k}} (f(m_{i,j} + 1) - f(m_{i,j})) \cdot y_{\{i,j\},\tau} && (\text{maximize } \hat{f}(Z + \{S\}) - \hat{f}(Z)) \\
& \text{subject to} && \sum_{1 \leq \tau \leq k} x_{i,\tau} = 1 && \forall i \in N, \text{ (each agent on one table)} \\
& && \lfloor n/k \rfloor \leq \sum_{i \in N} x_{i,\tau} \leq \lceil n/k \rceil && \forall 1 \leq \tau \leq k, \text{ (constrain table sizes)} \\
& && \ell_\varphi \leq \sum_{i \in A_\varphi} x_{i,\tau} \leq u_\varphi && \forall 1 \leq \tau \leq k, \varphi \in F, \text{ (representativeness)} \\
& && \left. \begin{array}{l} y_{\{i,j\},\tau} \geq x_{i,\tau} + x_{j,\tau} - 1 \\ y_{\{i,j\},\tau} \leq x_{i,\tau} \\ y_{\{i,j\},\tau} \leq x_{j,\tau} \end{array} \right\} && \forall \{i,j\} \in \binom{N}{2}, \\
& && && \quad \quad \quad 1 \leq \tau \leq k, \quad (y_{\{i,j\},\tau} = x_{i,\tau} \wedge x_{j,\tau}) \\
& && x_{i,\tau} \in \{0,1\}, y_{\{i,j\},\tau} \in \{0,1\} && \forall \{i,j\} \in \binom{N}{2}, 1 \leq \tau \leq k.
\end{aligned}$$

Observe that, for each pair  $\{i, j\}$ , at most one variable  $y_{\{i,j\},\tau}$  can be nonzero, which means that the pair contributes at most  $f(m_{i,j} + 1) - f(m_{i,j})$  to the objective, as intended. Due to the quadratically many  $y_{\{i,j\},\tau}$  variables and the constraints tying

[NWF78] Nemhauser et al. (1978): *An Analysis of Approximations for Maximizing Submodular Set Functions—I*.

4: Technically, Nemhauser et al. [NWF78] prove this for submodular set functions. Our setting differs slightly from theirs since we allow sets to be selected multiple times, but the claimed result for multiset functions follows directly by duplicating all sets  $T$  times. Whereas diminishing returns and submodularity are equivalent for set functions, submodularity is a strictly weaker property than diminishing returns for multiset functions [KPV13].

them to the  $x_{i,\tau}$ , this ILP is substantially more difficult to solve than just finding a valid partition, but we will show in [Section 5.6](#) that an off-the-shelf ILP solver can optimize these programs to sufficient accuracy.

We can run the greedy maximization algorithm by iterating the following steps  $T$  times: solving the ILP, extracting the new partition from the  $x_{i,\tau}$ , adding the new partition to  $Z$ , and updating the  $m_{i,j}$ . If the ILP solver optimizes all subproblems to optimality, the resulting schedule will  $(1 - 1/e)$ -approximate the objective  $\hat{f}$  as proved by Nemhauser et al. [[NWF78](#)], and the greedy algorithm is known to outperform this approximation factor in many cases [[PST20](#)]. Even if we should be forced to terminate some ILP calls before reaching optimality, our guarantees degrade smoothly: If all ILPs return a partition whose marginal increase is at least an  $\alpha > 0$  fraction of the optimal marginal increase, the resulting schedule is still at least a  $(1 - 1/e^\alpha)$ -approximation [[GS07](#)].

[[NWF78](#)] Nemhauser et al. (1978): *An Analysis of Approximations for Maximizing Submodular Set Functions—I*.

[[PST20](#)] Pokutta et al. (2020): *On the Unreasonable Effectiveness of the Greedy Algorithm*.

[[GS07](#)] Goundan and Schulz (2007): *Revisiting the Greedy Approach to Submodular Set Function Maximization*.

## 5.5 Simultaneously Optimizing All Saturation Functions

Even though we have found a way to optimize the objective for any given saturation function  $f$ , such an approach remains not entirely satisfying given that we chose the saturation function somewhat arbitrarily. As we discussed in [Section 5.3](#), how much the saturation function should encourage pairs to meet for the  $i$ 'th time across the different  $i$  seems to depend on which distribution of meeting numbers are possible, which is hard to predict for a given instance.

This challenge of settling on a single saturation function raises the question of whether it is possible to produce schedules that perform well relative to the objectives belonging to *all* saturation functions simultaneously. Since we have seen that different objective can lead to starkly different schedules, and since there is an infinite variety of saturation functions, it might seem that finding an simultaneous  $\alpha$ -approximation might only be possible for extremely small values of  $\alpha$ . Nonetheless, [Algorithm 1](#) describes an algorithm, SIMAPPROX, which is a simultaneous  $\Omega(1/\log T)$ -approximation to all objectives. This algorithm and our analysis apply not just to table allocation problems but to any concave balancing problem; however, in the table allocation setting, the ILP from the previous section allows us to efficiently implement [Line 4](#).

The structure of SIMAPPROX closely resembles that of greedy maximization in that (using the terminology of table allocation) it constructs a schedule  $Z$  partition by partition, greedily adds partitions whose marginal increase relative to some objective  $\hat{f}$  is largest, and uses the same ILP formulation to identify

---

### Algorithm 1: SIMAPPROX

---

```

1  $Z \leftarrow \emptyset$ 
2 for  $t = 0, 1, \dots, T - 1$  do
3    $p \leftarrow \lfloor (t/T) \cdot (1 + \log_2 T) \rfloor$ 
4    $Z \leftarrow Z + \left\{ \operatorname{argmax}_{S \in \mathcal{Z}} \hat{f}^{2^p}(Z + \{S\}) \right\}$     $\triangleright$  Use ILP from Section 5.4
5 return  $Z$ 

```

---

these partitions. The big difference between both algorithms is that SIMAPPROX does not optimize marginals of the same objective in each iteration. Instead, it first optimizes marginals for  $\hat{f}^{2^0}$  for some number of steps, then marginals for  $\hat{f}^{2^1}$ , then for  $\hat{f}^{2^2}$ , through the powers of two up to around  $\hat{f}^T$ , each for a roughly equal number of steps. In particular, SIMAPPROX is computationally no more complex than the greedy maximization algorithm.

The key insight of this algorithm is that  $\alpha$ -approximating the logarithmically many objectives of the form  $\hat{f}^{2^p}$  (for some  $p$ ) suffices to approximate *all* objectives within a constant factor of  $\alpha$ . Thus, our proof that SIMAPPROX is a simultaneous  $\Omega(1/\log T)$ -approximation proceeds in three steps: First, we show that the schedule returned by the algorithm  $\Omega(1/\log T)$ -approximates all  $\hat{f}^r$  where  $r$  is a power of two (Lemma 5.1). Second, we show that the solution approximates the objectives  $\hat{f}^r$  for all  $r$  (Lemma 5.2). Finally, we prove that this implies simultaneous approximation for all objectives  $\hat{f}$  (Theorem 5.3). We sketch these arguments below and defer the formal proofs to the appendix of the full version.

**Lemma 5.1** For each  $0 \leq p \leq \log_2 T$ , the solution  $Z$  returned by SIMAPPROX approximates  $\hat{f}^{2^p}$  within a factor of  $(1 - 1/e) \cdot (\frac{1}{1+\log_2 T} - \frac{1}{T})$ .

*Proof sketch.* Since  $\hat{f}^{2^p}$  is greedily optimized in roughly  $T/\log T$  of the steps, the objective value is at least a  $(1 - 1/e)$  fraction of the optimal objective value obtained by any schedule of length  $T/\log T$ , and this holds despite the steps optimizing other objectives coming before and after. Since  $\hat{f}^{2^p}$  has diminishing returns, the optimal objective value for a schedule of length  $T/\log T$  is at least a  $1/\log T$  fraction of the optimal objective value for a schedule of full length  $T$ .  $\square$

**Lemma 5.2** For each  $1 \leq r \leq T$ , the solution  $Z$  returned by SIMAPPROX approximates  $\hat{f}^r$  within a factor of  $\frac{1-1/e}{2} \cdot (\frac{1}{1+\log_2 T} - \frac{1}{T})$ .

*Proof sketch.* For two values  $r_1 \approx r_2$ , the objectives  $\hat{f}^{r_1}$  and  $\hat{f}^{r_2}$  are close together to the point that, if  $r_1 \leq r_2$ , any schedule that  $\alpha$ -approximates  $\hat{f}^{r_1}$  must at least  $\alpha \cdot \frac{r_1}{r_2}$ -approximate  $\hat{f}^{r_2}$ . For a given  $r$ , let  $2^p$  denote its next-lower power of two. By Lemma 5.1,  $\hat{f}^{2^p}$  is  $\Omega(1/\log T)$ -approximated by  $Z$ ; so  $Z$  must  $\frac{2^p}{r} \cdot \Omega(1/\log T) \geq 1/2 \cdot \Omega(1/\log T)$ -approximate  $\hat{f}^r$ .  $\square$

**Theorem 5.3** The solution  $Z$  returned by SIMAPPROX is a simultaneous  $\alpha$ -approximation, for  $\alpha = \frac{1-1/e}{2} \cdot (\frac{1}{1+\log_2 T} - \frac{1}{T}) \in \Omega(1/\log T)$ .

*Proof sketch.* As we show in the appendix of the full version, the  $f^r$  form a sort of “basis” of the space of saturation functions in the sense that, for any saturation function  $f$  and any  $T$ , there exist nonnegative weights  $\{w_i\}_{1 \leq i \leq T}$  such

that  $f(x) = \sum_{i=1}^T w_i \cdot f^i(x)$  for all  $0 \leq x \leq T$ . Note that it must then also hold that  $\hat{f} = \sum_{i=1}^T w_i \cdot \hat{f}^i$ . For any saturation function  $f$ , by [Lemma 5.2](#), it holds that

$$\begin{aligned} \hat{f}(Z) &= \sum_{i=1}^T w_i \cdot \hat{f}^i(Z) \geq \sum_{i=1}^T w_i \cdot \alpha \cdot \max_{\text{solution } Z'} \hat{f}^i(Z') \\ &= \alpha \cdot \sum_{i=1}^T \max_{\text{solution } Z'} w_i \cdot \hat{f}^i(Z') \geq \alpha \cdot \max_{\text{solution } Z'} \sum_{i=1}^T w_i \cdot \hat{f}^i(Z') \\ &= \alpha \cdot \max_{\text{solution } Z'} \hat{f}(Z'). \quad \square \end{aligned}$$

In the appendix of the full version, we show that SIMAPPROX’s simultaneous approximation ratio of  $\Omega(1/\log T)$  is nearly optimal for the concave balancing problem, up to a log log factor:

**Theorem 5.4** *There exists a family of concave balancing instances such that no solution has a simultaneous approximation ratio larger than  $O(\log \log T / \log T)$ . This holds even if all sets  $S \in \mathcal{Z}$  have equal cardinality as in the table allocation problem.*

In these instances, the ground elements are partitioned into multiple blocks, and each block represents a different tradeoff between (a) how many ground elements of the block are included in a set in  $\mathcal{Z}$  and (b) how many ground elements are in the block overall. For large values of  $r$ ,  $\hat{f}^r$  is maximized by choosing sets from blocks scoring high on (a) because they cover many ground elements per set. For small  $r$ , by contrast, blocks scoring high on (b) allow to avoid selecting ground elements more than  $r$  times, which would not help  $\hat{f}^r$ . Since scoring high on different objectives  $\hat{f}^r$  requires selecting disjoint sets, no solution can simultaneously approximate them within a high factor. We conjecture that [Theorem 5.4](#)’s impossibility on simultaneous approximation extends to the table allocation problem; however, the symmetry between tables and the transitivity between which pairs can simultaneously meet make constructing analogous instances highly cumbersome.

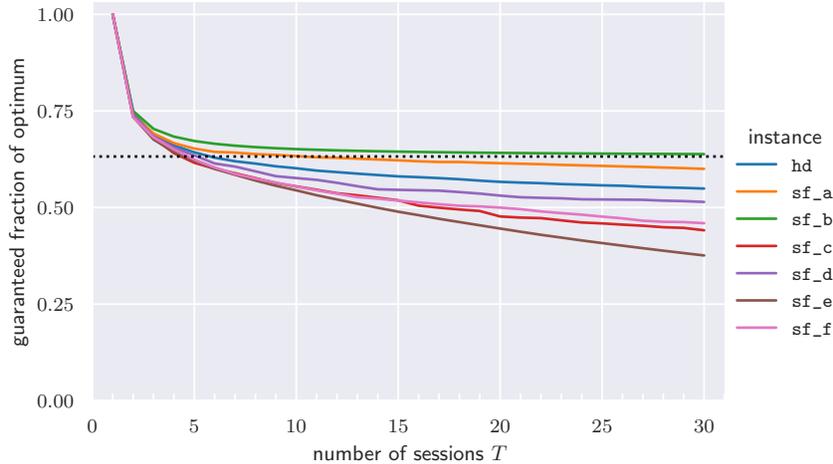
## 5.6 Implementation and Empirical Results

We have implemented all algorithms developed in this chapter in Python, using Gurobi as our ILP solver. We include our implementation in the supplementary material and will release it as open source. Currently, we are working with the Sortition Foundation to incorporate our algorithms into the tool that hosts GROUPSELECT [\[Ver22\]](#), which will allow users to switch to our improved algorithms with little effort.

[\[Ver22\]](#) Verpoort (2022): *GroupSelect*.

We perform our experiments on seven datasets, each based on data from a real citizens’ assembly. Two of these datasets, sf\_e and sf\_f, directly correspond to assemblies coorganized by the Sortition Foundation. The other five datasets, sf\_a through sf\_d and hd, are based on data from other assemblies that we already used in [Chapter 2](#). For these latter events, we do not have access to the members who ended up being drawn for the assembly, but we can “re-run” the lottery process using the selection software Panelot [\[GR20\]](#) to obtain an assembly that satisfies the actual representativeness constraints. In the appendix of the full

[\[GR20\]](#) Gözl and Rusak (2020): *Panelot*.



**Figure 5.1:** Approximation certificates for the greedy algorithm on  $\hat{g}^{1/2}$ , guaranteeing near-optimality. The dashed line marks  $1 - 1/e$ .

version, we describe the processing of these datasets and the experimental setup in more detail. To compute experiments in parallel, we run them on an AWS EC2 C5 instance with a 3.6 GHz processor, 16 threads, and 32 GB of RAM. Given that we limit each experiment to a single thread, individual running times of our algorithms are comparable to consumer hardware.

### 5.6.1 How Well Does the Greedy Algorithm Optimize Its Objective?

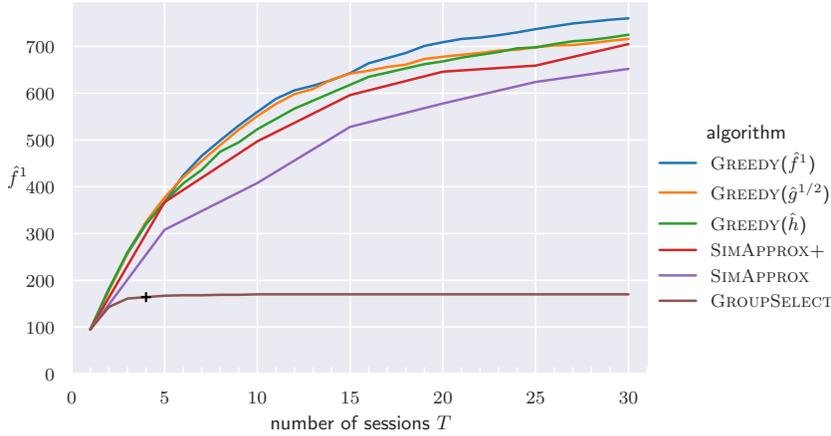
Since our greedy optimization approach is predicated on an ILP solver’s ability to solve ILPs of the form presented in Section 5.4 in reasonable time, we first evaluate how much time the ILP solver needs to be given per session for the greedy algorithm to optimize its submodular objective well. In the appendix of the full version, we show that when optimizing  $\hat{f}^1$ ,  $\hat{g}^{1/2}$ , and  $\hat{h}$ , increasing the timeouts generally leads to higher objective values, but that these increases level off after around 60 seconds. On instance `sf_c`, however, optimizing  $\hat{f}^1$  still leads to erratic optimization behavior at this timeout, which is indicative of insufficient optimization time. In order to increase the clarity of our empirical results, we thus set the optimization timeout to 120 seconds from here on, a running time which we believe to still be acceptable in practice.<sup>5</sup>

Ideally, we want to know how close the schedules produced by the greedy algorithm are to optimal, but this is impossible to exactly evaluate because we see no way of finding the optimal schedules for nontrivial instances. We can, however, modify the greedy algorithms to produce, in addition to a schedule, what we call a *certificate of approximation*, which is a fraction  $\alpha$  such that the produced schedule is guaranteed to be at least an  $\alpha$ -approximation of the optimal schedule.<sup>6</sup> As shown in Figure 5.1, for example, greedily optimizing the objective  $\hat{g}^{1/2}$  produces schedules that are a 0.45-approximation or better across all instances and numbers of sessions we study. We stress that these certificates are lower bounds, and that the schedules are likely to be much closer to optimal than is guaranteed by the certificates. For example, the perfect greedy algorithm (i.e., with perfectly optimal ILP solutions) would have a certificate of  $1 - (1 - \frac{1}{T})^T \approx 0.63$ , but typically performs much closer to optimal.<sup>7</sup> The proximity of the certificates

5: For an assembly with a high number of sessions (say, 30), the total optimization runs in around one hour, which is the runtime of LEXIMIN for large assemblies (Table 2.1).

6: Calculating these certificates is possible since (1) the ILP solver returns, in every step, not only a new partition but also an upper bound on the largest possible marginal increase, and since (2) these bounds naturally fit into the approximation bound by Nemhauser et al. [NWF78]. This ex post analysis combines the strengths of ILP and submodular maximization and is, to our knowledge, novel.

7: A second place where the certificates are conservative is that the ILP solver often struggles with tightening the upper bounds. Thus, each partition’s marginals are probably closer to optimal than reflected in our bounds.



**Figure 5.2:** Performance of different algorithms on instance `sf_f` as measured by  $\hat{f}^1$ . The cross marks the historically chosen schedule.

to this number suggests to us that terminating the ILP solver yields schedules that are nearly as good as those of the perfect greedy algorithm and not far from the optimal objective value.

### 5.6.2 Comparison across Table-Allocation Algorithms

After having measured the greedy algorithm in terms of the objective it specifically aims to optimize, we now compare the performance of different algorithms on a given instance and according to the same metric. In Figure 5.2, we show such results for `sf_f` and  $\hat{f}^1$ ; experiments for other instances and objectives can be found in the appendix of the full version. This scenario is particularly interesting to investigate, since the Sortition Foundation did, in fact, maximize  $\hat{f}^1$  using GROUPSELECT for this assembly, and since we know the table allocation (for  $T = 4$  sessions) determined at the time. As the figure shows quite dramatically, GROUPSELECT cannot compete with our other algorithms. Indeed, the Sortition Foundation chose a schedule with 164 distinct meetings for four sessions. By contrast, greedily maximizing  $\hat{f}^1$  yields an objective value of nearly twice that, at 320 distinct meetings. Across our datasets and objective functions, GROUPSELECT leads to objective values that stagnate at a much lower level than what our algorithms can achieve. This observation is a powerful argument for practitioners to move away from GROUPSELECT.

As is not very surprising, greedily optimizing  $\hat{f}^1$  produces schedules with many unique meetings. Given that `sf_f` has  $\binom{40}{2} = 780$  pairs of agents, around 90% of pairs meet at least once within the first 20 sessions. More surprisingly, greedily optimizing a geometric objective or the harmonic objective leads to numbers of distinct meetings that are nearly as high, across all numbers of sessions  $T$  we study. Indeed, throughout our experiments, we see that greedily optimizing  $\hat{g}^{1/2}$  or  $\hat{h}$  leads to “well-rounded” schedules in the sense that they perform well according to other objective metrics, which makes either algorithm an attractive option for adoption in practice. Optimizing  $\hat{f}^1$  tends to perform very well on other objectives when  $T$  is small but falls behind for larger  $T$ , when encouraging, say, second meetings becomes an important aspect of what makes a partition contribute to the objective.

A straight-forward implementation of SIMAPPROX does not perform as well as the above-mentioned algorithms, even if still much better than GROUPSELECT. A possible explanation is that SIMAPPROX spends much of its time optimizing objectives  $\hat{f}^r$  for fairly large  $r$ . If most pairs have met more rarely than  $r$  times at that point, the ILP might have a large number of optimal solutions, between which the ILP has no preference. To mitigate this problem, we test a variant of SIMAPPROX called SIMAPPROX+, which spends an extra 30 seconds after each ILP call to break ties in favor of partitions with better objective  $\hat{g}^{1/2}$ . As shown in the figure, SIMAPPROX+ gets substantially closer to the performance of the best greedy algorithms. While such variants of the simultaneous-approximation algorithm might have value for highly constrained table allocation problems or for large numbers of sessions, greedily optimizing  $\hat{g}^{1/2}$  or  $\hat{h}$  seems more worthwhile on the practical instances we study.

## 5.7 Discussion

As the last section shows, our algorithms produce schedules that excel in terms of the objective chosen by the practitioners, as well as in terms of the generalized objectives we introduced. The fundamental research problem, however — optimizing the group assignment in a way that increases the quality of deliberation — remains wide open and will require a multi-faceted approach. According to a handbook for assembly organizers, mixing groups up has a whole range of benefits: it helps assembly members “find common ground across the *whole* diverse group” (emphasis added), avoids situations where they “form cliques,” breaks up unproductive group dynamics, and overall “keeps things energised” [nU18]. Not only might each of these benefits suggest a different schedule, but predicting how well a schedule promotes each of these effects is also an open question. We believe that an approach combining optimization, behavioral research, and dynamic models of deliberation [CD20; FGMS17] can substantially support citizens’ assemblies and, by extension, democratic innovation.

[nU18] newDemocracy Foundation and United Nations Democracy Fund (2018): *Enabling National Initiatives to Take Democracy Beyond Elections*.

[CD20] Chung and Duggan (2020): *A Formal Theory of Democratic Deliberation*.

[FGMS17] Fain et al. (2017): *Sequential Deliberation for Social Choice*.

**PROPOSALS FOR OTHER ASPECTS OF DEMOCRACY:  
LIQUID DELEGATION AND APPORTIONMENT**

# Avoiding the Concentration of Power in Liquid Democracy

# 6

## 6.1 Introduction

Moving on from citizens' assemblies, this chapter studies another democratic innovation, *liquid democracy*. Like direct democracy, liquid democracy allows agents to vote on every issue by themselves. Alternatively, however, agents may delegate their vote, i.e., entrust it to any other agent who then votes on their behalf. Delegations are transitive; for example, if agents 2 and 3 delegate their votes to 1, and agent 4 delegates to 3, then agent 1 would vote with the weight of all four agents (themselves included). Just like representative democracy, this system allows for separation of labor, but provides for stronger accountability: each delegator is connected to their transitive delegate by a path of personal trust relationships, and each delegator on this path can withdraw their delegation at any time if they disagree with their delegate's choices.

Although the roots of liquid democracy can be traced back to the work of Miller [Mil69], it is only in recent years that it has gained recognition among practitioners. Most prominently, the German Pirate Party adopted the platform *LiquidFeedback* for internal decision-making in 2010. At the highest point, their installation counted more than 10 000 active users [KKH+15]. More recently, two parties — the Net Party in Argentina, and Flux in Australia — have run in national elections on the promise that their elected representatives would vote according to decisions made via their respective liquid-democracy-based systems. Although neither party was able to win any seats in parliament, their bids enhanced the promise and appeal of liquid democracy.

However, these real-world implementations also exposed a weakness in the liquid democracy approach: Certain individuals, the so-called *super-voters*, seem to amass enormous weight, whereas most agents do not receive any delegations. In the case of the Pirate Party, this phenomenon is illustrated by an article in *Der Spiegel* [Bec12], according to which one particular super-voter's "vote was like a decree," although he held no office in the party. As Kling et al. [KKH+15] describe, super-voters were so controversial that "the democratic nature of the system was questioned, and many users became inactive." Besides the negative impact of super-voters on perceived legitimacy, super-voters might also be more exposed to bribing. Although delegators can retract their delegations as soon as they become aware of suspicious voting behavior, serious damage might be done in the meantime. Furthermore, if super-voters jointly have sufficient power, they might find it more efficient to organize majorities through deals between super-voters behind closed doors, rather than to try to win a broad majority through public discourse. Finally, recent work by Kahng et al. [KMP18] indicates that, even if delegations go only to more competent agents, a high concentration of power might still be harmful for social welfare, by neutralizing benefits corresponding to the Condorcet Jury Theorem.

6.1 Introduction . . . . .	65
6.1.1 Approach and Results . . . . .	66
6.1.2 Related Work . . . . .	67
6.2 Algorithmic Results . . . . .	68
6.2.1 Problem Statement . . . . .	69
6.2.2 Confluent Flow . . . . .	69
6.2.3 Algorithms . . . . .	71
6.2.4 Hardness of Approximation . . . . .	72
6.3 Probabilistic Results . . . . .	75
6.3.1 Preferential Delegation . . . . .	76
6.3.2 Single Delegation . . . . .	77
6.3.3 Double Delegation . . . . .	80
6.4 Empirical Results . . . . .	85
6.4.1 Multiple vs. Single Delegations . . . . .	85
6.4.2 Evaluating Mechanisms . . . . .	86
6.5 Discussion . . . . .	90

[Mil69] Miller (1969): *A Program for Direct and Proxy Voting in the Legislative Process.*

[KKH+15] Kling et al. (2015): *Voting Behaviour and Power in Online Democracy.*

[Bec12] Becker (2012): *Web Platform Makes Professor Most Powerful Pirate.*

[KMP18] Kahng et al. (2018): *Liquid Democracy.*

While all these concerns suggest that the weight of super-voters should be limited, the exact metric to optimize for varies between them and is often not even clearly defined. For the purposes of this chapter, we choose to minimize the weight of the heaviest voter. As is evident in the Spiegel article, the weight of individual voters plays a direct role in the perception of super-voters. But even beyond that, we are confident that minimizing this measure will lead to substantial improvements across all presented concerns.

Just how can the maximum weight be reduced? One approach might be to restrict the power of delegation by imposing caps on the weight. However, as argued by Behrens et al. [BKNS14], delegation is always possible by coordinating outside of the system and copying the desired delegate's ballot. Pushing delegations outside of the system would not alleviate the problem of super-voters, just reduce transparency. Therefore, we instead adopt a voluntary approach: If agents are considering multiple potential delegates, all of whom they trust, they are encouraged to leave the decision for one of them to a centralized mechanism. With the goal of avoiding high-weight agents in mind, our research challenge is twofold:

*First, investigate the algorithmic problem of selecting delegations to minimize the maximum weight of any agent, and, second, show that allowing multiple delegation options does indeed provide a significant reduction in the maximum weight compared to the status quo.*

### 6.1.1 Our Approach and Results

We formally define our problem in Section 6.2. In addition to minimizing the maximum weight of any voter, we specify how to deal with delegators whose vote cannot possibly reach any voter. In general, our problem is closely related to minimizing congestion for confluent flow as studied by Chen et al. [CKL+07]. Not only does this connection suggest an optimal algorithm based on mixed integer linear programming, but we also get a polynomial-time  $(1 + \ln |V|)$ -approximation algorithm, where  $V$  is the set of voters. In addition, we show that approximating our problem to within a factor of  $\frac{1}{2} \log_2 |V|$  is NP-hard.

In Section 6.3, to evaluate the benefits of allowing multiple delegations, we propose a probabilistic model for delegation behavior — inspired by the well-known *preferential attachment* model [BA99] — in which we add agents successively. With a certain probability  $d$ , a new agent delegates; otherwise, they vote themselves. If the agent delegates, they choose  $k$  many delegation options among the previously inserted agents. A third parameter  $\gamma$  controls the bias of this selection towards agents who already receive many delegations. Assuming  $\gamma = 0$ , i.e., that the choice of delegates is unbiased, we prove that allowing two choices per delegator ( $k = 2$ ) asymptotically leads to dramatically lower maximum weight than classical liquid democracy ( $k = 1$ ). In the latter case, with high probability, the maximum weight is at least  $\Omega(t^\beta)$  for some  $\beta > 0$ , whereas the maximum weight in the former case is only  $O(\log \log t)$  with high probability, where  $t$  denotes simultaneously the time step of the process and the number of agents. Our analysis draws on a phenomenon called the *power of choice* that can be observed in many different load balancing models. In fact, even a greedy mechanism that, as agents arrive, myopically selects the delegation option whose transitive delegate has the

[BKNS14] Behrens et al. (2014): *The Principles of LiquidFeedback*.

[CKL+07] Chen et al. (2007): *(Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows*.

[BA99] Barabási and Albert (1999): *Emergence of Scaling in Random Networks*.

least weight so far exhibits this asymptotic behavior, which upper-bounds the maximum weight for optimal resolution.

In [Section 6.4](#), we complement our theoretical findings with empirical results. Our simulations demonstrate that our approach continues to outperform classical preferential attachment for higher values of  $\gamma$ . We also show that the most substantial improvements come from increasing  $k$  from one to two, i.e., that increasing  $k$  even further only slightly reduces the maximum weight. We continue to see these improvements in terms of maximum weight even if just some fraction of delegators give two options while the others specify a single delegate. Finally, we compare the optimal maximum weight with the maximum weight produced by the approximation algorithm and greedy heuristics.

### 6.1.2 Related Work

Kling et al. [[KKH+15](#)] conduct an empirical investigation of the existence and influence of super-voters. The analysis is based on daily data dumps, from 2010 until 2013, of the German Pirate Party installation of LiquidFeedback. As noted above, Kling et al. find that super-voters exist, and have considerable power. The results do suggest that super-voters behave responsibly, as they “do not fully act on their power to change the outcome of votes, and they vote in favour of proposals with the majority of voters in many cases.” Of course, this does not contradict the idea that a balanced distribution of power would be desirable.

[[KKH+15](#)] Kling et al. (2015): *Voting Behaviour and Power in Online Democracy*.

In recent years, there has been an increasing number of theoretical analyses of liquid democracy. In the field of political theory, Blum and Zuber [[BZ16](#)] give a normative justification of liquid democracy. They consider two accounts of democracy, which differ in the stated goal of a democratic system. In the *epistemic* framework, the success of a democratic system should lead to good decisions with respect to some objective notion of quality, whereas, in the *egalitarian* framework, a democratic system should allow each individual to impose their particular interests to the same degree. Blum and Zuber conclude that liquid democracy improves upon purely representative democracy with respect to both metrics. They see unequal voting weights as problematic and suggest public deliberation before a vote to attenuate this problem.

[[BZ16](#)] Blum and Zuber (2016): *Liquid Democracy*.

In the spirit of the egalitarian framework, Green-Armytage [[Gre15](#)] justifies liquid democracy in a spatial model of political preferences similar to facility placement. When an agent has incomplete information about a topic, transitive delegations can help to express the agent’s preferences more accurately by harnessing the expertise of like-minded, more qualified agents.

[[Gre15](#)] Green-Armytage (2015): *Direct Voting and Proxy Voting*.

While delegations to more qualified agents can lead to better decisions in certain situations, Kahng et al. [[KMP18](#)] show that such delegations do not always improve upon the baseline of direct democracy. They assume an epistemic model, where a single binary issue has one “correct” and one “incorrect” outcome. Voters are modeled as biased coins that each choose the correct outcome with an individually assigned probability, or *competence level*. Agents can either vote themselves or delegate to one of their neighbors in a social network. Assuming that agents only delegate to more qualified agents, Kahng et al. show that no local delegation policy (in which an agent’s delegate is selected based on the

[[KMP18](#)] Kahng et al. (2018): *Liquid Democracy*.

agent’s competence level and that of their neighbors) universally increases the probability of making the right decision. In a very similar model, Caragiannis and Micha [CM19] point out further situations in which local delegation mechanisms fail to promote the socially preferable choice. In contrast to these papers, our work fits better into the egalitarian approach. Furthermore, it is completely independent of the (strong) assumptions underlying the aforementioned results. In particular, our approach is agnostic to the final outcome of the voting process, does not assume access to competence information that would be inaccessible in practice, and is compatible with any number of alternatives and choice of voting rule used to aggregate votes. In other words, the goal is not to use liquid democracy to promote a particular outcome, but rather to adapt the process of liquid democracy so that more voices will be heard.

In this chapter, we consider a single delegation network. Other works allow agents to specify different delegations for multiple interconnected issues, where the binary preferences and outcomes are restricted to satisfy a propositional formula [CG17] or to correspond to binary comparisons in a ranking [BT18]. Both papers propose ways of reconciling contradictory choices made by different delegates.

We also highlight related work that considers models of network formation and influence attenuation in the context of liquid democracy. Bloembergen et al. [BGL19] introduce a game-theoretic model of delegation in order to study rational delegation behavior in liquid-democracy networks. In their model, delegation networks might be formed by a best-response dynamic or as Nash equilibria of a delegation game. Escoffier et al. [EGP19] study a similar delegation game with different incentives.

Boldi et al. [BBCV11] study a variant of liquid democracy in which a voter’s weight decreases by a discount factor every time their vote is transitively delegated, penalizing long delegation chains. They argue that this variant is more appropriate in online communities, where trust relationships are typically less deep than in the real world. While not intended as such, this variant of liquid democracy can also reduce the weight of super-voters, at least of those who receive most of their delegations indirectly. However, such a variant violates the principle of “one person, one vote” and incentivizes delegation outside of the system [BKNS14]. By contrast, our approach reduces the weight of super-voters while preserving each voter’s individual influence.

[CM19] Caragiannis and Micha (2019): *A Contribution to the Critique of Liquid Democracy*.

[CG17] Christoff and Grossi (2017): *Binary Voting with Delegable Proxy*.

[BT18] Brill and Talmon (2018): *Pairwise Liquid Democracy*.

[BGL19] Bloembergen et al. (2019): *On Rational Delegations in Liquid Democracy*.

[EGP19] Escoffier et al. (2019): *The Convergence of Iterative Delegations in Liquid Democracy in a Social Network*.

[BBCV11] Boldi et al. (2011): *Viscous Democracy for Social Networks*.

[BKNS14] Behrens et al. (2014): *The Principles of LiquidFeedback*.

## 6.2 Algorithmic Model and Results

Let us consider a delegative voting process where agents may specify multiple potential delegations. This gives rise to a directed graph, whose nodes represent agents and whose edges represent potential delegations. In the following, we will conflate nodes and the agents they represent. A distinguished subset of nodes corresponds to agents who have voted directly, the *voters*. Since voters forfeit the right to delegate, the voters are a subset of the sinks of the graph. We call all non-voter agents *delegators*.

Each agent has an inherent voting weight of 1. When the delegations will have been resolved, the weight of every agent will be the sum of weights of their

delegators plus their inherent weight. We aim to choose a delegation for every delegator in such a way that the maximum weight of any voter is minimized.

This task closely mirrors the problem of congestion minimization for confluent flow (with infinite edge capacity): There, a flow network is also a finite directed graph with a distinguished set of graph sinks, the *flow sinks*. Every node has a non-negative *demand*. If we assume unit demand, this demand is 1 for every node. Since the flow is confluent, for every non-sink node, the algorithm must pick exactly one outgoing edge, along which the flow is sent. Then, the *congestion* at a node  $n$  is the sum of congestions at all nodes who direct their flow to  $n$  plus the demand of  $n$ . The goal in congestion minimization is to minimize the maximum congestion at any flow sink.<sup>1</sup>

In spite of the similarity between confluent flow and resolving potential delegations, the two problems differ when a node has no path to a voter / flow sink. In confluent flow, the result would simply be that no flow exists. In our setting however, this situation can hardly be avoided. If, for example, several friends assign all of their potential delegations to each other, and if all of them rely on the others to vote, their weight cannot be delegated to any voter. Our mechanism cannot simply report failure as soon as a small group of voters behaves in an unexpected way. Thus, it must be allowed to leave these votes unused. At the same time, of course, our algorithm should not exploit this power to decrease the maximum weight, but must primarily maximize the number of utilized votes. We formalize these issues in the following section.

### 6.2.1 Problem Statement

All graphs  $G = (N, E)$  mentioned in this section will be finite and directed. Furthermore, they will be equipped with a set  $V$  of distinguished sinks in the graph. For the sake of brevity, these assumptions will be implicit in the notion “graph  $G$  with  $V$ ”.

Some of these graphs represent situations in which all delegations have already been resolved and in which each vote reaches a voter: We call a graph  $(N, E)$  with  $V$  a *delegation graph* if it is acyclic, its sinks are exactly the set  $V$ , and every other vertex has outdegree one. In such a graph, define the *weight*  $w(n)$  of a node  $n \in N$  as

$$w(n) := 1 + \sum_{(m,n) \in E} w(m).$$

This is well-defined because  $E$  is a well-founded relation on  $N$ .

Resolving the delegations of a graph  $G$  with  $V$  can now be described as the MINMAXWEIGHT problem: Among all delegation subgraphs  $(N', E')$  of  $G$  with voting vertices  $V$  of maximum  $|N'|$ , find one that minimizes the maximum weight of the voting vertices.

### 6.2.2 Connections to Confluent Flow

We recall definitions from the flow literature as used by Chen et al. [CKL+07]. We slightly simplify the exposition by assuming unit demand at every node.

1: The close connection between our problem and confluent flow immediately suggests a variant corresponding to splittable flow; we discuss this variant at length in Section 6.5.

[CKL+07] Chen et al. (2007): (Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows.

Given a graph  $(N, E)$  with  $V$ , a *flow* is a function  $f : E \rightarrow \mathbb{R}_{\geq 0}$ . For any node  $n$ , set  $in(n) := \sum_{(m,n) \in E} f(m, n)$  and  $out(n) := \sum_{(n,m) \in E} f(n, m)$ . At every node  $n \in N \setminus V$ , a flow must satisfy *flow conservation*:

$$out(n) = 1 + in(n).$$

Note that all nodes in  $V$  are sinks in the graph, and thus have no outflow. The congestion at any node  $n$  is defined as  $1 + in(n)$ . A flow is *confluent* if every node has at most one outgoing edge with positive flow. We define **MINMAXCONGESTION** as the problem of finding a confluent flow on a given graph such that the maximum congestion is minimized.

To relate the two presented problems, we need to refer to the parts of a graph  $(N, E)$  with  $V$  from which  $V$  is reachable: The *active nodes*  $active_V(N, E)$  are all  $n \in N$  such that there exists a path from  $n$  to a sink  $v \in V$  using edges in  $E$ . The *active subgraph* is the restriction of  $(N, E)$  to  $active_V(N, E)$ . In particular,  $V$  is part of this subgraph.

**Lemma 6.1** *Let  $G = (N, E)$  with  $V$  be a graph. Its delegation subgraphs  $(N', E')$  that maximize  $|N'|$  are exactly the delegation subgraphs with  $N' = active_V(N, E)$ . At least one such subgraph exists.*

*Proof.* First, we show that all nodes of a delegation subgraph are active. Indeed, consider any node  $n_1$  in the subgraph. By following outgoing edges, we obtain a sequence of nodes  $n_1 n_2 \dots$  such that  $n_i$  delegates to  $n_{i+1}$ . Since the graph is finite and acyclic, this sequence must end with a vertex  $n_j$  without outgoing edges. This must be a voter; thus,  $n_1$  is active.

Furthermore, there exists a delegation subgraph of  $(N, E)$  with nodes exactly  $active_V(N, E)$ . Indeed, the shortest-paths-to-set- $V$  forest (with edges pointed in the direction of the paths) on the active subgraph is a delegation graph.

By the first argument, all delegation subgraphs must be subgraphs of the active subgraph. By the second argument, to have the maximum number of nodes, they must include all nodes of this subgraph.  $\square$

**Lemma 6.2** *Let  $(N, E)$  with  $V$  be a graph and let  $f : E \rightarrow \mathbb{R}_{\geq 0}$  be a confluent flow (for unit demand). By eliminating all zero-flow edges from the graph, we obtain a delegation graph.*

*Proof.* We first claim that the resulting graph is acyclic. Indeed, for the sake of contradiction, suppose that there is a cycle including some node  $n$ . Consider the flow out of  $n$ , through the cycle and back into  $n$ . Since the flow is confluent, and thus the flow cannot split up, the demand can only increase from one node to the next. As a result,  $in(n) \geq out(n)$ . However, by flow conservation and unit demand,  $out(n) = in(n) + 1$ , which contradicts the previous statement.

Furthermore, the sinks of the graph are exactly  $V$ : By assumption, the nodes of  $V$  are sinks in the original graph, and thus in the resulting graph. For any other

node, flow conservation dictates that its outflow be at least its demand 1, thus every other node must have outgoing edges.

Finally, every node not in  $V$  must have outdegree 1. As detailed above, the outdegree must be at least 1. Because the flow was confluent, the outdegree cannot be greater.

As a result of these three properties, we have a delegation graph.  $\square$

**Lemma 6.3** *Let  $(N, E)$  with  $V$  be a graph in which all vertices are active, and let  $(N, E')$  be a delegation subgraph. Let  $f : E \rightarrow \mathbb{R}_{\geq 0}$  be defined such that, for every node  $n \in N \setminus V$  with (unique) outgoing edge  $e \in E'$ ,  $f(e) := w(n)$ . On all other edges  $e \in E \setminus E'$ , set  $f(e) := 0$ . Then,  $f$  is a confluent flow.*

*Proof.* For every non-sink, flow conservation holds by the definition of weight and flow. By construction, the flow must be confluent.  $\square$

### 6.2.3 Algorithms

The observations made above allow us to apply algorithms — even approximation algorithms — for MINMAXCONGESTION to our MINMAXWEIGHT problem; that is, we can reduce the latter problem to the former.

**Theorem 6.4** *Let  $\mathcal{A}$  be an algorithm for MINMAXCONGESTION with approximation ratio  $c \geq 1$ . Let  $\mathcal{A}'$  be an algorithm that, given  $(N, E)$  with  $V$ , runs  $\mathcal{A}$  on the active subgraph, and translates the result into a delegation subgraph by eliminating all zero-flow edges. Then  $\mathcal{A}'$  is a  $c$ -approximation algorithm for MINMAXWEIGHT.*

*Proof.* By Lemma 6.1, removing inactive parts of the graph does not change the solutions to MINMAXWEIGHT, so we can assume without loss of generality that all vertices in the given graph are active.

Suppose that the optimal solution for MINMAXCONGESTION on the given instance has maximum congestion  $\alpha$ . By Lemma 6.2, it can be translated into a solution for MINMAXWEIGHT with maximum weight  $\alpha$ . By Lemma 6.3, the latter instance has no solution with maximum weight less than  $\alpha$ , otherwise it could be used to construct a confluent flow with the same maximum congestion. It follows that the optimal solution to the given MINMAXWEIGHT instance has maximum weight  $\alpha$ .

Now,  $\mathcal{A}$  returns a confluent flow with maximum congestion at most  $c \cdot \alpha$ . Using Lemma 6.2,  $\mathcal{A}'$  constructs a solution to MINMAXWEIGHT with maximum weight at most  $c \cdot \alpha$ . Therefore,  $\mathcal{A}'$  is a  $c$ -approximation algorithm.  $\square$

Note that Theorem 6.4 works for  $c = 1$ , i.e., even for exact algorithms. Therefore, it is possible to solve MINMAXWEIGHT by adapting any exact algorithm for MINMAXFLOW. In particular, congestion minimization for confluent flow can be expressed as a mixed integer linear program (MILP).

To stress the connection to MINMAXWEIGHT, denote the congestion at a voter  $i$  by  $w(i)$ . For each potential delegation  $(u, v)$ ,  $f(u, v)$  gives the amount of flow between  $u$  and  $v$ . This flow must be nonnegative (6.2) and satisfy flow conservation (6.3). Congestion is defined in Equation (6.4). To minimize maximum congestion, we introduce a variable  $z$  that is higher than the congestion of any voter (6.5), and minimize  $z$  (6.1).

So far, we have described a Linear Program for optimizing splittable flow. To restrict the solutions to confluent flow, we must enforce an ‘all-or-nothing’ constraint on outflow from any node, i.e. at most one outgoing edge per node can have positive flow. We express this using a convex-hull reformulation. We introduce a binary variable  $x_{u,v}$  for each edge Equation (6.6), and set the sum of binary variables for all outgoing edges of a node to 1 (6.7). If  $M$  is a constant larger than the maximum possible flow, we can then bound  $f(u, v) \leq M x_{u,v}$  (6.8) to have at most one positive outflow per node.

The final MILP is thus

$$\text{minimize } z \quad (6.1)$$

$$\text{subject to } f(m, n) \geq 0 \quad \forall (m, n) \in E, \quad (6.2)$$

$$\sum_{(n,m) \in E} f(n, m) = 1 + \sum_{(m,n) \in E} f(m, n) \quad \forall n \in N \setminus V, \quad (6.3)$$

$$w(v) = 1 + \sum_{(n,v) \in E} f(n, v) \quad \forall v \in V, \quad (6.4)$$

$$z \geq w(v) \quad \forall v \in V, \quad (6.5)$$

$$x_{m,n} \in \{0, 1\} \quad \forall (m, n) \in E, \quad (6.6)$$

$$\sum_{(n,m) \in E} x_{n,m} = 1 \quad \forall n \in N \setminus V, \quad (6.7)$$

$$f(m, n) \leq M \cdot x_{m,n} \quad \forall (m, n) \in E. \quad (6.8)$$

Since the foregoing algorithm is based on solving an NP-hard problem, it might be too inefficient for typical use cases of liquid democracy with many participating agents. Fortunately, it might be acceptable to settle for a slightly non-optimal maximum weight if this decreases computational cost. To our knowledge, the best polynomial approximation algorithm for MINMAXCONGESTION is due to Chen et al. [CKL+07] and achieves an approximation ratio of  $1 + \ln |V|$ . Their algorithm starts by computing the optimal solution to the splittable-flow version of the problem, by solving a linear program. The heart of their algorithm is a non-trivial, deterministic rounding mechanism. This scheme drastically outperforms the natural, randomized rounding scheme, which leads to an approximation ratio of  $\Omega(|N|^{1/4})$  with arbitrarily high probability [CRSo6].

[CKL+07] Chen et al. (2007): *(Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows*.

[CRSo6] Chen et al. (2006): *Meet and Merge*.

## 6.2.4 Hardness of Approximation

In this section, we demonstrate the NP-hardness of approximating the MINMAXWEIGHT problem to within a factor of  $\frac{1}{2} \log_2 |V|$ . On the one hand, this justifies the absence of an exact polynomial-time algorithm. On the other hand,

this shows that the approximation algorithm is optimal up to a multiplicative constant.

**Theorem 6.5** *It is NP-hard to approximate the MINMAXWEIGHT problem to a factor of  $\frac{1}{2} \log_2 |V|$ , even when each node has outdegree at most 2.*

Not surprisingly, we derive hardness via a reduction from MINMAXCONGESTION, i.e., a reduction in the opposite direction from the one given in [Theorem 6.4](#). As shown by Chen et al. [\[CKL+07\]](#), approximating MINMAXCONGESTION to within a factor of  $\frac{1}{2} \log_2 |V|$  is NP-hard. However, in our case, nodes have unit demands. Moreover, we are specifically interested in the case where each node has outdegree at most 2, as in practice we expect outdegrees to be very small, and this case plays a special role in [Section 6.3](#).

We begin with a lemma that slightly strengthens a hardness result by Fortune et al. [\[FHW80\]](#):

**Lemma 6.6** *Let  $G$  be a directed graph in which all vertices have an outdegree of at most 2. Given vertices  $s_1, s_2, t_1, t_2$ , it is NP-hard to decide whether there exist vertex-disjoint paths from  $s_1$  to  $t_1$  and from  $s_2$  to  $t_2$ .*

*Proof.* Without the restriction on the outdegree, the problem is NP-hard [\[FHW80\]](#). We reduce the general case to our special case.

Let  $G'$  be an arbitrary directed graph; let  $s'_1, s'_2, t'_1, t'_2$  be distinguished vertices. To restrict the outdegree, replace each node  $n$  with outdegree  $d$  by a binary arborescence (directed binary tree with edges facing away from the root) with  $d$  sinks. All incoming edges into  $n$  are redirected towards the root of the arborescence; outgoing edges from  $n$  instead start from the different leaves of the arborescence. Call the new graph  $G$ , and let  $s_1, s_2, t_1, t_2$  refer to the roots of the arborescences replacing  $s'_1, s'_2, t'_1, t'_2$ , respectively.

Clearly, our modifications to  $G'$  can be carried out in polynomial time. It remains to show that there are vertex-disjoint paths from  $s_1$  to  $t_1$  and from  $s_2$  to  $t_2$  in  $G$  iff there are vertex-disjoint paths from  $s'_1$  to  $t'_1$  and from  $s'_2$  to  $t'_2$  in  $G'$ .

If there are disjoint paths in  $G'$ , we can translate these paths into  $G$  by visiting the arborescences corresponding to the nodes on the original path one after another. Since both paths visit disjoint arborescences, the new paths must be disjoint.

Suppose now that there are disjoint paths in  $G$ . Translate the paths into  $G'$  by visiting the nodes corresponding to the sequence of visited arborescences. Since each arborescence can only be entered via its root, disjointness of the paths in  $G$  implies disjointness of the translated paths in  $G'$ .  $\square$

Now, we can strengthen the hardness of approximation for MINMAXCONGESTION by Chen et al. [\[CKL+07\]](#). We believe the lemma is of independent interest, as it shows a surprising separation between the case of outdegree 1 (where the problem is moot) and outdegree 2, and that the asymptotically optimal approximation ratio is independent of degree. But it also allows us to prove [Theorem 6.5](#) almost directly.

[\[CKL+07\]](#) Chen et al. (2007): *(Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows.*

[\[FHW80\]](#) Fortune et al. (1980): *The Directed Subgraph Homeomorphism Problem.*

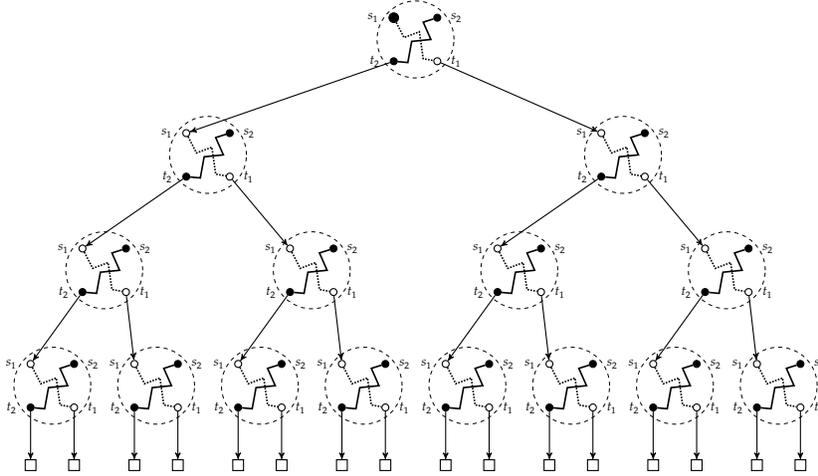


Figure 6.1: Auxiliary network generated from  $G$ , here for  $k = 16$ . Recreation of [CKL+07, Fig. 2].

**Lemma 6.7** *It is NP-hard to approximate the MINMAXCONGESTION problem to a factor of  $\frac{1}{2} \log_2 k$ , where  $k$  is the number of sinks, even when each node has unit demand and outdegree at most 2.*

*Proof of Lemma 6.7.* We adapt the proof of Theorem 1 of Chen et al. [CKL+07].

Let  $G = (V, E), s_1, s_2, t_1, t_2$  be given as in Lemma 6.6. Without loss of generality,  $G$  only contains nodes from which  $t_1$  or  $t_2$  is reachable,  $t_1$  and  $t_2$  are sinks and all four vertices are distinct. Let  $\ell = \lceil \log_2 |V| \rceil$  and  $k = 2^\ell$ . Build the same auxiliary network as that built by Chen et al. [CKL+07], which consists of a binary arborescence whose  $k - 1$  nodes are copies of  $G$ . The construction is illustrated in Figure 6.1. For more details, refer to their paper.

For ease of exposition, we describe our reduction as returning a flow network with polynomially-bounded positive integer demands. Implicitly, the described network is subsequently translated into one with unary demand; to express a demand of  $d$  at a node  $n$  in our unit-demand setting, add  $d - 1$  fresh nodes with a single outgoing edge to  $n$ .

Denote the number of nodes in the network by  $\phi := (k - 1) \cdot |V| + k$ , and set  $\Phi := \ell \cdot \phi + 1$ . In the proof by Chen et al. [CKL+07], every copy of  $s_2$  and  $t_2$  has demand 1, the copy of  $s_1$  at the root has demand 2, and all other nodes have demand 0. Instead, we give these nodes demands of  $\Phi$ ,  $2\Phi$  and 1, respectively. Note that the size of the generated network<sup>2</sup> is polynomial in the size of  $G$  and that the outdegree of each node is at most 2. From every node, one of the sinks  $S$  displayed as rectangles in Figure 6.1 is reachable. Since the minimum-distance-to- $S$  spanning forest describes a flow, a flow in the network exists.

Suppose that  $G$  contains vertex-disjoint paths  $P_1$  from  $s_1$  to  $t_1$  and  $P_2$  from  $s_2$  to  $t_2$ . In each copy of  $G$  in the network, route the flow along these paths. We can complete the confluent flow inside of this copy in such a way that the demand of every node is routed to  $t_1$  or  $t_2$ : By assumption, each of the nodes can reach one of these two path endpoints. Iterate over all nodes in order of ascending distance to the closest endpoint and make sure that their flow is routed to an endpoint. For the endpoints themselves, there is nothing to do. For positive distance, a node might be part of a path and thus already connected to an endpoint. Else, look at

[CKL+07] Chen et al. (2007): (Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows.

<sup>2</sup>: Even after unfolding our non-unitary-demand nodes.

its successor in a shortest path to an endpoint. By the induction hypothesis, all flow from this successor is routed to an endpoint, so route the node's flow to this successor. If we also use the edges between copies of  $G$  and between the copies and the sinks, we obtain a confluent flow. Each sink except for the rightmost one can only collect the demand of two nodes with demand  $\Phi$  plus a number of nodes with demand 1. The rightmost sink collects the demand from the single node with demand  $2\Phi$  plus some unitary demands. Thus, the congestion of the system can be at most  $2\Phi + \phi$ .

Now, consider the case in which  $G$  does not have such vertex-disjoint paths. In every confluent flow and in every copy, there are three options:

- ▶ the flow from  $s_1$  flows to  $t_2$  and the flow from  $s_2$  flows to  $t_1$ ,
- ▶ the flow from  $s_1$  and  $s_2$  flows to  $t_1$ , or
- ▶ the flow from  $s_1$  and  $s_2$  flows to  $t_2$ .

In each case, the flow coming in through  $s_1$  is joined by additional demand of at least  $\Phi$ . Consider the path from the copy of  $s_1$  at the root to a sink. By a simple inductive argument, the congestion at the endpoint of the  $i$ th copy of  $G$  on this path is at least  $(i + 1) \cdot \Phi$ . Thus, the total congestion at the sink must be at least  $(\ell + 1) \cdot \Phi$ . The lemma now follows from the fact that

$$\frac{\log_2 k}{2} (2\Phi + \phi) = \frac{\ell}{2} (2\Phi + \phi) < (\ell + 1) \cdot \Phi. \quad \square$$

*Proof of Theorem 6.5.* We reduce (gap) MINMAXCONGESTION with unit demand and outdegree at most 2 to (gap) MINMAXWEIGHT with outdegree at most 2. First, we claim that if there are inactive nodes, there is no confluent flow. Indeed, let  $n_1$  be an inactive node. For the sake of contradiction, suppose that there exists a flow  $f$ . Follow the positive flow to obtain a sequence  $n_1 n_2 \dots$ . By definition, none of the nodes reachable from  $n_1$  can be a voter. Since, by flow conservation and unit demand, each node must delegate, the sequence must be infinite. As detailed in the proof of Lemma 6.2, a confluent flow with unit demand cannot contain cycles. Thus, the sequence contains infinitely many different nodes, which contradicts the finiteness of  $G$ .

Therefore, we can assume without loss of generality that in the given instance of MINMAXCONGESTION, all nodes are active (as the problem is still NP-hard). The reduction creates an instance of MINMAXWEIGHT that has the same graph as the given instance of MINMAXCONGESTION. Using an argument analogous to the proof of Theorem 6.4 (reversing the roles of Lemma 6.2 and Lemma 6.3 in its proof), we see that this is a strict approximation-preserving reduction.  $\square$

### 6.3 Probabilistic Model and Results

Our generalization of liquid democracy to multiple potential delegations aims to decrease the concentration of weight. Accordingly, the success of our approach should be measured by its effect on the maximum weight in real elections. Since, at this time, we do not know of any available datasets,<sup>3</sup> we instead propose a probabilistic model for delegation behavior, which can serve as a credible proxy.

<sup>3</sup>: There is one relevant dataset that we know of, which was analyzed by Kling et al. [KKH+15]. However, due to stringent privacy constraints, the data privacy officer of the German Pirate Party was unable to share this dataset with us.

Our model builds on the well-known preferential attachment model, which generates graphs possessing typical properties of social networks.

The evaluation of our approach will be twofold: In Sections 6.3.2 and 6.3.3, for a certain choice of parameters in our model, we establish a striking separation between traditional liquid democracy and our system. In the former case, the maximum weight at time  $t$  is  $\Omega(t^\beta)$  for a constant  $\beta$  with high probability, whereas in the latter case, it is in  $O(\log \log t)$  with high probability, even if each delegator only suggests two options. For other parameter settings, we empirically corroborate the benefits of our approach in Section 6.4.

### 6.3.1 The Preferential Delegation Model

Many real-world social networks have degree distributions that follow a power law [KNT10; New01]. Additionally, in their empirical study, Kling et al. [KKH+15] observed that the weight of voters in the German Pirate Party was “power law-like” and that the graph had a very unequal indegree distribution. In order to meld the previous two observations in our liquid democracy delegation graphs, we adapt a standard preferential attachment model [BA99] for this specific setting. At a high level, our *preferential delegation* model is characterized by three parameters:  $0 < d < 1$ , the probability of delegation;  $k \geq 1$ , the number of delegation options from each delegator; and  $\gamma \geq 0$ , an exponent that governs the probability of delegating to nodes based on current weight.

At time  $t = 1$ , we have a single node representing a single voter. In each subsequent time step, we add a node for agent  $i$  and flip a biased coin to determine the agent’s delegation behavior. With probability  $d$ , they delegate to other agents. Else, they vote independently. If  $i$  does not delegate, their node has no outgoing edges. Otherwise, add edges to  $k$  many independently selected, previously inserted nodes, where the probability of choosing node  $j$  is proportional to  $(\text{indegree}(j) + 1)^\gamma$ . Note that this model might generate multiple edges between the same pair of nodes, and that all sinks are voters. Figure 6.2 shows example graphs for different settings of  $\gamma$ .

In the case of  $\gamma = 0$ , which we term *uniform delegation*, a delegator is equally likely to attach to any previously inserted node. Already in this case, a “rich-get-richer” phenomenon can be observed, i.e., voters at the end of large networks of potential delegations will likely see their network grow even more. Indeed, a larger network of delegations is more likely to attract new delegators. In traditional liquid democracy, where  $k = 1$  and all potential delegations will be realized, this explains the emergence of super-voters with excessive weight observed by Kling et al. [KKH+15]. We aim to show that for  $k \geq 2$ , the resolution of potential delegations can strongly outweigh these effects. In this, we profit from an effect known as the “power of two choices” in load balancing described by Azar et al. [ABKU94].

For  $\gamma > 0$ , the “rich-get-richer” phenomenon additionally appears at the degrees of nodes. Since the number of received potential delegations is a proxy for an agent’s competence and visibility, new agents are more likely to attach to agents with high indegree. In total, this is likely to further strengthen the inherent inequality between voters. For increasing  $\gamma$ , the graph becomes increasingly flat,

[KNT10] Kumar et al. (2010): *Structure and Evolution of Online Social Networks*.

[New01] Newman (2001): *Clustering and Preferential Attachment in Growing Networks*.

[KKH+15] Kling et al. (2015): *Voting Behaviour and Power in Online Democracy*.

[BA99] Barabási and Albert (1999): *Emergence of Scaling in Random Networks*.

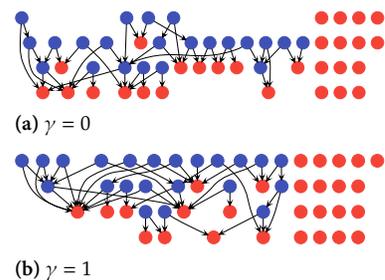


Figure 6.2: Example graphs generated by the preferential delegation model for  $k = 2$  and  $d = 0.5$ .

[ABKU94] Azar et al. (1994): *Balanced Allocations*.

as a few super-voters receive nearly all delegations. This matches observations from the LiquidFeedback dataset [KKH+15] that “the delegation network is slowly becoming less like a friendship network, and more like a bipartite networks of super-voters connected to normal voters.” The special case of  $\gamma = 1$  corresponds to preferential attachment as described by Barabási and Albert [BA99].

The most significant difference we expect to see between graphs generated by the preferential delegation model and real delegation graphs is the assumption that agents always delegate to more senior agents. In particular, this causes generated graphs to be acyclic, which need not be the case in practice. It does seem plausible that the majority of delegations goes to agents with more experience on the platform. Even if this assumption should not hold, there is a second interpretation of our process if we assume — as do Kahng et al. [KMP18] — that agents can be ranked by competence and only delegate to more competent agents. Then, we can think of the agents as being inserted in decreasing order of competence. When a delegator chooses more competent agents to delegate to, the delegator’s choice would still be biased towards agents with high indegree, which is a proxy for popularity.

It may be useful to note that the MINMAXWEIGHT approach based on confluent flow does not require the underlying delegation graph to be acyclic, as the objective tries to minimize the maximum weight of any voter over all possible delegation choices that maximize the total number of utilized votes. In this sense, unavoidable cycles result in lost voting power.

In our theoretical results, we focus on the cases of  $k = 1$  and  $k = 2$ , and assume  $\gamma = 0$  to make the analysis tractable. The parameter  $d$  can be chosen freely between 0 and 1. Note that our upper bound for  $k = 2$  directly translates into an upper bound for larger  $k$ , since the resolution mechanism always has the option of ignoring all outgoing edges except for the first two. Therefore, to understand the effect of multiple delegation options, we can restrict our attention to  $k = 2$ . This crucially relies on  $\gamma = 0$ , where potential delegations do not influence the probabilities of choosing future potential delegations. Based on related results by Malyshkin and Paquette [MP15], it seems unlikely that increasing  $k$  beyond 2 will reduce the maximum weight by more than a constant factor.

### 6.3.2 Lower Bounds for Single Delegation ( $k = 1, \gamma = 0$ )

As mentioned above, we first assume uniform delegation and a single delegation option per delegator, and derive a lower bound on the maximum weight. To state our results rigorously, we say that a sequence  $(\mathcal{E}_m)_m$  of events happens *with high probability* if  $\mathbb{P}[\mathcal{E}_m] \rightarrow 1$  for  $m \rightarrow \infty$ . Since the parameter going to infinity is clear from the context, we omit it.

**Theorem 6.8** *In the preferential delegation model with  $k = 1, \gamma = 0$ , and  $d \in (0, 1)$ , with high probability, the maximum weight of any voter at time  $t$  is in  $\Omega(t^\beta)$ , where  $\beta > 0$  is a constant that depends only on  $d$ .*

*Proof.* It suffices to show that, with high probability, there exists a voter at every time  $t$  whose weight is bounded from below by a function in  $\Omega(t^\beta)$ .

[KKH+15] Kling et al. (2015): *Voting Behaviour and Power in Online Democracy*.

[BA99] Barabási and Albert (1999): *Emergence of Scaling in Random Networks*.

[KMP18] Kahng et al. (2018): *Liquid Democracy*.

[MP15] Malyshkin and Paquette (2015): *The Power of Choice over Preferential Attachment*.

For ease of exposition, we pretend that  $i_{max} := \log_2 \frac{t}{\ln t}$  is an integer.<sup>4</sup> We divide the  $t$  agents into  $i_{max} + 1$  blocks  $B_0, \dots, B_{i_{max}}$ . The first block  $B_0$  contains agents 1 to  $\tau := \ln t$ , and every subsequent block  $B_i$  contains agents  $(\tau 2^{i-1}, \tau 2^i]$ .

4: The same argument works for  $i_{max} := \lfloor \log_2 \frac{t}{\ln t} \rfloor$  if we appropriately bound the term.

We keep track of the total weight  $S_i$  of all voters in  $B_0$  after the entirety of block  $B_i$  has been added. Furthermore, we define an event  $X_i$  saying that a high enough number of agents in block  $B_i$  transitively delegate into  $B_0$ . If all  $X_i$  hold,  $S_{i_{max}}$  scales like a power function. Then, we show that, as  $t$  increases, the probability of any  $X_i$  failing goes to zero. Thus, our bound on  $S_{i_{max}}$  holds with high probability. The total weight of  $B_0$  and the weight of the maximum-weight voter in  $B_0$  can differ by at most a factor of  $\tau$ , which is logarithmic in  $t$ . Thus, with high probability, there is a voter in  $B_0$  whose weight is a power function.

In more detail, let  $\varepsilon := \frac{1}{2}$  and let  $d' := (1 - \varepsilon)d = \frac{d}{2}$ . For each  $i \geq 0$ , let  $Y_i$  denote the number of votes from block  $i$  transitively going into  $B_0$ . Clearly,  $S_i = \sum_{j=0}^i Y_j$ . For  $i > 0$ , let  $X_i$  denote the event that

$$Y_i > d' \frac{\tau \left(1 + \frac{d'}{2}\right)^{i-1}}{2}.$$

**Bounding the Expectation of  $Y_i$**  We first prove by induction on  $i$  that, if  $X_1$  through  $X_i$  hold, then

$$S_i \geq \tau \left(1 + \frac{d'}{2}\right)^i. \quad (6.9)$$

For  $i = 0$ ,  $S_0 = \tau$  and the claim holds. For  $i > 0$ , by the induction hypothesis,  $S_{i-1} \geq \tau \left(1 + \frac{d'}{2}\right)^{i-1}$ . By the assumption  $X_i$ ,

$$Y_i > d' \frac{\tau \left(1 + \frac{d'}{2}\right)^{i-1}}{2}.$$

Thus,

$$\begin{aligned} S_i &= S_{i-1} + Y_i \geq \tau \left(1 + \frac{d'}{2}\right)^{i-1} + d' \frac{\tau \left(1 + \frac{d'}{2}\right)^{i-1}}{2} \\ &= \tau \left(1 + \frac{d'}{2}\right)^{i-1} \left(1 + \frac{d'}{2}\right) = \tau \left(1 + \frac{d'}{2}\right)^i. \end{aligned}$$

This concludes the induction and establishes [Equation \(6.9\)](#).

Now, for any agent  $j$  in  $B_i$ , the probability of transitively delegating into  $B_0$  is

$$d \frac{\sum_{v \in V \cap B_0} w_{j-1}(v)}{j-1} \geq d \frac{S_{i-1}}{\tau 2^i}.$$

Conditioned on  $X_1, \dots, X_{i-1}$ , we can thus lower-bound  $Y_i$  by a binomial variable

$\text{Bin}\left(\tau 2^{i-1}, d \frac{S_{i-1}}{\tau 2^i}\right)$  to obtain

$$\mathbb{E}[Y_i \mid X_1, \dots, X_{i-1}] \geq \tau 2^{i-1} d \frac{S_{i-1}}{\tau 2^i} = d \frac{S_{i-1}}{2} \geq d \frac{\tau \left(1 + \frac{d'}{2}\right)^{i-1}}{2}.$$

Denoting the right hand side by

$$\mu := d \frac{\tau \left(1 + \frac{d'}{2}\right)^{i-1}}{2},$$

note that  $X_i$  holds if  $Y_i > (1 - \varepsilon) \mu$ .

**Failure Probability Goes to 0** Now, we must show that, with high probability, all  $X_i$  hold. By underapproximating the probability of delegation by a binomial random variable as before and by using a Chernoff bound, we have for all  $i > 0$

$$\mathbb{P}[X_i \mid X_1, \dots, X_{i-1}] \geq \mathbb{P}\left[\text{Bin}\left(\tau 2^{i-1}, d \frac{\tau (1 + d'/2)^{i-1}}{\tau 2^i}\right) > (1 - \varepsilon) \mu\right] \geq 1 - e^{-\frac{\varepsilon^2 \mu}{2}}.$$

By the union bound,

$$\mathbb{P}[\exists i, 1 \leq i \leq i_{\max} \text{ such that } X_i \text{ fails}] \leq \sum_{i=1}^{i_{\max}} e^{-\frac{\varepsilon^2 d \tau (1+d'/2)^{i-1}}{4}}.$$

We wish to show that the right hand side goes to 0 as  $t$  increases. We have

$$\begin{aligned} \sum_{i=1}^{i_{\max}} e^{-\frac{\varepsilon^2 d \tau (1+d'/2)^{i-1}}{4}} &\leq i_{\max} \left( e^{-\frac{\varepsilon^2 d \tau}{4}} \right) && \text{(by monotonicity)} \\ &= \left( \log_2 \frac{t}{\ln t} \right) \left( t^{-\frac{\varepsilon^2 d}{4}} \right), && \text{(by definitions of } i_{\max}, \tau) \end{aligned}$$

which indeed approaches 0 as  $t$  increases.

**Bounding the Maximum Weight** Note that the weight of  $B_0$  at time  $t$  is exactly  $S_{i_{\max}}$ . Set  $x := 1 + d'/2 > 1$ , which is a constant. With high probability, by [Equation \(6.9\)](#),

$$\frac{S_{i_{\max}}}{\tau} \geq \left(1 + \frac{d'}{2}\right)^{i_{\max}} = x^{\log_2 \frac{t}{\ln t}} = \left(\frac{t}{\ln t}\right)^{\log_2 x}.$$

Since  $x > 1$ ,  $\log_2 x > 0$ . For any  $0 < \beta < \log_2 x$ ,  $\frac{S_{i_{\max}}}{\tau} \in \Omega(t^\beta)$  with high probability. Since  $B_0$  has weight  $S_{i_{\max}}$  and contains at most  $\tau$  voters, with high probability there is some voter in  $B_0$  with that much weight.  $\square$

Before proceeding to the upper bound and showing the separation, we would like to point out that — with a minor change to our model — these lower bounds also hold for  $\gamma = 1$ . Consider a model in which the probability of attaching to a delegator  $n$  remains proportional to  $(1 + \text{indegree}(n))^\gamma$ , but the probability

for voters  $n$  is now proportional to  $(2 + \text{indegree}(n))^\gamma$ .<sup>5</sup> If we represent voters with a self-loop edge, both terms just equal  $\text{degree}(n)^\gamma$ , which arguably makes this implementation of preferential attachment cleaner to analyze (e.g., [BRo4]). Thus, we can interpret preferential attachment for  $\gamma = 1$  as uniformly picking an edge and then flipping a fair coin to decide whether to attach the new node to the edge's start or endpoint. Since every node has exactly one outgoing edge, this is equivalent to uniformly choosing a node and then, with probability  $\frac{1}{2}$ , instead picking its successor. This has the same effect on the distribution of weights as just uniformly choosing a node in uniform delegation, so [Theorem 6.8](#) also holds for  $\gamma = 1$  in our modified setting. Real-world delegation networks, which we suspect to resemble the case of  $\gamma = 1$ , should therefore exhibit similar behavior.

5: Clearly, our results for  $\gamma = 0$  hold for both variants.

[BRo4] Bollobás and Riordan (2004): *Mathematical Results on Scale-Free Random Graphs*.

### 6.3.3 Upper Bound for Double Delegation ( $k = 2, \gamma = 0$ )

Analyzing cases with  $k > 1$  is considerably more challenging. One obstacle is that we do not expect to be able to incorporate optimal resolution of potential delegations into our analysis, because the computational problem is hard even when  $k = 2$  (see [Theorem 6.5](#)). Therefore, we give a pessimistic estimate of optimal resolution via a greedy delegation mechanism, which we can reason about alongside the stochastic process. Clearly, if this stochastic process can guarantee an upper bound on the maximum weight with high probability, this bound must also hold if delegations are optimally resolved to minimize maximum weight.

In more detail, whenever a new delegator is inserted into the graph, the greedy mechanism immediately selects one of the delegation options. As a result, at any point during the construction of the graph, the algorithm can measure the weight of the voters. Suppose that a new delegator suggests two delegation options, to agents  $a$  and  $b$ . By following already resolved delegations, the mechanism obtains voters  $a^*$  and  $b^*$  such that  $a$  transitively delegates to  $a^*$  and  $b$  to  $b^*$ . The greedy mechanism then chooses the delegation whose voter currently has lower weight, resolving ties arbitrarily.

This situation is reminiscent of a phenomenon known as the “power of choice.” In its most isolated form, it has been studied in the *balls-and-bins* model, for example by Azar et al. [ABKU94]. In this model,  $n$  balls are to be placed in  $n$  bins. In the classical setting, each ball is sequentially placed into a bin chosen uniformly at random. With high probability, the fullest bin will contain  $\Theta(\log n / \log \log n)$  balls at the end of the process. In the choice setting, two bins are independently and uniformly selected for every ball, and the ball is placed into the emptier one. Surprisingly, this leads to an exponential improvement, where the fullest bin will contain at most  $\Theta(\log \log n)$  balls with high probability.

[ABKU94] Azar et al. (1994): *Balanced Allocations*.

We show that, at least for  $\gamma = 0$  in our setting, this effect outweighs the “rich-get-richer” dynamic described earlier:

**Theorem 6.9** *In the preferential delegation model with  $k = 2, \gamma = 0$ , and  $d \in (0, 1)$ , the maximum weight of any voter at time  $t$  is  $\log_2 \ln t + \Theta(1)$  with high probability.*

Because the proof of [Theorem 6.9](#) is quite intricate and technical, we only present a sketch of its structure in this thesis. In our proof we build on work by Malyshkin and Paquette [[MP15](#)], who study the maximum *degree* in a graph generated by preferential attachment with the power of choice. In addition, we incorporate ideas by Haslegrave and Jordan [[HJ16](#)].<sup>6</sup> Proofs for the individual lemmas can be found in [Appendix A](#) of the full version.

For our analysis, it would be natural to keep track of the number of voters  $v$  with a specific weight  $w_j(v) = k$  at a specific point  $j$  in time. In order to simplify the analysis, we instead keep track of random variables

$$F_j(k) := \sum_{\substack{v \in V \\ w_j(v) \geq k}} w_j(v),$$

i.e., we sum up the weights of all voters with weight at least  $k$ . Since the total weight increases by one in every step, we have

$$\forall j. F_j(1) = j, \text{ and} \quad (6.10)$$

$$\forall j, k. F_j(k) \leq j. \quad (6.11)$$

If  $F_j(k) < k$  for some  $j$  and  $k$ , the maximum weight of any voter must be below  $k$ .

If we look at a specific  $k > 1$  in isolation, the sequence  $(F_j(k))_j$  evolves as a Markov process initialized at  $F_1(k) = 0$  and then governed by the rule

$$F_{m+1}(k) - F_m(k) = \begin{cases} 1 & \text{with probability } d \cdot \left(\frac{F_m(k)}{m}\right)^2 \\ k & \text{with probability } d \cdot \left(\left(\frac{F_m(k-1)}{m}\right)^2 - \left(\frac{F_m(k)}{m}\right)^2\right) \\ 0 & \text{otherwise} \end{cases} \quad (6.12)$$

In the first case, both potential delegations of a new delegator lead to voters who already had weight at least  $k$ . We must thus give the delegator's vote to one of them, increasing  $F_m(k)$  by one. In the second case, a new delegator offers two delegations leading to voters of weight at least  $k - 1$ , at least one of which has exactly weight  $k - 1$ . Our greedy algorithm will then choose a voter with weight  $k - 1$ . Because this voter is counted in the definition of  $F_j(k)$ ,  $F_m(k)$  increases by  $k$ . Finally, if a new voter appears, or if a new delegator can transitively delegate to a voter with weight less than  $k - 1$ , then  $F_m(k)$  does not change.

In order to bound the maximum weight of a voter, we first need to get a handle on the general distribution of weights. For this, we define a sequence of real numbers  $(\alpha_k)_k$  such that, for every  $k \geq 1$ , the sequence  $F_j(k)/j$  converges in probability to  $\alpha_k$ . Set  $\alpha_1 := 1$ . For every  $k > 1$ , let  $\alpha_k$  be the unique root  $0 < x < \alpha_{k-1}$  of the polynomial

$$a_k(x, p) := d x^2 + k d (p^2 - x^2) - x \quad (6.13)$$

for  $p$  set to  $\alpha_{k-1}$ .<sup>7</sup> Since  $a_k(0, \alpha_{k-1}) > 0$  and  $a_k(\alpha_{k-1}, \alpha_{k-1}) < 0$ , such a solution exists by the intermediate value theorem. Because the polynomial is quadratic, such a solution must be unique in the interval. It follows that the  $\alpha_k$  form a strictly decreasing sequence in the interval  $(0, 1]$ .

[[MP15](#)] Malyshkin and Paquette (2015): *The Power of Choice over Preferential Attachment*.

[[HJ16](#)] Haslegrave and Jordan (2016): *Preferential Attachment with Choice*.

6: More precisely, for the definition of the sequence  $(\alpha_k)_k$  as well as in Lemmas 13 and 14 in [Appendix A.1](#) of the full version.

7: The equation  $0 = a_k(x, p)$  can be obtained from [Equation \(6.12\)](#) by naïvely assuming that  $F_j(k-1)/j$  converges to a value  $p$  and  $F_j(k)/j$  converges to  $x$ , then plugging these values in the expectation of the recurrence.

The sequence  $(\alpha_k)_k$  converges to zero, and eventually does so very fast. However, this is not obvious from the definition and, depending on  $d$ , the sequence can initially decrease slowly. In [Lemma 13](#) in the full version, we demonstrate convergence to zero, and in [Lemma 14](#) we show that the sequence decreases at a rate in  $O(k^{-2})$ . Based on this, in [Lemma 15](#), we choose an integer  $k_0$  such that the sequence decreases very fast from there. In the same lemma, we define a more nicely behaved sequence  $(f(k))_{k \geq k_0}$  that is a strict upper bound on  $(\alpha_k)_{k \geq k_0}$  and that is contained between two doubly-exponentially decaying functions.

**Lemma 6.10** For all  $k \geq 1$ ,  $\varepsilon > 0$  and functions  $\omega(m)$  such that  $\omega(m) \rightarrow \infty$  and  $\omega(m) < m$  (for sufficiently large  $m$ ),

$$\mathbb{P}[\exists j, \omega(m) \leq j \leq m \text{ s.t. } F_j(k)/j > \alpha_k + \varepsilon] \rightarrow 0.$$

*Proof sketch (detailed in [Appendix A.2](#) of the full version).* The proof proceeds by induction on  $k$ . For  $k = 1$ , the claim directly holds. For larger  $k$ , we use a suitably chosen  $\delta$  in place of  $\varepsilon$  and  $\omega_0$  in place of  $\omega$  for the induction hypothesis. With the induction hypothesis, we bound the  $\frac{F_m(k-1)}{m}$  term in the recurrence in [Equation \(6.12\)](#). Furthermore, all increments  $F_j(k) - F_{j-1}(k)$  where  $\frac{F_{j-1}(k)}{j-1} \geq \alpha_k$  holds can be dominated by independent and *identically distributed* random variables  $\eta'_j$ .

Denote by  $\pi$  the first point  $j \geq \omega_0(m)$  such that  $F_j(k)/j \leq \alpha_k + \varepsilon/2$ . The  $\eta'_j$  then dominate all increments  $F_j(k) - F_{j-1}(k)$  for  $\omega_0(m) < j \leq \pi$ . Using Chernoff's bound and suitably chosen  $\delta$  and  $\omega_0$ , we show that, with high probability,  $\pi \leq \omega(m)$ .

Because of this, if  $F_j(k)/j > \alpha_k + \varepsilon$  for some  $j \geq \omega(m)$ , the sequence  $(F_j(k)/j)_j$  must eventually cross from below  $\alpha_k + \frac{\varepsilon}{2}$  to above  $\alpha_k + \varepsilon$  without in between falling below  $\alpha_k$ . On this segment, we can overapproximate the sequence by a random walk with increments distributed as  $\eta'_j$ . Since the sequence might previously decrease below  $\alpha_k$  an arbitrary number of times, we overapproximate the probability of ever crossing  $\alpha_k + \varepsilon$  for  $j \geq \omega(m)$  by a sum over infinitely many random walks. This sum converges to 0 for  $m \rightarrow \infty$ , which shows our claim.  $\square$

The above lemma gives us a good characterization of the behavior of  $(F_j(k))_j$  for any fixed  $k$  (and large enough  $j$ ). To prove an upper bound on the maximum weight, however, we are ultimately interested in statements about  $F_j(k(m))$ , where  $k(m) \in \Theta(\log_2 \ln m)$  and the range of  $j$  varies with  $m$ . In order to obtain such results, we will first show in [Lemma 6.11](#) that whole ranges of  $k$  simultaneously satisfy bounds with high probability.

As in the previous lemma, we can only show our bounds with high probability for  $j$  past a certain period of initial chaos. We will define a function,  $\phi(m, k)$ , that takes a role similar to  $\omega(m)$  in [Lemma 6.10](#). The function  $\phi(m, k)$  gives each  $k$  a certain amount of time to satisfy the bounds, depending on  $m$ : Let  $\rho(m) := (\ln \ln m)^{\frac{1}{3}}$  and define  $\phi(m, k) := \rho(m) C^{2^{k+1}}$ , where  $C$  is an integer that is sufficiently large to satisfy

$$\ln C > \max\left(1, c_1, \ln\left(\frac{2}{1-d}\right) + \frac{c_1}{2}\right). \quad (6.14)$$

In the above,  $c_1$  is a positive constant defining the lower bound on  $f(k)$  in [Lemma 15](#) in the full version.

Additionally, let  $k_*(m)$  be the smallest integer such that

$$C^{2^{k_*(m)+1}} \geq \sqrt{m}. \quad (6.15)$$

Note that  $C^{2^{k_*(m)+1}} < m$  because increasing the double exponent in increments of 1 is equivalent to squaring the term. By applying logarithms to  $C^{2^{k_*(m)+1}} \geq \sqrt{m}$  and  $C^{2^{k_*(m)+1}} < m$ , we obtain  $\log_2 \log_C m - 2 \leq k_*(m) < \log_2 \log_C m - 1$ , from which it follows that  $k_*(m) = \log_2 \ln m + \Theta(1)$ .

**Lemma 6.11** *With high probability, for all  $k_0 \leq k \leq k_*(m)$ , and for all  $\phi(m, k) \leq j \leq m$ ,  $F_j(k)/j \leq f(k)$ .*

*Proof sketch (detailed in [Appendix A.3](#) of the full version).* Let  $\mathcal{E}_k$  be the event

$$\mathcal{E}_k := \left\{ \forall j, \phi(m, k) \leq j \leq m, F_j(k)/j \leq f(k) \right\}.$$

Our goal is to show that  $\mathcal{E}_k$  holds for all  $k$  in our range. In the spirit of an inductive argument, we begin by showing  $\mathcal{E}_{k_0}$  with high probability and then give evidence for how, under the assumption  $\mathcal{E}_k$ ,  $\mathcal{E}_{k+1}$  is likely to happen. Instead of an explicit induction, we piece together these parts in a union bound.

The base case  $\mathcal{E}_{k_0}$  follows from [Lemma 6.10](#) with  $\omega(m) := \phi(m, k_0)$  and  $\varepsilon := f(k_0) - \alpha_{k_0}$ .

For the step, fix some  $k \geq k_0$ , and assume  $\mathcal{E}_k$ . We want to give an upper bound on the probability that  $\mathcal{E}_{k+1}$  happens. We split this into multiple substeps: First, we prove that, given  $\mathcal{E}_k$ , some auxiliary event  $\overline{\mathcal{E}}(k+1)$  happens only with probability converging to 0. Then, we show that  $\overline{\mathcal{E}}(k+1) \subseteq \mathcal{E}_{k+1}$  where  $\overline{\mathcal{E}}$  denotes the complement of an event  $\mathcal{E}$ . This means that, whenever the unlikely event does not take place,  $\mathcal{E}_{k+1}$  holds. This allows the step to be repeated.

If  $\mathcal{E}_k$  does not hold for any  $k_0 \leq k \leq k_*(m)$ , then  $\overline{\mathcal{E}_{k_0}}$  or one of the  $\overline{\mathcal{E}}(k)$  must have happened. The union bound converges to zero for  $m \rightarrow \infty$ , proving our claim.  $\square$

As promised, the last lemma enables us to speak about the behavior of  $F_j(k(m))$ . We will use a sequence of such statements to show that, with high probability,  $F_j(k(m))$  for some  $k(m)$  does not change over a whole range of  $j$ :

**Lemma 6.12** *There exists  $M > 0$  and an integer  $r > 0$  such that, for  $j_0(m) := (\ln \ln m)^M$ ,  $F_m(k_*(m) + r) = F_{j_0(m)}(k_*(m) + r)$  holds with high probability. In addition, there is  $\beta > \frac{1}{2}$  such that, with high probability,*

$$F_{j_0(m)}(k_*(m) + r - 1) \leq j_0(m)^{1-\beta}. \quad (6.16)$$

*Proof sketch (detailed in [Appendix A.4](#) of the full version).* In [Lemma 17](#) in the full version, we finally get a statement about  $F_j(k_*(m))$ : By choosing different

$k$  for different  $j$  in [Lemma 6.11](#), we obtain a constant  $\beta_0 > 0$  such that, with high probability,

$$\forall j, \ln \ln m \leq j \leq m. F_j(k_*(m))/j \leq j^{-\beta_0}.$$

We now increase  $\beta_0$  until it is larger than  $\frac{1}{2}$ . Set  $r'_0 := 0$  and  $M_0 := 1$ . In [Lemma 18](#) in the full version, we obtain a stronger proposition of the form

$$\forall j, (\ln \ln m)^{M_i} \leq j \leq m. F_j(k_*(m) + r'_i)/j \leq j^{-\beta_i}$$

holding with high probability to obtain, for some  $M_{i+1} > 0$  and with high probability,

$$\forall j, (\ln \ln m)^{M_{i+1}} \leq j \leq m. F_j(k_*(m) + r'_i + 1)/j \leq j^{-\frac{3}{2}\beta_i}.$$

If we set  $r'_{i+1} := r'_i + 1$  and  $\beta_{i+1} := \frac{3}{2}\beta_i$ , we can repeatedly apply this argument until some  $\beta_i > \frac{1}{2}$ . Let  $M, r'$  and  $\beta$  denote  $M_i, r'_i$  and  $\beta_i$ , respectively, for this  $i$ . If, furthermore,  $r := r' + 1$ , [Equation \(6.16\)](#) follows as a special case.

We then simply union-bound the probability of  $F_j(k_*(m) + r)$  increasing for any  $j$  between  $j_0(m)$  and  $m$ . Using the above over-approximation in [Equation \(6.12\)](#) gives us an over-harmonic series, whose value goes to zero with  $m \rightarrow \infty$ .  $\square$

We can now prove [Theorem 6.9](#). Let  $Q_i$  denote the maximum weight after  $i$  time steps.

*Proof of Theorem 6.9.* By [Lemma 6.12](#), with high probability,  $F_m(k_*(m) + r) = F_{j_0(m)}(k_*(m) + r)$ . Therefore, we have that with high probability

$$\begin{aligned} F_m(k_*(m) + r) &= F_{j_0(m)}(k_*(m) + r) \\ &\leq F_{j_0(m)}(k_*(m) + r - 1) && \text{(by monotonicity)} \\ &\leq j_0(m)^{1-\beta} && \text{(by Equation (6.16))} \\ &= \left( (\ln \ln m)^M \right)^{1-\beta} \\ &\leq (\ln \ln m)^{M+1}. \end{aligned}$$

For any  $j$  and  $k$ ,  $Q_j \leq \max\{k, F_j(k)\}$ . Since, for large enough  $m$ ,  $k_*(m) + r < (\ln \ln m)^{M+1}$ , the maximum weight  $Q_m$  is at most  $(\ln \ln m)^{M+1}$  with high probability. This result holds for general  $m$ , so we are allowed to plug in  $j_0(m)$  for  $m$ . Then,  $Q_{j_0(m)} \leq (\ln \ln j_0(m))^{M+1}$ . Moreover,  $(\ln \ln j_0(m))^{(M+1)^2} < j_0(m)$  for sufficiently large  $m$  because  $M$  is a constant and polylogarithmic terms grow asymptotically slower than polynomial terms. Rewriting this yields

$$Q_{j_0(m)} \leq (\ln \ln j_0(m))^{M+1} < j_0(m)^{1/(M+1)}. \quad (6.17)$$

Now, note that  $k_*(m) + r \geq (j_0(m))^{1/(M+1)}$  for large enough  $m$ . Therefore, [Equation \(6.17\)](#) implies that, with high probability, a graph generated in  $j_0(m)$  time steps has no voters of weight  $k_*(m) + r$  or higher. In other words, with high probability,  $F_{j_0(m)}(k_*(m) + r) = 0$ , so with high probability  $F_m(k_*(m) + r) = 0$  (again by

**Lemma 6.12).** This means that the maximum weight after  $m$  time steps is also upper-bounded by  $k_*(m) + r = \log_2 \ln m + \Theta(1)$ .  $\square$

## 6.4 Empirical Results

In this section, we present our simulation results, which support the two main messages of this chapter: that allowing multiple delegation options significantly reduces the maximum weight, and that it is computationally feasible to resolve delegations in a way that is close to optimal.

Our simulations were performed on a MacBook Pro (2017) on MacOS 10.12.6 with a 3.1 GHz Intel Core i5 and 16 GB of RAM. All running times were measured with at most one process per processor core. Our simulation software is written in Python 3.6 using Gurobi 8.0.1 to solve MILPs. All of our simulation code is open-source and available at <https://github.com/pgoelz/fluid>.

### 6.4.1 Multiple vs. Single Delegations

For the special case of  $\gamma = 0$ , we have established a doubly exponential, asymptotic separation between single delegation ( $k = 1$ ) and two delegation options per delegator ( $k = 2$ ). While the strength of the separation suggests that some of this improvement will carry over to the real world, we still have to examine via simulation whether improvements are visible for realistic numbers of agents and other values of  $\gamma$ .

To this end, we empirically evaluate two different mechanisms for resolving delegations. First, we optimally resolve delegations by solving the MILP for confluent flow with the Gurobi optimizer. Our second mechanism is the greedy “power of choice” algorithm used in the theoretical analysis and introduced in [Section 6.3.3](#).

In [Figure 6.3](#), we compare the maximum weight produced by a single-delegation process to the optimal maximum weight in a double-delegation process, for different values of  $\gamma$  and  $d$ . Since our theoretical analysis used a greedy over-approximation of the optimum, we also run the greedy mechanism on the double-delegation process. Corresponding figures for  $\gamma = 0.5$  can be found in [Appendix B.1](#) of the full version.

These simulations show that our asymptotic findings translate into considerable differences even for small numbers of agents, across different values of  $d$ . Moreover, these differences remain nearly as pronounced for values of  $\gamma$  up to 1, which corresponds to classical preferential attachment. This suggests that our mechanism can outweigh the social tendency towards concentration of votes; however, evidence from real-world elections is needed to settle this question. Lastly, we would like to point out the similarity between the graphs for the optimal maximum weight and the result of the greedy algorithm, which indicates that a large part of the separation can be attributed to the power of choice.

If we increase  $\gamma$  to large values, the separation between single and double delegation disappears. In [Figure 6.4a](#), for  $\gamma = 2$ , all three curves are hardly distinguishable from the linear function  $d \cdot \text{time}$ , meaning that one voter receives nearly all the weight. The reason is simple: In the simulations used for that figure, 99% of all delegators give two identical delegation options, and 99.8% of these delegators (98.8% of all delegators) give both potential delegations to the heaviest voter in the graph. There are even values of  $\gamma > 1$  and  $d$  such that the curve for single delegation falls below the ones for double delegation. This can be seen in [Figure 6.4b](#), where 87.7% of voters give two identical delegation options. Since adding two delegation options per step makes the indegrees grow faster, the delegations concentrate toward a single voter more quickly, and again lead to a wildly unrealistic concentration of weight. Thus, it seems that large values of  $\gamma$  do not actually describe our scenario of multiple delegations.

As we have seen, switching from single delegation to double delegation greatly improves the maximum weight in plausible scenarios. It is natural to wonder whether increasing  $k$  beyond 2 will yield similar improvements. As [Figure 6.5](#) shows, however, the returns to increasing  $k$  quickly diminish, which is common to many incarnations of the power of choice [[ABKU94](#)].

[[ABKU94](#)] Azar et al. (1994): *Balanced Allocations*.

### 6.4.2 Evaluating Mechanisms

Already the case of  $k = 2$  appears to have great potential; but how easily can we tap it?

We have observed that, on average, the greedy “power of choice” mechanism comes surprisingly close to the optimal solution. However, this greedy mechanism depends on seeing the order in which our random process inserts agents and on the fact that all generated graphs are acyclic, which need not be true in practice. If the graphs were acyclic, we could simply first sort the agents topologically and then present the agents to the greedy mechanism in reverse order. On arbitrary active graphs, we instead proceed through the strongly connected components in reverse topological order, breaking cycles and performing the greedy step over the agents in the component. To avoid giving the greedy algorithm an unfair advantage, we use this generalized greedy mechanism throughout this section. Thus, we compare the generalized greedy mechanism, the optimal solution, the  $(1 + \ln |V|)$ -approximation algorithm<sup>8</sup> and a random mechanism that chooses a uniformly chosen option per delegator.

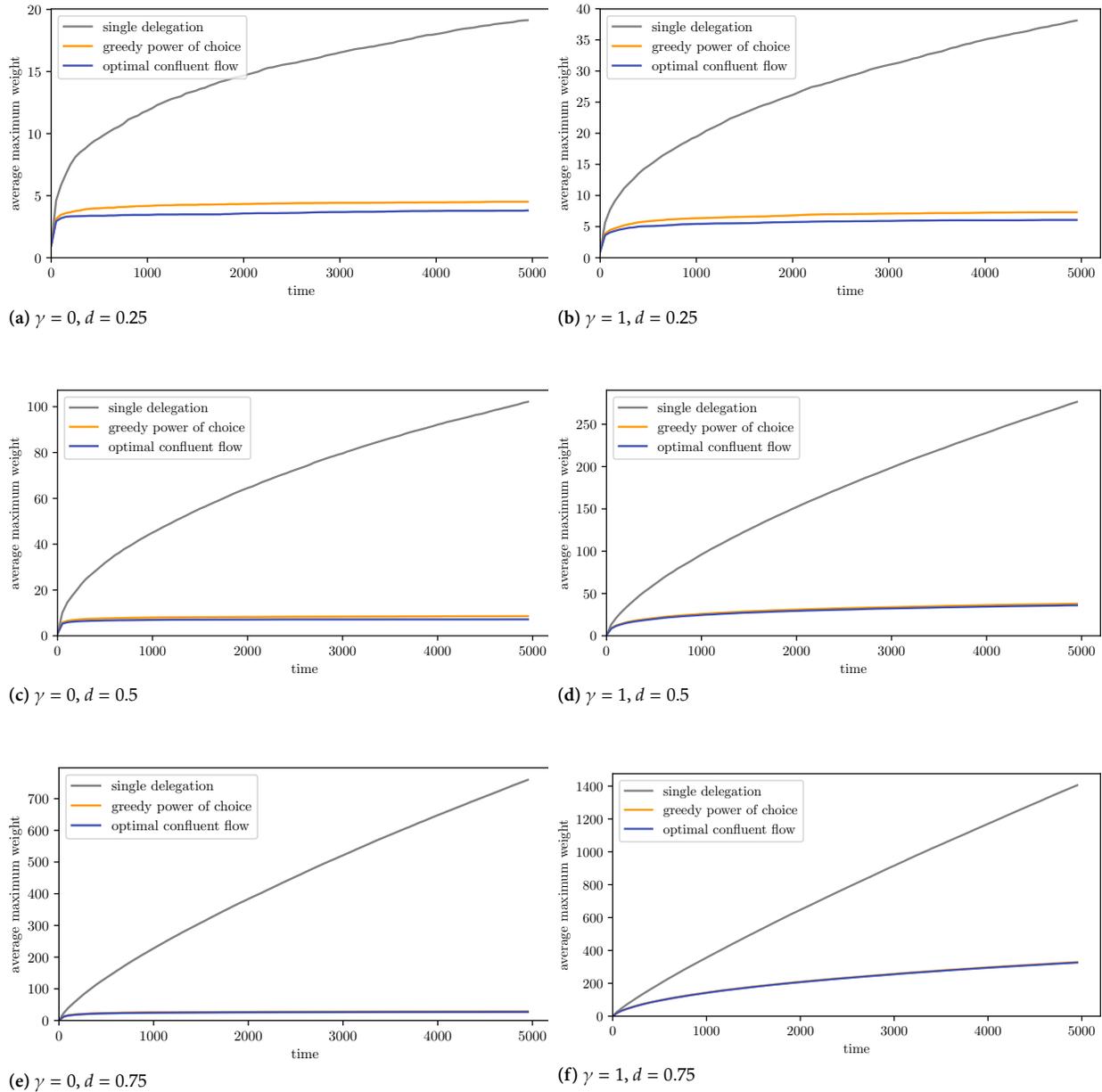
At a high level, we find that both the generalized greedy algorithm and the approximation algorithm perform comparably to the optimal confluent flow solution, as shown in [Figure 6.6](#) for  $d = 0.5$  and  $\gamma = 1$ . As [Figure 6.7](#) suggests, all three mechanisms seem to exploit the advantages of double delegation, at least on our synthetic benchmarks. These trends persist for other values of  $d$  and  $\gamma$ , as presented in [Appendix B.3](#) of the full version.

The similar success of these three mechanisms might indicate that our probabilistic model for  $k = 2$  generates delegation networks that have low maximum weights for arbitrary resolutions. However, this is not the case: The random mechanism does quite poorly on instances with as few as  $t = 100$  agents, as shown in [Figure 6.6a](#). With increasing  $t$ , the gap between random and the other

8: For one of their subprocedures, instead of directly optimizing a convex program, Chen et al. [[CKL+07](#)] reduce this problem to finding a lexicographically optimal maximum flow in  $O(n^5)$ . We choose to directly optimize the convex problem in Gurobi, hoping that this will increase efficiency in practice.

mechanisms only grows further, as indicated by [Figure 6.7](#). In general, the graph for random delegations looks more similar to single delegation than to the other mechanisms on double delegation. Indeed, for  $\gamma = 0$ , random delegation is equivalent to the process with  $k = 1$ , and, for higher values of  $\gamma$ , it performs even slightly worse since the unused delegation options make the graph more centralized (see [Appendix B.2](#) of the full version). Because of the poor performance of random delegation, if simplicity is a primary desideratum, we recommend using the generalized greedy algorithm instead.

As [Figure 6.8](#) demonstrates, all three other mechanisms, including the optimal solution, easily scale to input sizes as large as the largest implementations of liquid democracy to date. Whereas the three mechanisms were close with respect to maximum weight, our implementation of the approximation algorithm is typically slower than the optimal solution (which requires a single call to Gurobi), and the generalized greedy algorithm is blazing fast. These results suggest that it would be possible to resolve delegations almost optimally even at a national scale.



**Figure 6.3:** Maximum weight averaged over 100 simulations of length 5 000 time steps each. Maximum weight has been computed every 50 time steps.

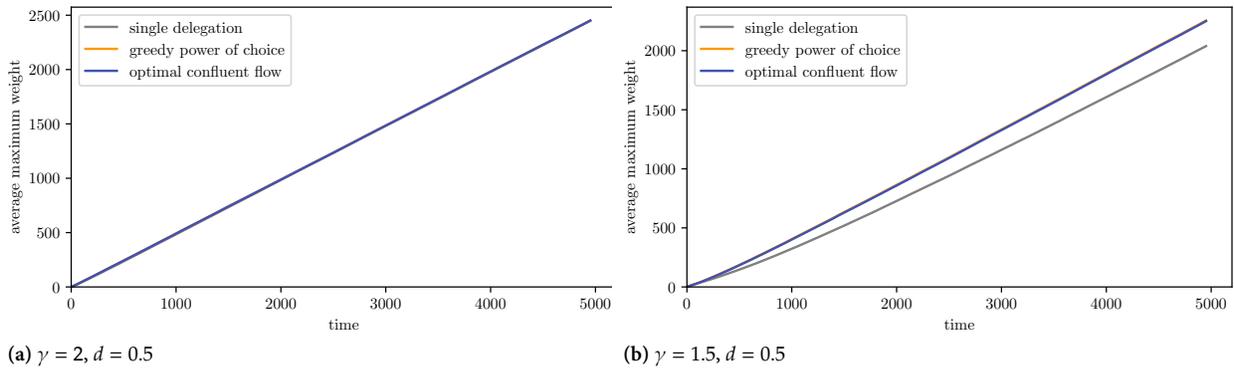


Figure 6.4: Maximum weight averaged over 100 simulations, computed every 50 time steps.

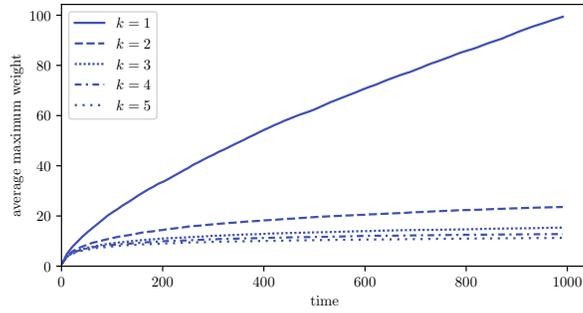


Figure 6.5: Optimal maximum weight for different  $k$  averaged over 100 simulations, computed every 10 steps.  $\gamma = 1, d = 0.5$ .

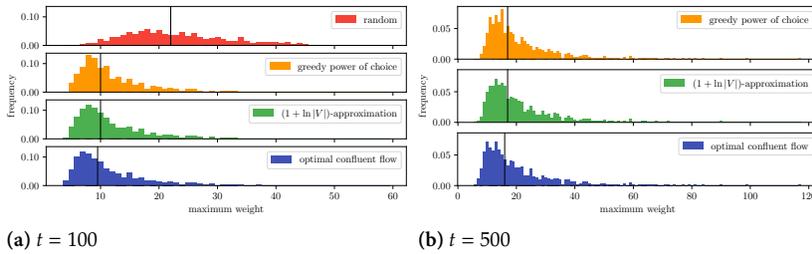


Figure 6.6: Frequency of maximum weights at time  $t$  over 1 000 runs.  $\gamma = 1, d = 0.5, k = 2$ . The black lines mark the medians.

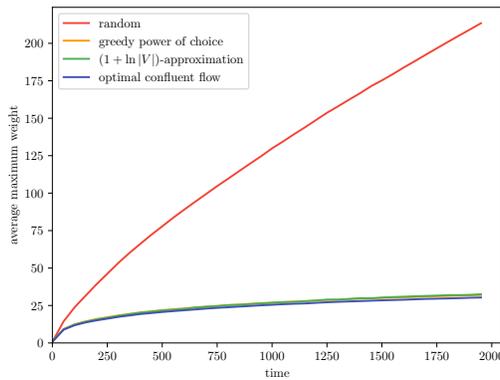


Figure 6.7: Maximum weight per algorithm for  $d = 0.5, \gamma = 1, k = 2$ , averaged over 100 simulations.

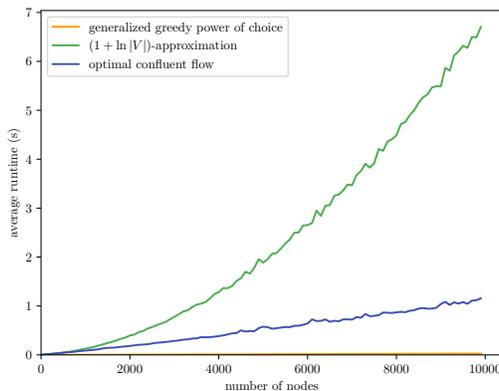
## 6.5 Discussion

The approach we have presented and analyzed revolves around the idea of allowing agents to specify multiple delegation options, and selecting one such option per delegator. As mentioned in [Section 6.2](#), a natural variant of this approach corresponds to splittable — instead of confluent — flow. In this variant, the mechanism would not have to commit to a single outgoing edge per delegator. Instead, a delegator’s weight could be split into arbitrary fractions between their potential delegates. Indeed, such a variant would be computationally less expensive, and the maximum voting weight can be no higher than in our setting. However, we view our concept of delegation as more intuitive and transparent: Whereas, in the splittable setting, a delegator’s vote can disperse among a large number of agents, our mechanism assigns just one representative to each delegator. As suggested in the introduction, this is needed to preserve the high level of accountability guaranteed by classical liquid democracy.

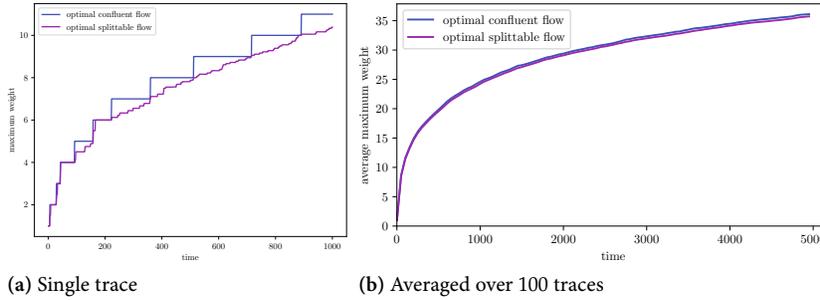
We find that this fundamental shortcoming of splittable delegations is not counterbalanced by a marked decrease in maximum weight. Indeed, representative empirical results given in [Figure 6.9](#) show that the maximum weight trace is almost identical under splittable and confluent delegations. [Figure 6.9a](#) plots a single run of the two solutions over time and suggests that the confluent solution is very close to the ceiling of the fractional LP solution. [Figure 6.9b](#) averages the optimal confluent and splittable solutions over 100 traces to demonstrate that, in our setting, the solution for confluent flow closely approximates the less constrained solution to splittable flow on average. This conclusion is supported by additional results in [Appendix B.3](#) of the full version.

Furthermore, note that in the preferential delegation model with  $k = 1$ , splittable delegations do not make a difference, so the lower bound given in [Theorem 6.8](#) goes through. And, when  $k \geq 2$ , the upper bound of [Theorem 6.9](#) directly applies to the splittable setting. Therefore, our main technical results in [Section 6.3](#) are just as relevant to splittable delegations.

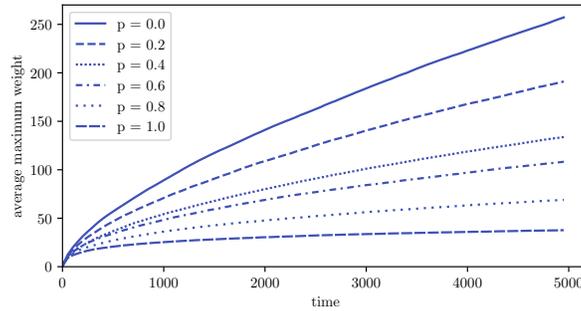
To demonstrate the benefits of multiple delegations as clearly as possible, we assumed that every agent provides two possible delegations. In practice, of course, we expect to see agents who want to delegate but only trust a single person to a sufficient degree. This does not mean that delegators should be required to specify multiple delegations. For instance, if this was the case, delegators might be



**Figure 6.8:** Running time of mechanisms on graphs for  $d = 0.5$ ,  $\gamma = 1$ , averaged over 20 simulations.



**Figure 6.9:** Confluent vs. splittable flow:  $\gamma = 1$ ,  $d = 0.5$ ,  $k = 2$ .



**Figure 6.10:** Optimal maximum weight averaged over 100 simulations. Voters give two delegations with probability  $p$ ; else one.  $\gamma = 1$ ,  $d = 0.5$ .

incentivized to pad their delegations with very popular agents who are unlikely to receive their votes. Instead, we encourage voters to specify multiple delegations on a voluntary basis, and we hope that enough voters participate to make a significant impact. Fortunately, as demonstrated in Figure 6.10, most of the benefits of multiple delegation options persist even if only a fraction of delegators specify two delegations.

The question remains whether sufficiently many agents will indeed be sufficiently close to indifferent between multiple delegates for these benefits to be relevant in practice. Leaving individual incentives aside, how should one trade off the limitation of super-voters against the level of trust in realized delegations? Such questions can be posed in models like the one by Kahng et al. [KMP18], which we leave for future work.

Without doubt, a centralized mechanism for resolving delegations wields considerable power. Even though we only use this power for our specific goal of minimizing the maximum weight, agents unfamiliar with the employed algorithm might suspect it of favoring specific outcomes. To mitigate these concerns, we propose to divide the voting process into two stages. In the first, agents either specify their delegation options or register their intent to vote. Since the votes themselves have not yet been collected, the algorithm can resolve delegations without seeming partial. In the second stage, voters vote using the generated delegation graph, just as in classic liquid democracy, which allows for transparent decisions on an arbitrary number of issues. Additionally, we also allow delegators to change their mind and vote themselves if they are dissatisfied with how delegations were resolved. This gives each agent the final say on their share of votes, and can only further reduce the maximum weight achieved by our mechanism. We believe that this process, along with education about the mechanism's goals and design, can win enough trust for real-world deployment.

Beyond our specific extension, one can consider a variety of different approaches

[KMP18] Kahng et al. (2018): *Liquid Democracy*.

that push the current boundaries of liquid democracy. For example, in a recent position paper, Brill [Bri18] raises the idea of allowing delegators to specify a ranked list of potential representatives. His proposal is made in the context of alleviating delegation cycles, whereas our focus is on avoiding excessive concentration of weight. But, at a high level, both proposals envision centralized mechanisms that have access to richer inputs from agents. Making and evaluating such proposals now is important, because, at this early stage in the evolution of liquid democracy, scientists can still play a key role in shaping this exciting paradigm.

[Bri18] Brill (2018): *Interactive Democracy*.

# Party-List Apportionment with Approval Votes

# 7

## 7.1 Introduction

Whereas the previous chapters all investigated new additions to representative democracies, the next two chapters develop new approaches to an element of representative with a long and colorful history: *legislative apportionment*, the allocation of seats in a legislature to states or political parties. The main principle underlying apportionment is *proportional representation*, which means that the number of representatives in a legislature should be proportional to the number of constituents they represent.<sup>1</sup> Apportionment is used in two major ways: to represent regions in proportion to their populations as in [Chapter 8](#), or to represent political parties in proportion to their electoral support as in this chapter.

In this chapter, we consider apportionment in electoral systems based on the proportional representation of parties, the so-called, “party-list proportional representation” systems [[ACE22](#)]. In these systems, candidates are members of political parties and voters are asked to choose their favorite party; each party is then allocated a number of seats that is (approximately) proportional to the number of votes it received. We will refer to such elections as *party-choice elections* to stress that voters can only vote for a single party. The problem of transforming a voting outcome into a distribution of seats is known as *apportionment*. Analyzing the advantages and disadvantages of different apportionment methods has a long and illustrious political history and has given rise to an elegant mathematical theory [[BY01](#); [Puk14](#)]. (We will touch more on this background in [Chapter 8](#).)

Forcing voters to choose a single party prevents them from communicating any preferences beyond their most preferred alternative. For example, if a voter feels equally well represented by several political parties, there is no way to express this preference within the voting system. In the context of single-winner elections, *approval voting* has been put forward as a solution to this problem as it strikes an attractive compromise between simplicity and expressivity [[BF07](#); [LS10](#)]. Under approval voting, each voter is asked to specify a set of candidates she “approves,” *i. e.* voters can arbitrarily partition the set of candidates into approved candidates and disapproved ones. Proponents of approval voting argue that its introduction could increase voter turnout, “help elect the strongest candidate,” and “add legitimacy to the outcome” of an election [[BF07](#), p. 4–8].

The practical and theoretical appeal of approval voting in single-winner elections has led a number of scholars to suggest to also use approval voting for multiwinner elections, in which a fixed number of candidates need to be elected [[KM12](#)]. Whereas, in the single-winner setting, the straightforward voting rule “choose the candidate approved by the highest number of voters” enjoys a strong axiomatic foundation [[Aló06](#); [BP21](#); [Fis78](#); [Fis79](#)], several ways of aggregating approval ballots have been proposed for the multiwinner setting [[KM12](#); [LS22](#)].

Most studies of approval-based multiwinner elections assume that voters directly express their preference over individual candidates; we refer to this setting

7.1	Introduction	93
7.1.1	Related Work	94
7.1.2	Relation to Other Settings	95
7.1.3	Contributions	96
7.2	Model	96
7.3	Multiwinner Voting Approach	97
7.3.1	Preliminaries	98
7.3.2	Embedding	98
7.3.3	Core Stability Result	99
7.4	Portioning and Apportionment Approach	102
7.4.1	Preliminaries	102
7.4.2	Rules That Fail EJR	104
7.4.3	Rules That Satisfy EJR	105
7.5	Computational Aspects	106
7.6	Discussion	109

1: We saw a closely related notion of proportional representation in our work on citizens’ assemblies in the first part of this thesis. From a mathematical angle, these notions seem very similar, but there are differences in the conception of “representation” from the perspective of political theory [[Pit67](#)].

[[ACE22](#)] ACE Electoral Knowledge Network (2022): *Electoral System (Chamber 1)*.

[[BY01](#)] Balinski and Young (2001): *Fair Representation*.

[[Puk14](#)] Pukelsheim (2014): *Proportional Representation*.

[[BF07](#)] Brams and Fishburn (2007): *Approval Voting*.

[[LS10](#)] Lashier and Sanver (2010): *Handbook on Approval Voting*.

[[KM12](#)] Kilgour and Marshall (2012): *Approval Balloting for Fixed-Size Committees*.

[[Aló06](#)] Alós-Ferrer (2006): *A Simple Characterization of Approval Voting*.

[[BP21](#)] Brandl and Peters (2021): *Approval Voting under Dichotomous Preferences*.

[[Fis78](#)] Fishburn (1978): *Axioms for Approval Voting*.

[[Fis79](#)] Fishburn (1979): *Symmetric and Consistent Aggregation with Dichotomous Voting*.

[[LS22](#)] Lackner and Skowron (2022): *Multi-Winner Voting with Approval Preferences*.

as *candidate-approval* elections. This assumption runs counter to widespread democratic practice, in which candidates belong to political parties and voters indicate preferences over these parties (which induce implicit preferences over candidates). In this chapter, we therefore study *party-approval* elections, in which voters express approval votes over parties and a given number of seats must be distributed among the parties. We refer to the process of allocating these seats as *approval-based apportionment*.

Throughout this chapter, we interpret a ballot that approves a set  $S$  of parties as a preference for legislatures with a larger total number of members from parties in  $S$ . This interpretation generalizes the natural interpretation of party-choice ballots as preferences for legislatures with a larger number of members of the chosen party. Our interpretation implicitly imputes perfect indifference between approved parties. This means that we assume voters to be indifferent to the distribution of seats between approved parties. For example, consider only legislatures with a fixed total number of seats given to approved parties. Then a voter would be indifferent between a legislature where the approved parties all get an equal number of seats, and a legislature where just one of the approved parties obtains all those seats. While this assumption is restrictive, it does allow for a simple voting process, and the additional expressivity of approval ballots compared to party-choice ballots seems attractive.

Indeed, we believe that party-approval elections are a promising framework for legislative elections in the real world, especially since allowing voters to approve multiple parties enables the aggregation mechanism to coordinate like-minded voters. For example, under party-choice elections, two groups of voters might vote for parties that they mutually disapprove. Approval ballots could reveal that both groups approve a third party of more general appeal. Given this information, a voting rule could then allocate more seats to this third party, leading to mutual gain. This cooperation is particularly necessary for small minority opinions that are not centrally coordinated. In such cases, finding a commonly approved party can make the difference between being represented or votes being wasted because the individual parties receive insufficient support.

One aspect that makes it easier to transition from party-choice elections to party-approval elections (rather than to candidate-approval elections) is that party-approval elections can be implemented as *closed-list* systems. That is, parties can retain the power to choose the ordering in which their candidates are allocated seats, as they do in many current democratic systems. By contrast, candidate-approval elections necessarily confer this power to the voters (leading to an *open-list* system), which might give parties an incentive to oppose a change of the voting system. Of course, party-approval elections are compatible with an open-list approach, since we can run a secondary mechanism alongside the party-approval election to determine the order of party candidates.

### 7.1.1 Related Work

To the best of our knowledge, this work is the first to formally develop and systematically study approval-based apportionment. That said, several scholars have previously explored possible generalizations of existing aggregation procedures to allow for approval votes over parties.

For instance, Brams et al. [BKP19] study multiwinner approval rules that are inspired by classical apportionment methods. Besides the setting of candidate approval, they explicitly consider the case where voters cast party-approval votes. They conclude that these rules could “encourage coalitions across party or factional lines, thereby diminishing gridlock and promoting consensus.”

Such desire for compromise is only one motivation for considering party-approval elections, as exemplified by recent work by Speroni di Fenizio and Gewurz [SG19]. To allow for more efficient governing, they aim to concentrate the power of a legislature in the hands of few big parties, while nonetheless preserving the principle of proportional representation. To this end, they let voters cast party-approval votes and transform these votes into a party-choice election by assigning each voter to one of her approved parties. Specifically, they propose to assign voters to parties so that the strongest party has as many votes as possible. We later call this method *majoritarian portioning*.

Several other papers consider extensions of approval-based voting rules to accommodate party-approval elections. In their paper introducing the *satisfaction approval voting* rule, Brams and Kilgour [BK14] discuss a variant of this rule adapted for party-approval votes. Mora and Oliver [MO15] and Camps et al. [CMS19] study two approval-based multiwinner rules due to Phragmén and Eneström, and note that they also work for party-approval elections (which is true for any multiwinner rule using the embedding that we discuss in Section 7.3). Both papers consider a monotonicity axiom for party-approval elections (“if a party receives additional approvals, it should receive additional seats”) but find that their two methods fail it. For the case of two parties, they analyze the behavior of these rules as the house size approaches infinity. They find that both rules fail to converge to the most natural seat distribution. Janson and Öberg [JÖ19] analyze the limit behavior in more detail, and also show that Thiele’s sequential rule (also known as SeqPAV) does converge to the ideal value.

### 7.1.2 Relation to Other Settings

We can position party-approval elections between two well-studied voting settings (see Figure 7.1).

First, our setting can be viewed as a *special case of approval-based multiwinner voting*, in which voters cast *candidate-approval* votes. A party-approval election can be embedded in this setting by replacing each party by multiple candidates belonging to this party, and by interpreting a voter’s approval of a party as approval of all of its candidates. This embedding establishes party-approval elections as a subdomain of candidate-approval elections (see arrow (i) in Figure 7.1). In Section 7.3, we explore the axiomatic and computational ramifications of this domain restriction.

Second, approval-based apportionment *generalizes standard apportionment* (arrow (ii)), which corresponds to party-approval elections in which all approval sets are singletons (i.e., party-choice elections). In Section 7.4, we propose a method to generalize apportionment methods to the party-approval setting using so-called *portioning* methods.

[BKP19] Brams et al. (2019): *Multiwinner Approval Voting*.

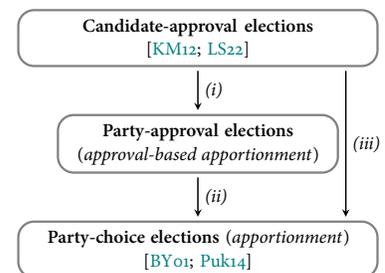
[SG19] Speroni di Fenizio and Gewurz (2019): *The Space of All Proportional Voting Systems and the Most Majoritarian among Them*.

[BK14] Brams and Kilgour (2014): *Satisfaction Approval Voting*.

[MO15] Mora and Oliver (2015): *Eleccions Mitjançant El Vot d’aprovació. El Mètode de Phragmén i Algunes Variants*.

[CMS19] Camps et al. (2019): *The Method of Eneström and Phragmén for Parliamentary Elections by Means of Approval Voting*.

[JÖ19] Janson and Öberg (2019): *A Piecewise Contractive Dynamical System and Phragmén’s Election Method*.



**Figure 7.1:** Relations between the different settings of multiwinner elections. An arrow from  $X$  to  $Y$  signifies that  $X$  is a generalization of  $Y$ . The relationship corresponding to arrow (iii) has been explored by Brill et al. [BLS18]. We establish and explore the relationship (i) in Section 7.3 and the relationship (ii) in Section 7.4.

### 7.1.3 Contributions

In this whapter, we formally introduce the setting of approval-based apportionment and explore different possibilities of constructing axiomatically desirable aggregation methods for this setting. Besides its conceptual appeal, this setting is also interesting from a technical perspective.

Exploiting the relations described in Section 7.1.2, we resolve problems that remain open in the more general setting of candidate-approval elections. First, we show that the core of an approval-based apportionment problem is always nonempty, and that a popular multiwinner rule known as Proportional Approval Voting (PAV) always returns a core-stable committee. We also present a polynomial-time variant of PAV that is also core stable. Second, we prove that house monotonicity is compatible with extended justified representation (a representation axiom proposed by Aziz et al. [ABC+17]) by providing a rule that satisfies both properties.

Some familiar multiwinner rules (in particular, PAV) provide stronger representation guarantees when applied in the party-approval setting. However, for many standard multiwinner voting rules, we give examples that show that their axiomatic guarantees do not improve in the party-approval setting. From a computational complexity perspective, we show that some rules known to be NP-hard in the candidate-approval setting remain NP-hard to evaluate in the party-approval setting. However, it becomes computationally easier to reason about proportionality axioms. Specifically, we show that it is tractable to check whether a given committee satisfies extended justified representation (or the weaker axiom of proportional justified representation). The analogs of these problems for candidate-approval elections are coNP-hard. These tractability results do not extend to checking whether a committee is core-stable: we show that this problem is coNP-complete for both party-approval and candidate-approval elections.

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

## 7.2 Model

A *party-approval election* is a tuple  $(N, P, A, k)$  consisting of a set of voters  $N = \{1, \dots, n\}$ , a finite set of parties  $P$ , a ballot profile  $A = (A_1, \dots, A_n)$  where each ballot  $A_i \subseteq P$  is the set of parties approved by voter  $i$ , and the committee size  $k \in \mathbb{N}$ . We assume that  $A_i \neq \emptyset$  for all  $i \in N$ . When considering computational problems, we assume that  $k$  is encoded in unary (see Remark 7.1). This is a mild restriction since in most applications (such as legislative elections),  $k$  is smaller than the number of voters.

A *committee* in this setting is a multiset  $W : P \rightarrow \mathbb{N}$  over parties, which determines the number of seats  $W(p)$  assigned to each party  $p \in P$ . The size of a committee  $W$  is  $|W| = \sum_{p \in P} W(p)$ , and we denote multiset addition and subtraction by  $+$  and  $-$ , respectively. For a voter  $i$  and a committee  $W$ , we write  $u_i(W) = \sum_{p \in A_i} W(p)$  for the number of seats in  $W$  that are allocated to parties approved by voter  $i$ . A *party-approval rule* is a function that takes a party-approval election  $(N, P, A, k)$  as input and returns a committee  $W$  of valid size  $|W| = k$ .<sup>2</sup>

2: This definition implies that rules are *resolute*, i.e., they only return a single committee. In the case of a tie between multiple committees, a tiebreaking mechanism is necessary. Our results hold independently of the choice of a specific tiebreaking mechanism.

In our axiomatic study of party-approval rules, we focus on two axioms capturing proportional representation: extended justified representation and core stability. Both are derived from their analogs in candidate-approval elections (see Section 7.3.2) where they were proposed by Aziz et al. [ABC+17]. To state these axioms, it is helpful to define the *quota* of a subset  $S$  of voters as  $q(S) = \lfloor k \cdot |S|/n \rfloor$ . Intuitively,  $q(S)$  corresponds to the number of seats that the group  $S$  “deserves” to be represented by (rounded down).

**Definition 7.1** A committee  $W : P \rightarrow \mathbb{N}$  provides extended justified representation (EJR) for a party-approval election  $(N, P, A, k)$  if there is no subset  $S \subseteq N$  of voters such that  $\bigcap_{i \in S} A_i \neq \emptyset$  and  $u_i(W) < q(S)$  for all  $i \in S$ .

In words, EJR requires that for every voter group  $S$  with a commonly approved party, at least one voter of the group must approve at least  $q(S)$  committee members. A party-approval rule is said to *satisfy EJR* if it only produces committees providing EJR.

We can obtain a stronger representation axiom by removing the requirement of a commonly approved party.

**Definition 7.2** A committee  $W : P \rightarrow \mathbb{N}$  is core stable for a party-approval election  $(N, P, A, k)$  if there is no nonempty subset  $S \subseteq N$  and committee  $T : P \rightarrow \mathbb{N}$  of size  $|T| \leq q(S)$  such that  $u_i(T) > u_i(W)$  for all  $i \in S$ . The core of a party-approval election is the set of all core-stable committees.

Core stability requires adequate representation even for voter groups that cannot agree on a common party, by ruling out the possibility that the group can deviate to a smaller committee that represents all voters in the group strictly better. It follows from the definitions that core stability is a stronger requirement than EJR: If a committee violates EJR, there is a group  $S$  that would prefer any committee of size  $q(S)$  that assigns all seats to the commonly approved party.

Besides these representation axioms, a final axiom that we will discuss is *house monotonicity* [BCo8; EFSS17].<sup>3</sup> A party-approval rule  $f$  satisfies this axiom if, for all party-approval elections  $(N, P, A, k)$ , it holds that  $f(N, P, A, k) \subseteq f(N, P, A, k+1)$ . House monotonic rules avoid the so-called *Alabama paradox*, in which a party loses a seat when the committee size increases. They can also be used to construct proportional rankings [IB21; SLB+17].

### 7.3 Constructing Party-Approval Rules via Multiwinner Voting Rules

In this section, we show how party-approval elections can be translated into candidate-approval elections. This embedding allows us to apply established candidate-approval rules to our setting. Exploiting this fact, we will prove the existence of core-stable committees for party-approval elections.

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

[BCo8] Barberà and Coelho (2008): *How to Choose a Non-Controversial List with  $k$  Names*.

[EFSS17] Elkind et al. (2017): *Properties of Multiwinner Voting Rules*.

3: In the multi-winner elections literature, this axiom is called *committee monotonicity*. We adopt the term from the apportionment literature, in part to stress the connection to our work on house monotonicity in Chapter 8.

[IB21] Israel and Brill (2021): *Dynamic Proportional Rankings*.

[SLB+17] Skowron et al. (2017): *Proportional Rankings*.

### 7.3.1 Preliminaries

A *candidate-approval election* is a tuple  $(N, C, A, k)$ . Just as for party-approval elections,  $N = \{1, \dots, n\}$  is a set of voters,  $C$  is a finite set,  $A$  is an  $n$ -tuple of nonempty subsets of  $C$ , and  $k \in \mathbb{N}$  is the committee size. The conceptual difference is that  $C$  is a set of individual candidates rather than parties. This difference manifests itself in the definition of a committee because a single candidate cannot receive multiple seats. That is, a *candidate committee*  $W$  is now simply a subset of  $C$  with cardinality  $k$ . (Therefore, it is usually assumed that  $|C| \geq k$ .) A *candidate-approval rule* is a function that maps each candidate-approval election to a candidate committee.

A diverse set of such voting rules has been proposed since the late 19th century [Jan18a; KM12; LS22], out of which we will only introduce the one which we use for our main positive result. Let  $H_j$  denote the  $j$ th harmonic number, *i. e.*  $H_j = \sum_{t=1}^j 1/t$ . Given  $(N, C, A, k)$ , the candidate-approval rule *proportional approval voting* (PAV), introduced by Thiele [Thi95], chooses a candidate committee  $W$  maximizing the PAV score  $PAV(W) = \sum_{i \in N} H_{|W \cap A_i|}$ .

We now describe EJR and core stability in the candidate-approval setting, from which we derived our versions. Recall that  $q(S) = \lfloor k|S|/n \rfloor$ . A candidate committee  $W$  provides *EJR* if there is no subset  $S \subseteq N$  and no integer  $\ell > 0$  such that  $q(S) \geq \ell$ ,  $|\bigcap_{i \in S} A_i| \geq \ell$ , and  $|A_i \cap W| < \ell$  for all  $i \in S$ . (The requirement  $|\bigcap_{i \in S} A_i| \geq \ell$  is often called *cohesiveness*.) A candidate-approval rule satisfies EJR if it always produces EJR committees.

The definition of core stability is even closer to the version in party-approval elections: A candidate committee  $W$  is *core stable* if there is no nonempty group  $S \subseteq N$  and no set  $T \subseteq C$  of size  $|T| \leq q(S)$  such that  $|A_i \cap T| > |A_i \cap W|$  for all  $i \in S$ . The *core* consists of all core-stable candidate committees.

### 7.3.2 Embedding Party-Approval Elections

We have informally argued in Section 7.1.2 that party-approval elections constitute a subdomain of candidate-approval elections. We formalize this notion by providing an embedding of party-approval elections into the candidate-approval domain. Our approach is similar to that of Brill et al. [BLS18], who have formalized how apportionment problems can be phrased as candidate-approval elections.

For a given party-approval election  $(N, P, A, k)$ , we define a corresponding candidate-approval election  $(N, C, A', k)$  with the same set of voters  $N$  and the same committee size  $k$ . The set of candidates contains  $k$  many “clone” candidates  $p^{(1)}, \dots, p^{(k)}$  for each party  $p \in P$ , so  $C = \bigcup_{p \in P} \{p^{(1)}, \dots, p^{(k)}\}$ . Voter  $i$  approves a candidate  $p^{(j)}$  in the candidate-approval election if and only if she approves the corresponding party  $p$  in the party-approval election. Thus,  $A'_i = \bigcup_{p \in A_i} \{p^{(1)}, \dots, p^{(k)}\}$ .

A candidate committee  $W' \subseteq C$  corresponds to a committee  $W : P \rightarrow \mathbb{N}$  for the party-approval election with  $W(p) = |W' \cap \{p^{(1)}, \dots, p^{(k)}\}|$ . One can also convert in the other direction: a committee  $W : P \rightarrow \mathbb{N}$  for the party-approval election corresponds to the candidate committee  $W' = \bigcup_{p \in P} \{p^{(1)}, \dots, p^{(W(p))}\}$ .

[Jan18a] Janson (2018): *Phragmén's and Thiele's Election Methods*.

[KM12] Kilgour and Marshall (2012): *Approval Balloting for Fixed-Size Committees*.

[LS22] Lackner and Skowron (2022): *Multi-Winner Voting with Approval Preferences*.

[Thi95] Thiele (1895): *Om flerfoldvalg*.

[BLS18] Brill et al. (2018): *Multiwinner Approval Rules as Apportionment Methods*.

This embedding establishes party-approval elections as a subdomain of candidate-approval elections. As a consequence, we can apply rules from the more general candidate-approval setting to the party-approval setting, by

1. translating the party-approval election into a candidate-approval election,
2. applying the candidate-approval rule, and
3. converting its chosen candidate committee into a committee over parties.

Any axiom for candidate-approval rules implies an axiom for the derived party-approval rule. One can check that if a candidate-approval rule satisfies EJR or core stability (as defined in Section 7.3.1) then the derived party-approval rule satisfies the respective axiom (as defined in Section 7.2). In particular, note that by restricting our view to party approval, the cohesiveness requirement of EJR is reduced to requiring a single commonly approved party.

**Remark 7.1** By our assumption that  $k$  is encoded in unary for the purpose of complexity analysis (see Section 7.2), the translation of a party-approval election yields a polynomial-sized candidate-approval election. Thus, a polynomial-time candidate-approval rule applied to the party-approval election runs in polynomial time as well. If  $k$  was instead encoded in binary, elections with large  $k$  and few parties could be described so concisely that even straightforward candidate-approval algorithms would formally have exponential running time.<sup>a</sup> (The same issue does not appear in candidate-approval elections, where we need to list at least  $k$  candidates, which makes the description verbose.)

<sup>a</sup> However, some rules may admit implementations that remain efficient for binary  $k$ . For example, Rule X [PS20], also known as the Method of Equal Shares, will repeatedly assign seats to the same party until one of its supporters runs out of virtual money. Since this happens at most  $n$  times, this observation can be used to design an efficient algorithm (with runtime depending on  $\log k$  instead of  $k$ ). Still, a linear time dependence on  $k$  is acceptable in most applications.

In practice, rather than explicitly going via the embedding, it can be useful to write down the induced party-approval rule directly in terms of a party-approval election. For example, the party-approval version of PAV chooses a committee  $W : P \rightarrow \mathbb{N}$  that maximizes  $PAV(W) = \sum_{i \in N} H_{u_i(W)}$ .

### 7.3.3 PAV Guarantees Core Stability

A powerful stability concept in economics, core stability is a natural extension of EJR. It is particularly attractive because blocking coalitions do not need to unanimously approve any party; they only need to be able to coordinate for mutual gain.

Unfortunately, it is still unknown whether core-stable candidate committees exist for all candidate-approval elections.<sup>4</sup> All standard candidate-approval rules either already fail weaker representation axioms such as EJR, or are known to fail core stability. In particular, PAV satisfies EJR, but may produce non-core-stable committees for candidate-approval elections [ABC+17]. Peters and Skowron [PS20] show that a large class of candidate-approval rules (so-called welfarist rules) must all fail core stability.

4: However, it is known that *approximately* core-stable committees exist, for several different ways of approximating the core notion [CJMW19; FMS18; JMW20; PS20].

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

[PS20] Peters and Skowron (2020): *Proportionality and the Limits of Welfarism*.

For our main result, we show that core stability can always be achieved in the party-approval setting. Specifically, the committee selected by PAV is core stable for party-approval elections. Our proof uses a similar technique to the proof that PAV satisfies EJR for candidate-approval elections [ABC+17, Theorem 10]; we discuss the essential difference in Remark 7.3.

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

**Theorem 7.1** *For every party-approval election, PAV chooses a core-stable committee. Hence, the core of a party-approval election is nonempty.*

*Proof.* Consider a party-approval election  $(N, P, A, k)$  and let  $W_1 : P \rightarrow \mathbb{N}$  be the committee selected by PAV. Assume for a contradiction that  $W_1$  is not core stable. Then there is a nonempty coalition  $S \subseteq N$  and a committee  $T : P \rightarrow \mathbb{N}$  such that  $|T| \leq q(S) \leq k|S|/n$  and  $u_i(T) \geq u_i(W_1) + 1$  for every voter  $i \in S$ .

For each party  $p$ , we let  $\Delta^+(p, W_1)$  denote the marginal increase of the PAV score when we allocate an extra seat to  $p$ . Thus,

$$\Delta^+(p, W_1) = \text{PAV}(W_1 + \{p\}) - \text{PAV}(W_1) = \sum_{i \in N_p} \frac{1}{u_i(W_1) + 1},$$

where  $N_p = \{i \in N \mid p \in A_i\}$ . Let us calculate the average marginal increase when adding an elements of  $T$ :

$$\begin{aligned} \frac{1}{|T|} \sum_{p \in P} T(p) \Delta^+(p, W_1) &= \frac{1}{|T|} \sum_{i \in N} \sum_{p \in A_i} \frac{T(p)}{u_i(W_1) + 1} \geq \frac{1}{|T|} \sum_{i \in S} \sum_{p \in A_i} \frac{T(p)}{u_i(W_1) + 1} \\ &\geq \frac{1}{|T|} \sum_{i \in S} \sum_{p \in A_i} \frac{T(p)}{u_i(T)} = \frac{1}{|T|} \sum_{i \in S} \frac{u_i(T)}{u_i(T)} = \frac{|S|}{|T|} \geq \frac{n}{k}. \end{aligned}$$

Thus, there is a party  $p_1$  with  $\Delta^+(p_1, W_1) \geq n/k$ . Let  $W_2 = W_1 + \{p_1\}$ .

Next, for each party  $p$  with  $W_2(p) > 0$ , let  $\Delta^-(p, W_2)$  be the marginal decrease of the PAV score if we take away a seat from  $p$  in  $W_2$ . Thus,

$$\Delta^-(p, W_2) = \text{PAV}(W_2) - \text{PAV}(W_2 - \{p\}) = \sum_{i \in N_p} \frac{1}{u_i(W_2)}.$$

The average marginal decrease of taking away a seat from  $W_2$  is

$$\begin{aligned} \frac{1}{k+1} \sum_{p \in P} W_2(p) \Delta^-(p, W_2) &= \frac{1}{k+1} \sum_{p \in P} \sum_{i \in N_p} \frac{W_2(p)}{u_i(W_2)} \\ &= \frac{1}{k+1} \sum_{i \in N} \sum_{p \in A_i} \frac{W_2(p)}{u_i(W_2)} \\ &= \frac{1}{k+1} |\{i \in N : u_i(W_2) > 0\}| \leq \frac{n}{k+1}. \end{aligned}$$

Thus, there is some party  $p_2$  with  $W_2(p_2) > 0$  such that  $\Delta^-(p_2, W_2) \leq \frac{n}{k+1}$ . Write  $W_3 = W_2 - \{p_2\} = W_1 + \{p_1\} - \{p_2\}$ . Then

$$\begin{aligned} \text{PAV}(W_3) &= \text{PAV}(W_2) - \Delta^-(p_2, W_2) \\ &= \text{PAV}(W_1) + \Delta^+(p_1, W_1) - \Delta^-(p_2, W_2) \\ &\geq \text{PAV}(W_1) + \frac{n}{k} - \frac{n}{k+1} \end{aligned}$$

$$> \text{PAV}(W_1),$$

contradicting the optimality of  $W_1$ .  $\square$

**Remark 7.2** Our proof of [Theorem 7.1](#) can be easily adapted to show that PAV satisfies the stronger version of core defined with respect to the *Droop quota* [[Dro81](#); [Jan18b](#)], by assuming  $|T| < (k + 1)|S|/n$  rather than  $|T| \leq k|S|/n$ .

**Remark 7.3** For candidate-approval elections, the proof of [Theorem 7.1](#) shows that PAV satisfies core stability restricted to “disjoint objections”: if  $W$  is the committee selected by PAV, then there can be no set  $T$  with  $T \cap W = \emptyset$  such that there is a coalition  $S$  with  $T \leq q(S)$  and  $u_i(T) > u_i(W)$  for all  $i \in S$ . Note that with our embedding of party-approval elections into candidate-approval elections, the disjointness assumption is without loss of generality if there are enough virtual candidates representing each party, and hence PAV satisfies core stability for party-approval elections. The disjoint objections property also implies the result of Peters and Skowron [[PS20](#), Thm. 6] that PAV satisfies the “2-core” property in the candidate-approval context: If there was an objection  $T$  that more than doubled the utility of each coalition member, then  $T \setminus W$  would be a disjoint core deviation, which is a contradiction.

**Remark 7.4** Because  $H_j = \Theta(\log j)$ , the PAV objective is closely related to the classical *maximum Nash welfare* (MNW) solution [[KN79](#); [Nas50](#)]. One can see PAV as a discretization of the MNW solution for selecting a probability distribution  $\sigma : P \rightarrow [0, 1]$  over parties, where we can interpret  $\sigma(p)$  as the fraction of seats that should be allocated to party  $p$ . That rule satisfies a continuous analog of the core condition [[ABM19](#); [FGM16](#)]. However, other natural discretizations of the Nash rule do not satisfy the core condition. In the next section, we will see that discretizing the Nash rule using common apportionment methods leads to violations of core stability. Furthermore, selecting a committee that maximizes Nash welfare (rather than the PAV objective function) may fail core stability, even in party-choice elections [[BLS18](#), Theorem 2].

Given that PAV satisfies core stability in party-approval elections but not in candidate-approval elections, do other candidate-approval rules satisfy stronger representation axioms when restricted to the party-approval subdomain? We have studied this question for various rules besides PAV, and the answer was always negative; see [Appendix B](#) of the full version for details.<sup>5</sup>

A major drawback of PAV is that it fails house monotonicity, and PAV continues to fail this axiom in the party-approval setting.<sup>6</sup> Therefore, parties may lose seats when the committee size is increased. In the next section, we construct party-approval rules that avoid this undesirable behavior.

[[Dro81](#)] Droop (1881): *On Methods of Electing Representatives*.

[[Jan18b](#)] Janson (2018): *Thresholds Quantifying Proportionality Criteria for Election Methods*.

[[PS20](#)] Peters and Skowron (2020): *Proportionality and the Limits of Welfareism*.

[[KN79](#)] Kaneko and Nakamura (1979): *The Nash Social Welfare Function*.

[[Nas50](#)] Nash (1950): *The Bargaining Problem*.

[[ABM19](#)] Aziz et al. (2019): *Fair Mixing*.

[[FGM16](#)] Fain et al. (2016): *The Core of the Participatory Budgeting Problem*.

[[BLS18](#)] Brill et al. (2018): *Multiwinner Approval Rules as Apportionment Methods*.

5: We present relevant counterexamples for the candidate-approval rules seq-Phragmén, leximax-Phragmén, Eneström-Phragmén, Rule X, and the Maximin Support Method. In addition, we verified for the candidate-approval rules SeqPAV, RevSeqPAV, var-Phragmén, Approval Voting (AV), SatisfactionAV, MinimaxAV, MonroeAV, GreedyMonroeAV, GreedyAV, HareAV, and Chamberlin–CourantAV that existing counterexamples can easily be adjusted to the party-approval setting.

6: Existing counterexamples for the candidate-approval setting [[LS22](#)] can be adapted in a straight-forward way.

## 7.4 Constructing Party-Approval Rules via Portioning and Apportionment

Party-approval elections are a generalization of party-choice elections, which can be thought of as party-approval elections in which all approval sets are singletons. Since there is a rich body of research on apportionment methods [BY01; Puk14] which act on party-choice elections, it is natural to examine whether we can employ these methods for our setting as well. To use them, we will need to translate party-approval elections into the party-choice domain on which apportionment methods operate. This translation thus needs to transform a collection of approval votes over parties into vote shares for each party. Motivated by time sharing, Bogomolnaia et al. [BMS05] have developed a theory of such transformation rules, further studied by Duddy [Dud15], Aziz et al. [ABM19], and Brandl et al. [BBPS21]. We will refer to this framework as *portioning*.

The approach explored in this section, then, divides the construction of a party-approval rule into two independent steps: (1) portioning, which maps a party-approval election to a vector of parties' shares; followed by (2) apportionment, which transforms the shares into a seat distribution.

Both the portioning and the apportionment literature have discussed representation axioms similar in spirit to EJR and core stability. For both settings, several rules have been found to satisfy these properties. One might hope that by composing two rules that are each representative, we obtain a party-approval rule that is also representative (and satisfies, say, EJR). If we succeed in finding such a combination, it is likely that the resulting voting rule will automatically satisfy house monotonicity since most apportionment methods satisfy this property. In the general candidate-approval setting (considered in Section 7.3), the existence of a rule satisfying both EJR and house monotonicity is an open problem [LS22].

### 7.4.1 Preliminaries

We start by introducing relevant notions from the portioning literature [ABM19; BMS05] and apportionment [BY01; Puk14], with notation suitably adjusted to our setting.

#### Portioning.

A *portioning problem* is a triple  $(N, P, A)$ , just as in party-approval voting but without a committee size. A *portioning* is a function  $r : P \rightarrow [0, 1]$  with  $\sum_{p \in P} r(p) = 1$ . We interpret  $r(p)$  as the vote share of party  $p$ . A *portioning method* maps each portioning problem  $(N, P, A)$  to a portioning.

Our minimum requirement on portioning methods will be that they uphold proportionality if all approval sets are singletons, *i.e.* if we are already in the party-choice domain. Formally, we say that a portioning method is *faithful* if for all  $(N, P, A)$  with  $|A_i| = 1$  for all  $i \in N$ , the resulting portioning  $r$  satisfies  $r(p) = |\{i \in N \mid A_i = \{p\}\}|/n$  for all  $p \in P$ . Among the portioning methods considered by Aziz et al. [ABM19], only the following three are faithful:

[BY01] Balinski and Young (2001): *Fair Representation*.

[Puk14] Pukelsheim (2014): *Proportional Representation*.

[BMS05] Bogomolnaia et al. (2005): *Collective Choice under Dichotomous Preferences*.

[Dud15] Duddy (2015): *Fair Sharing under Dichotomous Preferences*.

[ABM19] Aziz et al. (2019): *Fair Mixing*.

[BBPS21] Brandl et al. (2021): *Distribution Rules Under Dichotomous Preferences*.

[LS22] Lackner and Skowron (2022): *Multi-Winner Voting with Approval Preferences*.

**Conditional utilitarian portioning** selects, for each voter  $i$ ,  $p_i$  as a party in  $A_i$  approved by the highest number of voters. Then,  $r(p) = \{|i \in N \mid p_i = p\}|/n$  for all  $p \in P$ .

**Random priority** computes  $n!$  portionings, one for each permutation  $\sigma$  of  $N$ , and returns their average. The portioning for  $\sigma = (i_1, \dots, i_n)$  maximizes  $\sum_{p \in A_{i_1}} r(p)$ , breaking ties by maximizing  $\sum_{p \in A_{i_2}} r(p)$ , and so forth.

**Nash portioning** selects the portioning  $r$  that maximizes the Nash welfare, i.e.,  $\prod_{i \in N} \left( \sum_{p \in A_i} r(p) \right)$ .

When computing the outcomes of these rules, ties may occur. For our results it will not matter how ties are broken: we only use these rules in counterexamples in which no ties occur.

On first sight, Nash portioning seems particularly promising because it satisfies portioning versions of core stability and EJR [ABM19; GN14]. Concretely, it satisfies a property called *average fair share* introduced by Aziz et al. [ABM19], which requires that there is no subset  $S \subseteq N$  of voters such that  $\bigcap_{i \in S} A_i \neq \emptyset$  and  $\frac{1}{|S|} \sum_{i \in S} \sum_{p \in A_i} r(p) < |S|/|N|$ . However, despite these promising properties, we will see that Nash portioning does not work for our purposes. Instead, we will need to make use of a more recent portioning approach, which was proposed by Speroni di Fenizio and Gewurz [SG19] in the context of party-approval voting.

**Majoritarian portioning** proceeds in rounds  $j = 1, 2, \dots$ . Initially, all parties and voters are *active*. In iteration  $j$ , select the active party  $p_j$  that is approved by the highest number of active voters. Let  $N_j$  be the set of active voters who approve  $p_j$ . Then, set  $r(p_j)$  to  $|N_j|/n$ , and mark  $p_j$  and all voters in  $N_j$  as inactive. If active voters remain, start the next iteration; otherwise, return  $r$ .

Under majoritarian portioning, we ignore the approval preferences of voters after they have been “assigned” to a party. Note that conditional utilitarian portioning is a similar sequential method which does, however, not ignore the preferences of inactive voters.

## Apportionment.

An *apportionment problem* is a tuple  $(P, r, k)$ , which consists of a finite set of parties  $P$ , a portioning  $r : P \rightarrow [0, 1]$  specifying the vote shares of parties, and a committee size  $k \in \mathbb{N}$ . Committees are defined as for party-approval elections, and an *apportionment method* maps apportionment problems to committees  $W$  of size  $k$ .

An apportionment method satisfies *lower quota* if each party  $p$  is always allocated at least  $\lfloor k \cdot r(p) \rfloor$  seats in the committee. Furthermore, an apportionment method  $f$  is *house monotonic* if  $f(P, r, k) \subseteq f(P, r, k+1)$  for every apportionment problem  $(P, r, k)$ .

Among the standard apportionment methods, only two satisfy both lower quota and house monotonicity: the *D’Hondt method* (also known as the *Jefferson method*) and the *quota method*.<sup>7</sup> The D’Hondt method assigns the  $k$  seats iteratively, each time giving the next seat to the party  $p$  with the largest quotient  $r(p)/(s(p) + 1)$ , where  $s(p)$  denotes the number of seats already assigned to  $p$ . The

[ABM19] Aziz et al. (2019): *Fair Mixing*.

[GN14] Guerdjikova and Nehring (2014): *Weighing Experts, Weighing Sources: The Diversity Value*.

[ABM19] Aziz et al. (2019): *Fair Mixing*.

[SG19] Speroni di Fenizio and Gewurz (2019): *The Space of All Proportional Voting Systems and the Most Majoritarian among Them*.

7: All other *divisor methods* fail lower quota, and the *Hamilton method* is not house monotonic [BY01].

*quota method* [BY75] is identical to the D'Hondt method, except that, in the  $j$ th iteration, only parties  $p$  satisfying  $s(p)/j < r(p)$  are eligible for the allocation of the next seat. Ties should be broken in a consistent fashion so as to ensure house monotonicity, for example using a fixed tie-breaking order over parties.

[BY75] Balinski and Young (1975): *The Quota Method of Apportionment*.

### Composition.

If we take any portioning method and any apportionment method, we can compose them to obtain a party-approval rule. Formally, the composition of portioning method  $R$  and apportionment method  $M$  maps each party-approval election  $(N, P, A, k)$  to a committee  $M(P, R(N, P, A), k)$ . Note that if the apportionment method is house monotonic then so is the composed rule, since the portioning is independent of  $k$ .

### 7.4.2 Composed Rules That Fail EJR

Perhaps surprisingly, many pairs of portioning and apportionment methods fail EJR. This is certainly true if the individual parts are not representative themselves. For example, if an apportionment method  $M$  “properly” fails lower quota (in the sense that there is a rational-valued input  $r$  on which lower quota is violated), then one can construct an example profile on which any composed rule using  $M$  fails EJR: Construct a party-approval election with singleton approval sets in which the voter counts are proportional to the shares in the counterexample  $r$ . Then any faithful portioning method, applied to this election, must return  $r$ . Since  $M$  fails lower quota on  $r$ , the resulting committee will violate EJR. By a similar argument, suppose that an apportionment method violates house monotonicity, and that there is a rational-valued counterexample. Then the apportionment method, when composed with a faithful portioning method, will give rise to a party-approval rule that fails house monotonicity.

As mentioned above, D'Hondt and the quota method are the only standard apportionment methods to satisfy both lower quota and house monotonicity. However, the composition of either option with the conditional utilitarian, random priority, or Nash portioning methods fails EJR, as the following examples show.

**Example 7.1** Let  $n = k = 6$ ,  $P = \{p_0, p_1, p_2, p_3\}$ , and consider the ballot profile  $A = (\{p_0\}, \{p_0\}, \{p_0, p_1, p_2\}, \{p_0, p_1, p_2\}, \{p_1, p_3\}, \{p_2, p_3\})$ .

Then, the conditional utilitarian solution sets  $r(p_0) = 4/6$ ,  $r(p_1) = r(p_2) = 1/6$ , and  $r(p_3) = 0$ . Any apportionment method satisfying lower quota allocates four seats to  $p_0$ , one each to  $p_1$  and  $p_2$ , and none to  $p_3$ . The resulting committee does not provide EJR since the last two voters, who jointly approve  $p_3$ , have a quota of  $q(\{5, 6\}) = 2$  that is not met.

**Example 7.2** Let  $n = k = 6$ ,  $P = \{p_0, p_1, p_2, p_3\}$ , and consider the ballot profile  $A = (\{p_0\}, \{p_0\}, \{p_0, p_1, p_2\}, \{p_0, p_1, p_3\}, \{p_1\}, \{p_2, p_3\})$ .

Random priority chooses the portioning  $r(p_0) = 23/45$ ,  $r(p_1) = 23/90$ , and

$r(p_2) = r(p_3) = 7/60$ . Both D'Hondt and the quota method allocate four seats to  $p_0$ , two seats to  $p_1$ , and none to the other two parties. This violates the claim to representation of the sixth voter (with  $q(\{6\}) = 1$ ).

Nash portioning produces a fairly similar portioning, with  $r(p_0) \approx 0.5302$ ,  $r(p_1) \approx 0.2651$ , and  $r(p_2) = r(p_3) \approx 0.1023$ . D'Hondt and the quota method produce the same committee as above, leading to the same EJR violation.

It might be surprising that Nash portioning combined with a lower-quota apportionment method violates EJR. After all, Nash portioning satisfies core stability in the portioning setting, which is a strong notion of proportionality, and the lower-quota property limits the rounding losses when moving from the portioning to a committee. As expected, in the election of Example 7.2, the portioning produced by Nash gives sufficient representation to the sixth voter since  $r(p_2) + r(p_3) \approx 0.2047 > 1/6$ . However, since both  $r(p_2)$  and  $r(p_3)$  are below  $1/6$  on their own, lower quota does not apply to either of the two parties, and the sixth voter loses all representation in the apportionment step.<sup>8</sup>

8: There are similar examples where Nash portioning with D'Hondt apportionment violates EJR even though every party receives at least one seat, and examples where EJR is violated by a margin of more than one seat.

### 7.4.3 Composed Rules That Satisfy EJR

As we have seen, several initially promising portioning methods fail to compose to a rule that satisfies EJR. One reason is that these portioning methods are happy to assign small shares to several parties. The apportionment method may round several of those small shares down to zero seats. This can lead to a failure of EJR when not enough parties obtain a seat. It is difficult for an apportionment method to avoid this behavior since the portioning step obscures the relationships between different parties that are apparent from the approval ballots of the voters.

Since majoritarian portioning maximizes the seat allocations to the largest parties, it tends to avoid the problem we have just identified. While it fails the strong representation axioms that Nash portioning satisfies, this turns out not to be crucial: Composing majoritarian portioning with any apportionment method satisfying lower quota yields an EJR rule. If we use an apportionment method that is also house monotonic, such as D'Hondt or the quota method, we obtain a party-approval rule that satisfies both EJR and house monotonicity.

**Theorem 7.2** *Let  $M$  be a house monotonic apportionment method satisfying lower quota. Then, the party-approval rule composing majoritarian portioning and  $M$  satisfies EJR and house monotonicity.*

*Proof.* Consider a party-approval election  $(N, P, A, k)$  and let  $r$  be the outcome of majoritarian portioning applied to  $(N, P, A)$ . Let  $N_1, N_2, \dots$  and  $p_1, p_2, \dots$  be the voter groups and parties in the construction of majoritarian portioning, so that  $r(p_j) = |N_j|/n$  for all  $j$ .

Consider the committee  $W = M(P, r, k)$  and suppose that EJR is violated, i.e., that there exists a group  $S \subseteq N$  with  $\bigcap_{i \in S} A_i \neq \emptyset$  and  $u_i(W) < q(S)$  for all  $i \in S$ .

Let  $j$  be minimal such that  $S \cap N_j \neq \emptyset$ . We now show that  $|S| \leq |N_j|$ . By the definition of  $j$ , no voter in  $S$  approves any of the parties  $p_1, p_2, \dots, p_{j-1}$ ; thus, all

those voters remain active in round  $j$ . Consider a party  $p^* \in \bigcap_{i \in S} A_i$ . In the  $j$ th iteration of majoritarian portioning, this party had an approval score (among active voters) of at least  $|S|$ . Therefore, the party  $p_j$  chosen in the  $j$ th iteration has an approval score that is at least  $|S|$  (of course,  $p^* = p_j$  is possible). The approval score of party  $p_j$  equals  $|N_j|$ . Therefore,  $|N_j| \geq |S|$ .

Since  $|N_j| \geq |S|$ , we have  $q(N_j) \geq q(S)$ . Since  $M$  satisfies lower quota, it assigns at least  $\lfloor k \cdot r(p_j) \rfloor = \lfloor k(|N_j|/n) \rfloor = q(N_j)$  seats to party  $p_j$ . Now consider a voter  $i \in S \cap N_j$ . Since this voter approves party  $p_j$ , we have  $u_i(W) \geq W(p_j) \geq q(N_j) \geq q(S)$ , a contradiction.

This shows that EJR is indeed satisfied; house monotonicity follows from the house monotonicity of  $M$ .  $\square$

While the party-approval rules identified by [Theorem 7.2](#) satisfy EJR and house monotonicity, they do not reach our gold standard of representation, i.e., core stability:

**Example 7.3** Let  $n = k = 16$ ,  $P = \{p_0, \dots, p_4\}$ , and consider the following ballot profile:

$$\begin{array}{lll} 4 \times \{p_0, p_1\}, & 3 \times \{p_1, p_2\}, & 1 \times \{p_2\} \\ 4 \times \{p_0, p_3\}, & 3 \times \{p_3, p_4\}, & 1 \times \{p_4\} \end{array}$$

Majoritarian portioning allocates  $1/2$  to  $p_0$  and  $1/4$  each to  $p_2$  and  $p_4$ . Any lower-quota apportionment method must translate this into 8 seats for  $p_0$  and 4 seats each for  $p_2$  and  $p_4$ . This committee is not in the core: Let  $S$  be the coalition of all 14 voters who approve multiple parties, and let  $T$  allocate 4 seats to  $p_0$  and 5 seats each to  $p_1$  and  $p_3$ . This gives strictly higher representation to all members of the coalition.

The example makes it obvious why majoritarian portioning cannot satisfy core stability: All voters approving of  $p_0$  get deactivated after the first round, which makes  $p_2$  seem universally preferable to  $p_1$ . However,  $p_1$  is a useful vehicle for cooperation between the group approving  $\{p_0, p_1\}$  and the group approving  $\{p_1, p_2\}$ . Since majoritarian portioning is blind to this opportunity, it cannot guarantee core stability.

The example also illustrates the power of core stability: The deviating coalition does not agree on any single party they support, but would nonetheless benefit from the deviation. Core stability is sensitive to this demand for better representation.

## 7.5 Computational Aspects

To use a voting rule, we need to compute its output. Ideally, we would like an efficient (i.e., polynomial-time) algorithm for this task, so that we can announce the voting outcome soon after all votes have been cast. Fortunately, many rules admit fast algorithms. For example, the composed rules from [Section 7.4.3](#) can be computed efficiently as long as the employed portioning and apportionment

methods are computable in polynomial time (which is the case for majoritarian partitioning as well as for D'Hondt and the quota method). In addition, by our discussion in Remark 7.1, every multiwinner voting rule that runs in polynomial time for the candidate-approval setting also runs in polynomial time for the party-approval setting.

That being said, given our result about core stability in Section 7.3, we are particularly interested in computing the outcome of PAV, which is NP-hard to compute in the candidate-approval setting [AGG+15]. Since party-approval elections are a restricted domain, it is in principle possible that PAV is easier to compute on that domain, but, as we show in Appendix A of the full version, hardness still holds for party-approval elections.

[AGG+15] Aziz et al. (2015): *Computational Aspects of Multi-Winner Approval Voting*.

**Theorem 7.3** *For a given party-approval election and threshold  $s \in \mathbb{R}$ , deciding whether there exists a committee with PAV score at least  $s$  is NP-hard.*

Equally confronted with the computational complexity of PAV, Aziz et al. [AEH+18] proposed a local-search variant of PAV, which runs in polynomial time and guarantees EJR in the candidate-approval setting. Using the same approach, we can find a core-stable committee in the party-approval setting.

[AEH+18] Aziz et al. (2018): *On the Complexity of Extended and Proportional Justified Representation*.

**Theorem 7.4** *Given a party-approval election, a core-stable committee can be computed in polynomial time.*

We defer the proof of this theorem to Appendix A of the full version. In Appendix B.1 of the full version, we additionally show that an optimization variant of Phragmén's rule [BFJL17] remains intractable in the party-approval subdomain.

[BFJL17] Brill et al. (2017): *Phragmén's Voting Methods and Justified Representation*.

Lackner and Skowron [LS22] posed as an open problem to determine the complexity of checking whether a given committee satisfies core stability. We show that the problem is coNP-complete. Our proof is written for party-approval elections, but the result implies hardness for the candidate-approval setting because party-approval elections are a special case of candidate-approval elections.

[LS22] Lackner and Skowron (2022): *Multi-Winner Voting with Approval Preferences*.

**Theorem 7.5** *For a given party-approval (or candidate-approval) election and a committee  $W$ , it is coNP-complete to decide whether  $W$  satisfies core stability.*

*Proof.* The complement problem is clearly in NP since a core deviation provides a certificate. We reduce from the NP-complete problem exact cover by 3-sets (X3C). Here, given a set  $X$  with  $|X| = 3r$  and a collection  $\mathcal{B}$  of 3-element subsets of  $X$ , the question is whether there exists a selection  $\mathcal{B}' \subseteq \mathcal{B}$  of  $r$  of the subsets such that every element of  $X$  occurs in one of the sets in  $\mathcal{B}'$ .

We construct an instance of our problem as follows: For every set  $B \in \mathcal{B}$  we introduce a *set candidate* and for every element in  $X$  we introduce an *element voter*. We set  $k = r$  and introduce one *special voter*,  $k - 1$  *private candidates* and one *dummy candidate*. The approval sets are as follows: Each element voter  $x \in X$  approves exactly those set candidates  $B \in \mathcal{B}$  with  $x \in B$  and the special voter approves all candidates except the dummy candidate. (Thus, no voter approves the dummy candidate.) Finally, let  $W$  be the committee consisting of the private

candidates and the dummy candidate.<sup>9</sup> Note that  $|W| = k$ . We claim that  $W$  is not core stable if and only if the  $X_3C$  instance is a *yes* instance.

Suppose that  $W$  is not core stable, and let  $S \subseteq N$  and committee  $T : P \rightarrow \mathbb{N}$  witness this fact. Without loss of generality, we may assume that  $T$  only gives seats to set candidates, since all other candidates are dominated by set candidates. Suppose  $|T| = t$ . Then  $T$  provides positive utility to at most  $3t$  element voters. These  $3t$  voters on their own can afford  $\lfloor 3t \cdot k/n \rfloor \leq 3t \cdot k/(3k+1) < t$  candidates. Because all element voters in  $S$  must obtain positive utility from  $T$ , it follows that the special voter must be part of  $S$ . Because the special voter  $i$  has  $u_i(T) > u_i(W) = k-1$ , we have  $u_i(T) = k$ . Thus  $|T| = k$ , and a committee of this size can only be afforded by the grand coalition, so  $S = N$ . Thus, every element voter is part of  $S$  and thus obtains positive utility from  $T$ , and hence for every element,  $T$  contains at least one set candidate corresponding to a set containing that element. It follows that the  $X_3C$  instance has a solution.

Conversely, every solution to the  $X_3C$  instance induces a committee  $T$  consisting of the  $k$  set candidates chosen by the solution. Then  $T$  gives positive utility to all element voters and increases the special voter's utility from  $k-1$  to  $k$ . Hence  $T$  together with  $S = N$  show that  $W$  is not core stable.  $\square$

In the candidate-approval setting, checking whether a given committee satisfies EJR is coNP-complete [ABC+17; AEH+18]. In other words, given a committee, it is hard to find a cohesive coalition of voters that is underrepresented. Interestingly, this task is tractable in party-approval elections. Intuitively, checking becomes easier in party-approval elections as groups of voters are already cohesive when they have only a single approved party in common.

**Theorem 7.6** *Given a party-approval election  $(N, P, A, k)$  and a committee  $W : P \rightarrow \mathbb{N}$ , it can be checked in polynomial time whether  $W$  satisfies EJR.*

*Proof.* We describe a procedure to check whether a given committee  $W$  violates EJR. For each party  $p \in P$  and each  $\ell \in [k]$ , define

$$S_{p,\ell} = \{i \in N \mid p \in A_i \text{ and } u_i(W) \leq \ell\}$$

and check whether  $\ell < q(S_{p,\ell})$  holds. If so, the set  $S_{p,\ell}$  induces an EJR violation. This is because  $\bigcap_{i \in S_{p,\ell}} A_i \neq \emptyset$  and  $u_i(W) \leq \ell < q(S_{p,\ell})$  holds for all  $i \in S_{p,\ell}$ .

Now, assume that the condition is not satisfied for any party  $p \in P$  and any  $\ell \in [k]$ . We claim that this proves the nonexistence of an EJR violation. Assume for contradiction that there exists a group  $S \subseteq N$  inducing an EJR violation. Let  $p \in \bigcap_{i \in S} A_i$  and  $\ell = \max_{i \in S} u_i(W)$ . By definition,  $S \subseteq S_{p,\ell}$  and hence  $q(S_{p,\ell}) \geq q(S) > \ell$ , a contradiction. A straightforward implementation of this algorithm has polynomial running time  $O(|P|kn)$ .  $\square$

We observe a similar effect for *proportional justified representation (PJR)*, a proportionality axiom introduced by Sánchez-Fernández et al. [SEL+17] which is weaker than EJR. While checking whether a committee satisfies PJR is coNP-complete in the candidate-approval setting, we can solve the problem in polynomial time

9: The committee  $W$  assigns seats only to Pareto-dominated parties, making it clearly suboptimal. One can adjust the reduction to show that the problem remains hard for committees  $W$  that do not give seats to Pareto-dominated parties.

[ABC+17] Aziz et al. (2017): *Justified Representation in Approval-Based Committee Voting*.

[AEH+18] Aziz et al. (2018): *On the Complexity of Extended and Proportional Justified Representation*.

[SEL+17] Sánchez-Fernández et al. (2017): *Proportional Justified Representation*.

via submodular minimization in our setting. For a formal definition and the proof, see [Appendix A.3](#) of the full version.

## 7.6 Discussion

In this chapter, we have initiated the axiomatic analysis of approval-based apportionment. On a technical level, it would be interesting to see whether the party-approval domain allows us to satisfy other combinations of axioms that are not known to be attainable in candidate-approval elections. For instance, the compatibility between strong representation axioms and certain notions of support monotonicity is an open problem [[SF19](#)].

We have presented our setting guided by the application of apportioning parliamentary seats to political parties. But our formal setting has other interesting applications. An example would be participatory budgeting settings where items all have equal costs and come in different types. For instance, a university department could decide how to allocate Ph.D. scholarships across different research projects, in a way that respects the preferences of funding organizations.

As another example, the literature on multiwinner elections suggests many applications to recommendation problems [[SFL16](#)]. For instance, one might want to display a limited number of news articles, movies, or advertisements in a way that fairly represents the preferences of the audience. These preferences might be expressed not over individual pieces of content, but over content producers (such as newspapers, studios, or advertising companies), in which case our setting provides rules that decide how many items should be contributed by each source. Expressing preferences on the level of content producers is natural in repeated settings, where the relevant pieces of content change too frequently to elicit voter preferences on each occasion. Besides, content producers might reserve the right to choose which of their content should be displayed.

In the general candidate-approval setting, the search continues for rules that satisfy EJR and house monotonicity, or core stability. But for the applications mentioned above, these guarantees are already achievable today.

[[SF19](#)] Sánchez-Fernández and Fisteus (2019): *Monotonicity Axioms in Approval-Based Multi-Winner Voting Rules*.

[[SFL16](#)] Skowron et al. (2016): *Finding a Collective Set of Items*.

# Monotone Randomized Apportionment

# 8

## 8.1 Introduction

The Constitution of the United States says that

“Representatives [in the US House of Representatives] shall be apportioned among the several States according to their respective numbers, counting the whole number of persons in each State ...”

These “respective numbers,” or populations, of the states are determined every decade through the census. For example, on April 1, 2020, the population of the United States was 331 108 434, and the state of New York had a population of 20 215 751. New York therefore deserves 6.105% of the 435 seats in the House, which is 26.56 seats, for the next ten years.

The puzzle of apportionment is what to do about New York’s 0.56 seat — in this round of apportionment it was rounded down to 0, and New York lost its 27th seat — or, more generally, how to allocate fractional seats. This mathematical question has riveted the American political establishment since the country’s founding [Szpi10].

In 1792, Congress approved a bill that would enact an apportionment method proposed by Alexander Hamilton,<sup>1</sup> the first secretary of the treasury and star of the eponymous musical. If we denote the *standard quota* of state  $i$  by  $q_i$  ( $q_i = 26.56$  in the case of New York in 2020), Hamilton’s method allocates to each state its *lower quota*  $\lfloor q_i \rfloor$  (26 for NY). Then, Hamilton’s method goes through the states in order of decreasing *residue*  $q_i - \lfloor q_i \rfloor$  (0.56 for NY) and allocates an additional seat to each state until all house seats are allocated.

As sensible as Hamilton’s method appears, it repeatedly led to bizarre results, which became known as *apportionment paradoxes*.

**The Alabama paradox:** Using the 1880 census results, the chief clerk of the Census Office calculated the apportionment according to Hamilton’s method for all House sizes between 275 and 350, and discovered that, as the size increased from 299 to 300, Alabama lost a seat. In 1900, the Alabama paradox reappeared, this time affecting Colorado and Maine.

**The population paradox:** In 1900, the populations of Virginia and Maine were 1 854 184 and 694 466, respectively. Over the following year, the populations of the two states grew by 19 767 and 4 649, respectively. Even though Virginia’s growth was larger even relative to its population, Hamilton’s method would have transferred a seat from Virginia to Maine.

Past occurrences of these paradoxes invited partisan strife, which is only natural since a state’s representatives have a strong personal stake in their state not losing seats. Both in Congress and the courts, this strife took the form of a tug-of-war over the choice of apportionment method, the size of the House,<sup>2</sup> and the census

8.1	Introduction	110
8.1.1	Approach and Results	112
8.1.2	Related Work	113
8.2	Model	114
8.3	Population Monotonicity	116
8.3.1	Impossibility of Population Monotonicity & Quota	116
8.3.2	Population Monotonicity & Ex Ante Proportionality	117
8.4	House Monotonicity	119
8.4.1	Pitfalls	119
8.4.2	Cumulative Rounding	120
8.4.3	House Monotonicity, Quota & Ex Ante Proportionality	122
8.5	Cumulative Rounding Proof	125
8.6	Cumulative Rounding Applications	128
8.6.1	Sortition of the European Commission	128
8.6.2	Teaching or Work Shift Allocation	130
8.7	Discussion	131

[Szpi10] Szpiro (2010): *Numbers Rule*.

1: In fact, the bill was vetoed by George Washington and Hamilton’s method was only adopted in 1850.

2: For a long time, the House kept growing such that no state ever lost a seat, even though the influence of each seat diminished.

numbers, driven by the states’ parties’ and individual representatives’ self-interest rather than the public good.

This state of affairs improved in 1941 when Congress adopted an apportionment method that provably avoids the Alabama and population paradoxes, which had been developed by Edward Huntington, a Harvard mathematician, and Joseph Hill, the chief statistician of the Census Bureau. While the Huntington–Hill method is *house monotone* (i.e., it avoids the Alabama paradox) and *population monotone* (i.e., it avoids the population paradox), it has a different, equally bizarre weakness: it does not *satisfy quota*, that is, the allocation of some states may be different from  $\lfloor q_i \rfloor$  or  $\lceil q_i \rceil$ . A striking impossibility result by Balinski and Young [BY01] shows that this tension is inevitable: no apportionment method can simultaneously satisfy quota and be population monotone.<sup>3</sup>

While the Balinski–Young impossibility is troubling, in our view there is an even larger source of unfairness that plagues apportionment methods, which is rooted in their determinism. In addition to introducing bias (the Huntington–Hill method disadvantages larger states), deterministic methods often lead to situations where small counting errors can change the outcome.<sup>4</sup> For example, based on the 2020 census, New York lost its 27th House seat, but it would have kept it had its population count been higher by 89 residents! After the 1990 and 2000 censuses, similar circumstances were the basis for lawsuits brought by Massachusetts and Utah.

To address these issues, an obvious solution is to use randomization in order to realize the standard quota of each state in expectation, as Grimmett proposed in 2004 [Gri04]. If such a randomized method was used, 89 additional residents would have shifted New York’s expected number of seats by a negligible 0.0001, and the decision between 26 or 27 seats would have been made by an impartial random process, which is less accessible to political maneuvering than, say, the census [Sto11].

Grimmett’s proposed apportionment method is easy to describe. First, we choose a random permutation of the states; without loss of generality, that permutation is identity. Second, we draw  $U$  uniformly at random from  $[0, 1]$ , and let  $Q_i := U + \sum_{j=1}^i q_j$ . Finally, we allocate to each state  $i$  one seat for each integer contained in the interval  $[Q_{i-1}, Q_i)$ . (In particular, this implies that the allocation will satisfy quota.)

Why this particular method? Grimmett writes [Gri04, p. 302]:

“We offer no justification for this scheme apart from fairness and ease of implementation.”

Grimmett’s method is easy to implement for sure, and what he refers to as “fairness” — realizing the fractional quotas in expectation — is arguably a minimal requirement for any randomized apportionment method. But his two axioms, “fairness” and quota, allow for a vast number of randomized methods: Indeed, after allocating  $\lfloor q_i \rfloor$  seats to each agent, the problem of determining which states to round up reduces to so-called “ $\pi$ ps sampling” (“inclusion probability proportional to size”), and dozens of such schemes have been proposed in the literature [BH83]. We believe, therefore, that additional criteria are needed to guide the design of randomized apportionment methods. To identify such criteria, we return to the classics: house and population monotonicity.

[BY01] Balinski and Young (2001): *Fair Representation*.

3: We will revisit this result in Section 8.3 and show that, while Balinski and Young’s theorem makes additional implicit assumptions, the incompatibility between quota and population monotonicity continues to hold without these assumptions.

4: A second shortcoming of deterministic apportionment methods is a lack of fairness over time: For example, if the states’ populations remain static, a state with a standard quota of, say, 1.5 might receive a single seat in every single apportionment and therefore only receive 2/3 of its deserved representation. Using randomized apportionment, the long-term average of a state’s number of seats is proportional to the state’s average share of the total population.

[Gri04] Grimmett (2004): *Stochastic Apportionment*.

[Sto11] Stone (2011): *The Luck of the Draw*.

[BH83] Brewer and Hanif (1983): *An Introduction to Sampling with Unequal Probabilities*.

### 8.1.1 Our Approach and Results

In this paper, we seek randomized apportionment methods that satisfy natural extensions of house and population monotonicity to the randomized setting. We want these monotonicity axioms to hold even *ex post*, i.e., after the randomization has been realized. We find such methods by taking a parameterized class of deterministic methods all of which satisfy the desired *ex post* axioms (in our case, subsets of population monotonicity, house monotonicity, and quota), and to then randomize over the choice of parameters such that *ex ante* properties hold (here: *ex ante* proportionality).<sup>5</sup>

Guaranteeing monotonicity axioms *ex post* is helpful for preventing certain kinds of manipulation in the apportionment process. For instance, say that the census concludes and a randomized apportionment is determined, and only then does a state credibly contest that its population was undercounted (in the courts or in Congress with the support of a majority). Using an apportionment method without population monotonicity, states might strategically undercount their population in the census and only reveal the true count in case this turns out to be beneficial once the randomness is revealed. When using a population monotone method, by contrast, any revised apportionment would be made using the same deterministic and population monotone method, which implies that immediately revealing the full population count is a dominant strategy, even for coalitions of states.

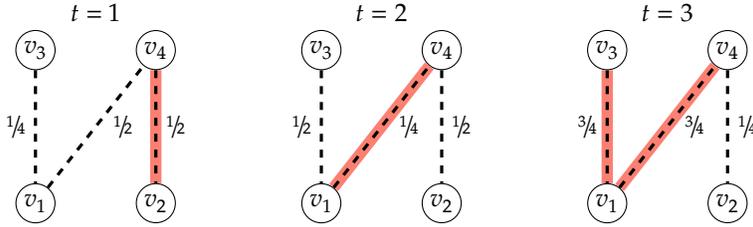
In [Section 8.3](#), we first show that no such randomized methods exist for population monotonicity. This impossibility is not due to randomization or *ex ante* proportionality, but due to the fact that population monotonicity and quota are outright incompatible. Thus, there do not exist suitable deterministic apportionment methods that a randomized apportionment method could randomize over. That population monotonicity and quota are incompatible is well-known from the Balinski–Young impossibility theorem [\[BY01\]](#), but their proof uses some “mild” background conditions (notably neutrality), which are not mild for our randomized purposes. We are able to prove a stronger version of their theorem, which derives the impossibility with no assumptions other than population monotonicity and quota. The deterministic apportionment methods that are most commonly used in practice (so called divisor methods, including the Huntington–Hill method) satisfy population monotonicity but fail quota. So it makes sense to ask whether population monotonicity can be combined with *ex ante* proportionality (without requiring quota). We construct such a method, which is reminiscent of the family of divisor methods, except that the so-called “divisor criterion” [\[BY01\]](#) is specific to each state and is given by a sequence of Poisson arrivals.

For house monotonicity, we provide in [Section 8.4](#) a randomized apportionment method that satisfies house monotonicity, quota, and *ex ante* proportionality. To obtain this result, we generalize the classic result of Gandhi et al. [\[GKPS06\]](#) on dependent rounding in a bipartite graph. We call this method *cumulative dependent randomized rounding* or just *cumulative rounding* ([Theorem 8.4](#)). Cumulative rounding allows to correlate dependent-rounding processes in multiple copies of the same bipartite graph such that the result satisfies an additional guarantee across copies of the graph. This guarantee, which we describe in the next paragraph, generalizes the quota axiom of apportionment. As a side product,

5: In mechanism design, a similar approach extends strategyproofness to universal strategyproofness [\[NR01\]](#).

[\[BY01\]](#) Balinski and Young (2001): *Fair Representation*.

[\[GKPS06\]](#) Gandhi et al. (2006): *Dependent Rounding and Its Applications to Approximation Algorithms*.



**Figure 8.1:** Illustration of cumulative rounding. Dashed lines indicate edges  $e \in E$  in the bipartite graph  $(V, E)$ , which are labeled with weights  $w_e^t$ . The red lines indicate a possible random outcome of cumulative rounding.

our existence proof for house monotonicity provides a new characterization of the deterministic apportionment methods satisfying house monotonicity and quota, which is based on the corner points of a bipartite matching polytope.

To describe cumulative rounding more precisely, we first sketch the result of Gandhi et al. [GKPS06]. For a bipartite graph  $(V, E)$  and edge weights  $\{w_e\}_{e \in E}$  in  $[0, 1]$ , dependent rounding randomly generates a subgraph  $(V, E')$  with  $E' \subseteq E$  providing three properties: *marginal distribution* (each edge  $e \in E$  is contained in  $E'$  with probability  $w_e$ ), *degree preservation* (in the rounded graph, the degree of a vertex  $v$  is the floor or the ceiling of  $v$ 's fractional degree  $\sum_{e \in E} w_e$ ), and *negative correlation*. Cumulative rounding allows us to randomly round  $T$  many copies of  $(V, E)$ , where each copy  $1 \leq t \leq T$  has its own set of weights  $\{w_e^t\}_{e \in E}$ . Each copy will provide marginal distribution, degree preservation, and negative correlation. As we prove in Section 8.5, cumulative rounding additionally guarantees *cumulative degree preservation*: for each vertex  $v$  and  $1 \leq t \leq T$ , the sum of degrees of  $v$  across copies 1 through  $t$  equals the sum of fractional degrees of  $v$  across copies 1 through  $t$ , either rounded up or down. For example, node  $v_1$  in Figure 8.1 is incident to edges with a total fractional weight of  $2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{2} = 1.5$  across copies  $t = 1, 2$ , and must hence be incident to 1 or 2 edges in total across the rounded versions of copies  $t = 1, 2$ . Since, across copies  $t = 1, 2, 3$ ,  $v_1$ 's total fractional degree is  $1.5 + 2 \cdot \frac{3}{4} = 3$ ,  $v_1$  must be incident to a total of exactly 3 rounded edges across the copies  $t = 1, 2, 3$ . Applying cumulative rounding to a star graph yields the desired randomized apportionment method satisfying house monotonicity, quota, and ex ante proportionality.

We believe that cumulative rounding is of broader interest, and in Section 8.6, we present applications of cumulative rounding beyond apportionment. First, we look at a proposal of Buchstein and Hein [BH09] for a reform of the European Commission of the European Union: They propose to use a weighted lottery to choose which countries get to nominate commissioners. Using cumulative rounding to implement this lottery would eliminate two key problems the authors identified in a simulation study, in particular the possibility that some member states might go without any commissioners for a long period of time. We also describe how cumulative rounding can be applied to round fractional allocations of goods or chores, and we discuss a specific application of assigning faculty to teach courses.

### 8.1.2 Related Work

Randomized apportionment was first suggested by Grimmett [Gri04], whose proposal we have already discussed. More recently, Aziz et al. [ALM+19] developed a randomized rounding scheme as part of a mechanism for strategy-proof peer

[GKPS06] Gandhi et al. (2006): *Dependent Rounding and Its Applications to Approximation Algorithms*.

[BH09] Buchstein and Hein (2009): *Randomizing Europe*.

[Gri04] Grimmett (2004): *Stochastic Apportionment*.

[ALM+19] Aziz et al. (2019): *Strategyproof Peer Selection Using Randomization, Partitioning, and Apportionment*.

selection, which they simultaneously propose as a randomized apportionment method. Just like Grimmett’s method, their method satisfies ex ante proportionality and quota. Aziz et al. argue that the main advantage of their method is that its support consists of only linearly (not exponentially) many deterministic apportionments. This, they claim, is useful in repeated apportionment settings, where one could repeat a periodic sequence of these deterministic apportionments and thereby limit the possibility of selecting the same state much too frequently or much too rarely due to random fluctuations. If this is the goal, cumulative rounding will arguably give better guarantees (see Section 8.6.1).

As a consequence of the Birkhoff–von Neumann Theorem [Bir46; vNeu53], any fractional matching in a bipartite graph can be implemented as a lottery over integral matchings, in the sense that each edge is present in the random matching with probability equal to its weight in the fractional matching. One algorithm for rounding a bipartite matching is pipage rounding [ASo4], which Gandhi et al. [GKPSo6] randomized in their dependent rounding technique. This rounding technique is powerful since it can directly accommodate fractional degrees larger than 1 and can provide negative-correlation properties such that Chernoff concentration bounds apply [PS97]. The technique of Gandhi et al. has found many applications in approximation algorithms [BGL+12; GKPSo6; KMPSo9] and in fair division [AN20; CJMW19; SS18a].

Steiner and Yeomans [SY93] study a problem in just-in-time industrial manufacturing: how to alternate between the production of different types of goods in a way that produces each type in specified proportions. As pointed out by Bautista et al. [BCC96], this problem is related to apportionment. In particular, a production schedule resembles a deterministic house monotone apportionment method: as the available production time increases by one slot, the schedule needs to decide which type to produce in the next slot. Steiner and Yeomans end up with a property that nearly guarantees quota because they aim to minimize how far the prevalence of types among the goods produced so far deviates from the desired proportions. Now, they only produce deterministic schedules, and the existence of deterministic house monotone and quota apportionment methods has long been known [BY75; Sti79]. But we believe that the main construction in their proof could be randomized to obtain an alternative proof of Theorem 8.6, without however providing the generality of cumulative rounding. In fact, a similar graph construction to that by Steiner and Yeomans is randomly rounded within a proof by Gandhi et al. [GKPSo6] to obtain an approximation result about broadcast scheduling.

## 8.2 Model

Throughout this paper, fix a set of  $n \geq 2$  states  $N = \{1, 2, \dots, n\}$ . For a given population profile  $\vec{p} \in \mathbb{N}_+^n$ , which assigns a population of  $p_i \in \mathbb{N}_+$  to each state  $i$ , and for a house size  $h \in \mathbb{N}_+$ , an apportionment solution deterministically allocates to each state  $i$  a number  $a_i \in \mathbb{N}$  of house seats such that the total number of allocated seats is  $h$ . Formally, a solution is a function  $f : \mathbb{N}_+^n \times \mathbb{N}_+ \rightarrow \mathbb{N}^n$  such that, for all  $\vec{p}$  and  $h$ ,  $\sum_{i \in N} f_i(\vec{p}, h) = h$ . For a population profile  $\vec{p}$  and house

[Bir46] Birkhoff (1946): *Three Observations on Linear Algebra*.

[vNeu53] von Neumann (1953): *A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem*.

[ASo4] Ageev and Sviridenko (2004): *Pipage Rounding*.

[GKPSo6] Gandhi et al. (2006): *Dependent Rounding and Its Applications to Approximation Algorithms*.

[PS97] Panconesi and Srinivasan (1997): *Randomized Distributed Edge Coloring via an Extension of the Chernoff–Hoeffding Bounds*.

[BGL+12] Bansal et al. (2012): *When LP Is the Cure for Your Matching Woes*.

[KMPSo9] Kumar et al. (2009): *A Unified Approach to Scheduling on Unrelated Parallel Machines*.

[AN20] Akbarpour and Nikzad (2020): *Approximate Random Allocation Mechanisms*.

[CJMW19] Cheng et al. (2019): *Group Fairness in Committee Selection*.

[SS18a] Saha and Srinivasan (2018): *A New Approximation Technique for Resource-Allocation Problems*.

[SY93] Steiner and Yeomans (1993): *Level Schedules for Mixed-Model, Just-in-Time Processes*.

[BCC96] Bautista et al. (1996): *A Note on the Relation between the Product Rate Variation (PRV) Problem and the Apportionment Problem*.

[BY75] Balinski and Young (1975): *The Quota Method of Apportionment*.

[Sti79] Still (1979): *A Class of New Methods for Congressional Apportionment*.

size  $h$ , state  $i$ 's *standard quota* is  $q_i := \frac{p_i}{\sum_{i \in N} p_i} h$ . Next, we define three axioms for solutions:

**Quota:** A solution  $f$  satisfies quota if, for any  $\vec{p}$  and  $h$ , it holds that  $f_i(\vec{p}, h) \in \{\lfloor q_i \rfloor, \lceil q_i \rceil\}$  for all states  $i$ .

**House monotonicity:** A solution  $f$  satisfies house monotonicity if, for any  $\vec{p}$  and  $h$ , increasing the house size to  $h + 1$  does not reduce any state's seat number, i.e., if  $f_i(\vec{p}, h) \leq f_i(\vec{p}, h + 1)$  for all  $i \in N$ .

**Population monotonicity:**<sup>6</sup> We say that a solution  $f$ , some  $\vec{p}, \vec{p}' \in \mathbb{N}_+^n$ , and some  $h, h' \in \mathbb{N}_+$  exhibit a *population paradox* if there are two states  $i \neq j$  such that  $p'_i \geq p_i$ ,  $p'_j \leq p_j$ ,  $f_i(\vec{p}', h') < f_i(\vec{p}, h)$ , and  $f_j(\vec{p}', h') > f_j(\vec{p}, h)$ , or, in words, if state  $i$  loses seats and  $j$  wins seats even though  $i$ 's population weakly grew and  $j$ 's population weakly shrunk. A solution  $f$  is *population monotone* if it exhibits no population paradoxes for any  $\vec{p}, \vec{p}', h, h'$ . By setting  $\vec{p} = \vec{p}'$ , one easily verifies that population monotonicity implies house monotonicity.

Finally, we will define randomized apportionment methods. One potential definition, used by Grimmett [Gri04], is a function that for each  $\vec{p}$  and  $h$  specifies a probability distribution over seat allocations  $(a_i)_{i \in N}$ . Instead, we are looking for a random process whose outcome  $\omega \in \Omega$  *simultaneously* determines apportionments for all population vectors  $\vec{p}$  and house sizes  $h$ , which will allow us to formulate axioms relating these different apportionments. Conceptually, we think of such a method as a solution-valued random variable. Formally, a randomized apportionment method, which we will just call a *method*, consists of a probability space  $\Omega = (\Omega, \mathcal{F}, \mathbb{P})$  and a function  $F$  mapping elements of  $\Omega$  to solutions such that, for all  $\vec{p}$  and  $h$ ,  $F(\vec{p}, h)$  is a random vector specifying the seat allocation. Typically, we will not need to think about the internal structure of  $\Omega$  and therefore leave it implicit. Using a programming metaphor, the “randomness” of a program is really determined by an implicit *random seed*. We can think of a method  $F$  as a procedure that is initialized with a seed  $\omega$  and then takes  $\vec{p}$  and  $h$  as its input in order to return an apportionment. When  $\omega$  is chosen at random, then  $F$  behaves as a random procedure, but for any fixed  $\omega$ ,  $F_\omega$  is just a deterministic procedure mapping  $\vec{p}, h$  to apportionments.<sup>7</sup> Our axioms, described in the next paragraph, constrain both the random behavior of  $F$  and the consistency of any  $F_\omega$  across inputs.

A method  $F$  satisfies *ex ante proportionality* if, for any  $\vec{p}, h$  and for any state  $i$ ,  $i$ 's expected number of seats equals  $i$ 's standard quota, i.e., if  $\mathbb{E}[F_i(\vec{p}, h)] = q_i$ . A method  $F$  satisfies *quota*, *house monotonicity*, or *population monotonicity* if all solutions in the method's support satisfy the respective axiom. In this paper, we mainly search for apportionment methods that combine quota and ex ante proportionality — the two axioms obtained by Grimmett [Gri04] — with either population or house monotonicity.

6: This definition of population monotonicity, taken from Robinson and Ullman [RU10], is slightly weaker than the definition of other authors, whose violation we describe in the introduction. All results extend to the alternative notion of *relative population monotonicity* [RU10]: the proof of Theorem 8.1 immediately applies, and the proof of Theorem 8.2 is easy to adapt.

[Gri04] Grimmett (2004): *Stochastic Apportionment*.

7: This is also how we would implement an apportionment method on a computer. A seed obtained using physical randomness would determine the solution, and the solution would be computed from the seed using a pseudo-random number generator.

state $i$	profile $\vec{p}^A$		profile $\vec{p}^B$		profile $\vec{p}^C$	
	$p_i^A$	$q_i^A$	$p_i^B$	$q_i^B$	$p_i^C$	$q_i^C$
1	824	8.24	824	6.99	824	9.02
2	44	0.44	44	0.37	1	0.01
3	44	0.44	44	0.37	1	0.01
4	44	0.44	44	0.37	44	0.48
5	44	0.44	222	1.88	44	0.48

**Table 8.1:** Populations and standard quotas for three population profiles, used in showing that population monotonicity and quota are incompatible. The house size is  $h = 10$ .

## 8.3 Population Monotonicity

### 8.3.1 Population Monotonicity Is Incompatible with Quota

We begin by showing that no apportionment method satisfies population monotonicity, quota, and ex ante proportionality. In fact, quota and population monotonicity alone are incompatible: We will show that no solution satisfies these two axioms. Since a method satisfying quota and population monotonicity would be a random choice over such solutions, no such method exists either.

At first glance, the incompatibility of quota and population monotonicity might seem to follow from existing results, but these results implicitly make neutrality assumptions that are not appropriate for randomized apportionment. Indeed, Balinski and Young [BY01], who originally proved this incompatibility, as well as variations of their proof [EL-19; RU10] all assume what Robinson and Ullman [RU10] call the *order-preserving property*, i.e., if state  $i$  has strictly larger population than state  $j$ , then  $i$  must receive at least as many seats as  $j$ . This property is usually proved as a consequence of neutrality together with population monotonicity.

While the order-preserving property is reasonable for developing deterministic apportionment methods, it is not desirable for the component solutions of a randomized apportionment method. This is clear for  $h = 1$ : The order-preserving property would mean that only the very largest state(s) can get a seat with positive probability; by contrast, the strength of randomization is that it allows us to allocate the seat to smaller states. To our knowledge, the existence of quota and population monotone solutions without the assumption of the order-preserving property was an open problem.

**Theorem 8.1** *No (deterministic) apportionment solution satisfies population monotonicity and quota.*

*Proof.* Fix a set of 5 states, and let  $f$  be a solution satisfying quota. We will show that  $f$  must violate population monotonicity by analyzing three types of population profiles, which are given in Table 8.1, all for house size  $h = 10$ . The starting profile is  $\vec{p}^A$  in this table. By quota, state 1 must receive either 8 or 9 seats on this profile, but we will show that either choice leads to a violation of population monotonicity: First, we show that allocating 9 seats implies a violation of population monotonicity with respect to profile  $\vec{p}^B$ ; second, we show that allocating 8 seats contradicts population monotonicity with respect to  $\vec{p}^C$ .

[BY01] Balinski and Young (2001): *Fair Representation*.

[EL-19] El-Helaly (2019): *The Mathematics of Voting and Apportionment*.

[RU10] Robinson and Ullman (2010): *A Mathematical Look at Politics*.

*Allocating 9 seats contradicts population monotonicity.* Suppose that  $f_1(\vec{p}^A, 10) = 9$ . Then, the remaining seat must be given to either state 2, 3, 4, or 5. Without loss of generality, we may assume that  $f(\vec{p}^A, 10) = (9, 0, 0, 0, 1)$ .

Next, consider the profile  $\vec{p}^B$ . Since quota prevents us from allocating more than 7 seats to state 1 or more than 2 seats to state 5, at least one of the states 2, 3, and 4 must receive a seat on  $\vec{p}^B$ . Thus, this state's allocation strictly increases from its allocation of zero seats on  $\vec{p}^A$ , even though the state's population has not changed. Moreover, state 1 can receive at most 7 seats on this profile by quota, which is strictly below the 9 seats on  $\vec{p}^A$ , and state 1's population has also remained the same. But population monotonicity forbids there to be a pair of states with unchanged population, such that one gains a seat and the other loses a seat. Hence, if state 1 receives 9 seats on  $\vec{p}^A$ , then  $f$  violates population monotonicity.

*Allocating 8 seats contradicts population monotonicity.* Suppose that  $f_1(\vec{p}^A, 10) = 8$ . The remaining two seats must be given to two states out of 2, 3, 4, and 5; without loss of generality, we may assume that  $f(\vec{p}^A, 10) = (8, 0, 0, 1, 1)$ .

On profile  $\vec{p}^C$ , quota implies that state 1 receives at least 9 seats — strictly more than the 8 given on  $\vec{p}^A$  even though the population has not changed. Given that there is at most one more seat to hand out, at least one state out of 4 and 5 must receive zero seats on  $\vec{p}^C$ , which is a strict reduction with respect to  $\vec{p}^A$  even though the state's population is the same. Thus, allocating 8 seats to state 1 on  $\vec{p}^A$  also leads to a violation of population monotonicity.

Since both possible choices for  $f_1(\vec{p}^A, 10)$  imply a monotonicity violation, no solution can satisfy both quota and population monotonicity.  $\square$

### 8.3.2 A Population Monotone and Ex Ante Proportional (But Not Quota) Method

The incompatibility between population monotonicity and quota leaves open the question of whether there are apportionment methods satisfying population monotonicity and ex ante proportionality. The answer is positive, as the following proposition shows:

**Theorem 8.2** *There exists an apportionment method  $F$  that satisfies population monotonicity and ex ante proportionality.*

*Proof.* Which solution is randomly chosen by the method will depend on the values taken on by  $n$  independent Poisson arrival processes with rate 1.<sup>8</sup> We fix an outcome  $\omega \in \Omega$  and will construct a solution  $F(\omega)$ . For each state  $i$ ,  $\omega$  determines an infinite sequence  $0 < x_1^i < x_2^i < \dots$  of arrival times. We will describe the apportionment given by  $F(\omega)$  on input  $\vec{p}$  and  $h$ , which we illustrate in Figure 8.2: First, we divide each arrival time  $x_t^i$  by the corresponding state's population, i.e., we set  $y_t^i := x_t^i/p_i$ . Second, we combine the  $y_t^i$  for all  $t$  and  $i$  in a single arrival sequence  $(z_1, i_1), (z_2, i_2), \dots$  labeled with states, i.e., each  $(z_j, i_j)$  corresponds to some arrival  $y_t^i$  for some  $i$  and  $t$ , such that  $z_j = y_t^i$  is the arrival time,  $i_j = i$  is the agent label, and the  $z_j$  are sorted in increasing order. Third, we

8: Hence, we can select  $\Omega$  as the probability space used in any standard construction [Kin93] of a random variable denoting a vector whose  $n$  components are the outcomes of independent Poisson processes.

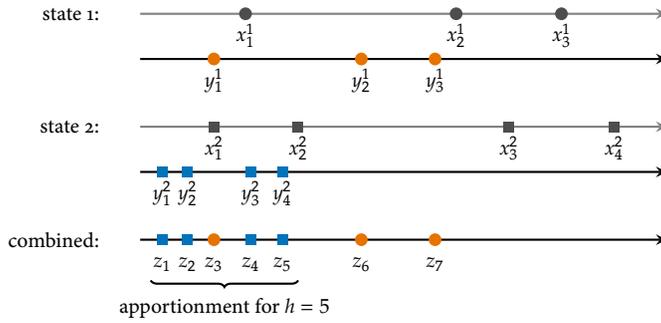


Figure 8.2: Illustration of the population-monotone method in Theorem 8.2.

allocate  $|\{1 \leq j \leq h \mid i_j = i\}|$  many seats to each state  $i$ , i.e., a number of seats equal to how many among the  $h$  smallest scaled arrival times belonged to  $i$ 's arrival process. This specifies the solution  $F(\omega)$ , and, moreover, the method  $F$ .

First, we show that  $F$  satisfies ex ante proportionality. For this, fix some  $\vec{p}$  and  $h$ . Then, the  $\{y_t^i\}_{t \geq 1}$  for each  $i$  are distributed as the arrival sequences of independent Poisson processes, where  $i$ 's arrival process has a rate of  $p_i$ . By the coloring theorem for Poisson processes [Kin93, p. 53], our labeled arrival sequence  $(z_j, i_j)$  has the same distribution as if we had sampled a Poisson arrival process  $0 < z_1 < z_2 < \dots$  with arrival rate  $\sum_{i \in N} p_i$  and had drawn each  $i_j$  independently, choosing each  $i \in N$  with probability proportional to  $p_i$ . Since the  $z_j$  and  $i_j$  are independent in this way,  $F(\vec{p}, h)$  is distributed as if sampling  $h$  states, with probability proportional to the states' populations and with replacement. In particular, this implies ex ante proportionality.

[Kin93] Kingman (1993): *Poisson Processes*.

It remains to show that  $F$  satisfies population monotonicity. Fix an  $\omega$ , i.e., the  $x_t^i$ , as well as two inputs  $\vec{p}, h$  and  $\vec{p}', h'$ , for which we will show that  $F(\omega)$  does not exhibit a population paradox. Denoting the inputs' respective variables by  $y_t^i, z_j$  and  $y_t^{i'}, z'_j$ , it is easy to see that, for all  $i$  for which  $p'_i \geq p_i$ ,  $y_t^{i'} \leq y_t^i$  for all  $t$ , and that, for all  $i$  for which  $p'_i \leq p_i$ ,  $y_t^{i'} \geq y_t^i$  for all  $t$ . Observe that each state  $i$  receives a number of seats equal to the number of its scaled arrival times  $y_t^i$  (resp.,  $y_t^{i'}$ ) that are at most  $z_h$  (resp.,  $z'_h$ ).

Suppose that  $z'_h \geq z_h$  (the reasoning for the case  $z'_h \leq z_h$  is symmetric). Then, whenever  $y_t^i \leq z_h$  for a state  $i$  for which  $p'_i \geq p_i$ , then  $y_t^{i'} \leq y_t^i \leq z_h \leq z'_h$ , which shows that  $i$ 's seat number must weakly increase. One verifies that this rules out a population paradox on  $\vec{p}, h$  and  $\vec{p}', h'$ . Together with the symmetric argument for  $z'_h \leq z_h$ , this establishes population monotonicity.  $\square$

Though the apportionment solutions used in the last theorem might seem esoteric, it is interesting to compare them to *divisor methods* (for consistency with our terminology, *divisor solutions*), which, under widely assumed regularity assumptions, exactly characterize the space of all population monotone solutions [BY01]. A divisor solution is characterized by a *divisor criterion*, which is a monotone increasing function  $d : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$  such that, for all  $t \in \mathbb{N}$ ,  $t \leq d(t) \leq t + 1$ . For instance, the Huntington-Hill solution is induced by  $d(t) := \sqrt{t(t+1)}$ . For a population profile  $\vec{p}$  and house size  $h$ , the divisor solution corresponding to  $d$  can be calculated by considering the sets  $\{p_i/d(t) \mid t \in \mathbb{N}\}$  for each state  $i$ , determining the  $h$  largest values across all sets, and allocating to each state  $i$  a number of seats equal to how many of the  $h$  largest values came from  $i$ 's set. The

[BY01] Balinski and Young (2001): *Fair Representation*.

solutions in the above proof could have been cast in similar terms, where state  $i$ 's set is  $\{1/y_i^t \mid t \in \mathbb{N}\} = \{p_i/x_i^t \mid t \in \mathbb{N}\}$ , i.e., where, for each state  $i$ ,  $t \mapsto x_i^t$  plays the role of a state-specific divisor criterion.

Clearly, the solutions' resemblance to divisor solutions enabled our proof of population monotonicity. At the same time, using different "divisor criteria" for different states allowed to avoid the order-preserving property, which would have prevented ex ante proportionality as described in Section 8.3.1. Less satisfying is that these "divisor criteria" do not satisfy any bounds such as  $t \leq d(t) \leq t + 1$ , which makes it likely that solutions substantially deviate from proportionality ex post. An interesting question for future work is whether Theorem 8.2 can be strengthened to additionally satisfy lower quota (" $F_i(\vec{p}, h) \geq \lfloor q_i \rfloor$ ") or upper quota (" $F_i(\vec{p}, h) \leq \lceil q_i \rceil$ ").

## 8.4 House Monotonicity

While we cannot obtain population monotonicity without giving up on quota, we now propose an apportionment method that combines *house monotonicity* with quota and ex ante proportionality.

### 8.4.1 Examples of Pitfalls

An intuitive strategy for constructing a house monotone randomized apportionment methods is to do it inductively, seat-by-seat. Thus, we would need a strategy for extending a method that works for all house sizes  $h' \leq h$  to a method that also works for house size  $h + 1$ . In this section, we give examples suggesting that this does not work, by showing that some reasonable methods cannot be extended without violating quota or ex ante proportionality. This motivates a search for a more "global" strategy for constructing a house-monotone method.

Our first example will show that there are apportionments for a given  $h$  that satisfy quota, but that are "toxic" in that they can never be chosen by a house monotone solution which satisfies quota:

**Example 8.1** Suppose that we have four states with populations  $\vec{p} = (1, 2, 1, 2)$ . The distribution that we will consider is the one given by Grimmett's method [Grio4] (as described in the introduction) for these inputs.<sup>9</sup> Let  $h = 2$ . Observe that, if the random permutation chosen by Grimmett's method is identity and if furthermore  $U > 2/3$ , then Grimmett's method will return the allocation  $(1, 0, 1, 0)$ . But we will show that no solution  $f$  such that  $f(\vec{p}, 2) = (1, 0, 1, 0)$  can satisfy house monotonicity and quota. Indeed, if  $f$  is house monotone, then at least one out of state 2 or state 4 must still be given zero seats by  $f$  when  $h = 3$ , but quota requires that both states receive exactly one seat when  $h = 3$ . It follows that Grimmett's method, or any other method satisfying quota and whose support contains solutions  $f$  with  $f(\vec{p}, 2) = (1, 0, 1, 0)$ , cannot be house monotone.

[Grio4] Grimmett (2004): *Stochastic Apportionment*.

9: It is easy to correlate an outcome for  $h = 1$  with this distribution in a way that preserves house monotonicity: Draw an apportionment  $\vec{a}$  from the  $h = 2$  distribution and then flip a coin to determine if the seat for  $h = 1$  should go to the smaller or the larger one of the states receiving a seat in  $\vec{a}$ . This satisfies quota, ex ante proportionality, and house monotonicity across the inputs  $(\vec{p}, 1)$  and  $(\vec{p}, 2)$ .

Thus, a first challenge that any quota and house monotone method must overcome is to never produce a toxic apportionment for a specific  $h$  that cannot be extended to all larger house sizes in a house monotone and quota-compliant way. Still [Sti79] and later Balinski and Young [BY79] give a characterization of non-toxic apportionments, but we found no way of transforming this characterization into an apportionment method that would be ex ante proportional.

Our second example shows that, even if there are no toxic apportionments in the support of a distribution, the wrong distribution over apportionments might still lead to violations of one of the axioms:

**Example 8.2** Let there be four states with populations  $\vec{p} = (45, 25, 15, 15)$  and let  $h = 3$ ; thus, the standard quotas are  $(1.35, 0.75, 0.45, 0.45)$ . We consider the following distribution over allocations:

$$\vec{a} = \begin{cases} (2, 1, 0, 0) & \text{with probability 35\%,} \\ (1, 1, 1, 0) & \text{with probability 20\%,} \\ (1, 1, 0, 1) & \text{with probability 20\%, and} \\ (1, 0, 1, 1) & \text{with probability 25\%.} \end{cases}$$

As we show in [Appendix A](#) of the full version, none of these allocations is toxic, and the distribution can be part of an apportionment method in which all three axioms hold for  $\vec{p}$  and all  $h' \leq 3$ . Nevertheless, we show in the following that any apportionment method  $F$  that satisfies house monotonicity and quota and that has the above distribution for  $F(\vec{p}, 3)$  must violate ex ante proportionality. Indeed, fix such an  $F$ . On the one hand, note that, for  $h = 4$ , state 2's standard quota is  $4 \cdot \frac{25}{100} = 1$ , so any quota apportionment must give the state 1 seat. Since any solution  $f$  in the support of  $F$  satisfies house monotonicity and quota by assumption, any  $f$  such that  $f(\vec{p}, 3) = (1, 0, 1, 1)$  must satisfy  $f(\vec{p}, 4) = (1, 1, 1, 1)$ . Thus, with at least 25% probability,  $F_1(\vec{p}, 4) = 1$ . On the other hand, since state 1's standard quota for  $h = 4$  is  $1.8 \leq 2$ ,  $F_1(\vec{p}, 4) \leq 2$  holds deterministically, by quota. It follows that  $\mathbb{E}[F_1(\vec{p}, 4)] \leq 25\% \cdot 1 + 75\% \cdot 2 = 1.75 < 1.8$ , which means that  $F$  must violate ex ante proportionality as claimed. To avoid this kind of conflict between house monotonicity, ex ante proportionality, and quota, the distribution of  $F(\vec{p}, 3)$  must allocate at least 5% combined probability to the allocations  $(2, 0, 1, 0)$  and  $(2, 0, 0, 1)$ , which to us is not obvious other than by considering the specific implications on  $h = 4$  as above.

### 8.4.2 Cumulative Rounding

The examples of the last section showed that it is difficult to construct house monotone apportionment methods seat-by-seat. In this section, we develop an approach that is able to explicitly take into account how rounding decisions constrain each other across house sizes. Our approach will be based on dependent randomized rounding in a bipartite graph that we construct. First, we state the main theorem by Gandhi et al. [GKPS06]:

**Theorem 8.3** (Gandhi et al.) *Let  $(A \cup B, E)$  be an undirected bipartite graph with bipartition  $(A, B)$ . Each edge  $e \in E$  is labeled with a weight  $w_e \in [0, 1]$ . For*

[Sti79] Still (1979): *A Class of New Methods for Congressional Apportionment.*

[BY79] Balinski and Young (1979): *Quota-tone Apportionment Methods.*

[GKPS06] Gandhi et al. (2006): *Dependent Rounding and Its Applications to Approximation Algorithms.*

each  $v \in A \cup B$ , we denote the fractional degree of  $v$  by  $d_v := \sum_{e \in E} w_e$ .

Then there is a random process, running in  $O((|A| + |B|) \cdot |E|)$  time, that defines random variables  $X_e \in \{0, 1\}$  for all  $e \in E$  such that the following properties hold:

**Marginal distribution:** for all  $e \in E$ ,  $\mathbb{E}[X_e] = w_e$ ,

**Degree preservation:** for all  $v \in A \cup B$ ,  $\sum_{e \in E} X_e \in \{\lfloor d_v \rfloor, \lceil d_v \rceil\}$ , and

**Negative correlation:** for all  $v \in A \cup B$  and  $S \subseteq \{e \in E \mid v \in e\}$ ,  $\mathbb{P}[\bigwedge_{e \in S} X_e = 1] \leq \prod_{e \in S} w_e$  and  $\mathbb{P}[\bigwedge_{e \in S} X_e = 0] \leq \prod_{e \in S} (1 - w_e)$ .

If  $X_e = 1$  for an edge  $e$ , we say that  $e$  gets *rounded up*, and if  $X_e = 0$  then  $e$  gets *rounded down*. We do not use the negative correlation property in our apportionment results, but it is crucial in many other applications of dependent rounding: It implies that linear combinations of the shape  $\sum_{e \in S} a_e X_e$  for some  $a_e \in [0, 1]$  obey Chernoff concentration bounds [PS97].

To see the connection to apportionment, let  $\vec{p}$  be a population profile. Then to warm up, the problem of apportioning a single seat can be easily cast as dependent rounding in a bipartite graph: Indeed, let  $A$  consist of a single special node  $a$  and let  $B$  contain a node  $b_i$  for each state  $i$ . We draw an edge  $e = \{a, b_i\}$  with weight  $w_e = p_i / \sum_{j \in N} p_j$  for each state  $i$ . Apply dependent rounding to this star graph. Then  $a$ 's fractional degree of exactly 1 means that, by degree preservation, exactly one edge  $\{a, b_i\}$  gets rounded up, which we interpret as the seat being allocated to state  $i$ . Moreover, marginal distribution ensures that each state receives the seat with probability proportional to its population. This shows that randomized rounding can naturally express ex ante proportionality, which will become a useful building block in the following.

Next, we will expand our construction to multiple house seats, and to satisfying house monotonicity across different house sizes. The most natural way is to duplicate the star-graph structure from the last paragraph, once per house size  $h = 1, 2, \dots$ <sup>10</sup> with nodes  $a^h, \{b_i^h\}_{i \in N}$  and edges  $\{\{a^h, b_i^h\}\}_{i \in N}$ . If  $\{a^h, b_i^h\}$  gets rounded up in the  $h$ -th copy of the star graph, we interpret this as the  $h$ -th seat going to state  $i$ . In other words, we determine how many seats get apportioned to state  $i$  for a house size  $h$  by counting how many edges  $\{a^{h'}, b_i^{h'}\}$  got rounded up across all  $h' \leq h$ . This construction automatically satisfies house monotonicity, and satisfies ex ante proportionality by the marginal distribution property, but it may violate quota by arbitrary amounts.

To explain how randomized rounding might be useful for guaranteeing quota, let us give a few details on how Gandhi et al.'s *pipage rounding* procedure randomly rounds a bipartite graph. In each step, pipage rounding selects either a cycle or a maximal path consisting of edges with *fractional* weights in  $(0, 1)$ . The edges along this cycle or path are then alternately labeled “even” or “odd”.<sup>11</sup> Depending on a biased coinflip and appropriate numbers  $\alpha, \beta > 0$ , the algorithm either (1) increases all odd edge weights by  $\alpha$  and decreases all even edge weights by  $\alpha$ , or (2) decreases all odd edge weights by  $\beta$  and increases all even edge weights by  $\beta$ . In this process, more and more edge weights become zero or one, which determines the  $X_e$  once no fractional edges remain.

The cycle/path rounding steps in pipage rounding represent an opportunity to couple the seat-allocation decisions across  $h$ , in a way that ultimately will allow

[PS97] Panconesi and Srinivasan (1997): *Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds*.

10: In this intuitive exposition, we will not consider any explicit upper bound on the house sizes we consider. Our formal result in [Theorem 8.6](#) will round a finite graph but this will suffice to obtain house monotonicity for all house sizes  $h \in \mathbb{N}_+$ .

11: This is possible because, in a bipartite graph, any cycle has an even number of edges.

us to guarantee quota. In our current graph consisting of disjoint stars, there are no cycles and the maximal paths are always pairs of edges  $\{a^h, b_i^h\}, \{a^h, b_j^h\}$  for two states  $i, j$  and some  $h$ . Thus, pipage rounding correctly anti-correlates the decision of giving the  $h$ -th seat to state  $i$  and the decision of giving the  $h$ -th seat to state  $j$ , but decisions for different seats remain independent. To guarantee quota, increasing (resp., decreasing) the probability of the  $h$ -th seat going to state  $i$  should also decrease (resp., increase) the probability of some nearby seats  $h'$  going to state  $i$  and increase (resp., decrease) the probability of seats  $h'$  going to some other state  $j$ . The difficulty is to choose these  $h'$  and  $j$  to provide quota, which is particular tricky since, in the course of running pipage rounding, some of the edge weights will be rounded to zero and one and no longer be available for paths or cycles.

Not only are we able to use pipage rounding to guarantee quota, but we will do so through a general construction that adds quota-like guarantees to an arbitrary instance of repeated randomized rounding; we refer to this technique as *cumulative rounding*. In the following statement, the “time steps”  $t$  take the place of our possible house sizes  $h$ .

**Theorem 8.4** *Let  $(A \cup B, E)$  be an undirected bipartite graph. For each time step  $t = 1, \dots, T$ , consider a set of edge weights  $\{w_e^t\}_{e \in E}$  in  $[0, 1]$  for this bipartite graph. For each  $v \in A \cup B$  and  $1 \leq t \leq T$ , we denote the fractional degree of  $v$  at time  $t$  by  $d_v^t := \sum_{v \in e \in E} w_e^t$ .*

*Then there is a random process, running in  $O(T^2 \cdot (|A| + |B|) \cdot |E|)$  time, that defines random variables  $X_e^t \in \{0, 1\}$  for all  $e \in E$  and  $1 \leq t \leq T$ , such that the following properties hold for all  $1 \leq t \leq T$ . Let  $D_v^t := \sum_{v \in e \in E} X_e^t$  denote the random degree of  $v$  at time  $t$ .*

**Marginal distribution:** *for all  $e \in E$ ,  $\mathbb{E}[X_e^t] = w_e^t$ ,*

**Degree preservation:** *for all  $v \in A \cup B$ ,  $D_v^t \in \{\lfloor d_v^t \rfloor, \lceil d_v^t \rceil\}$ ,*

**Negative correlation:** *for all  $v \in A \cup B$  and  $S \subseteq \{e \in E \mid v \in e\}$ ,  $\mathbb{P}[\bigwedge_{e \in S} X_e^t = 1] \leq \prod_{e \in S} w_e^t$  and  $\mathbb{P}[\bigwedge_{e \in S} X_e^t = 0] \leq \prod_{e \in S} (1 - w_e^t)$ ,*

**Cumulative degree preservation:** *for all  $v \in A \cup B$ , it holds that  $\sum_{t'=1}^t D_v^{t'} \in \{\lfloor \sum_{t'=1}^t d_v^{t'} \rfloor, \lceil \sum_{t'=1}^t d_v^{t'} \rceil\}$ .*

The first three properties could be achieved by applying [Theorem 8.3](#) in each time step independently. Cumulative rounding correlates these rounding processes such that cumulative degree preservation (a generalization of quota) is additionally satisfied.

### 8.4.3 House Monotone, Quota-Compliant, and Ex Ante Proportional Apportionment

Before we prove [Theorem 8.4](#), we will explain how cumulative rounding can be used to construct an apportionment method that is house monotone and satisfies quota and ex ante proportionality.

None of these three axioms connects the outcomes at different population profiles  $\vec{p}$  and so it suffices to consider them independently. Thus, let us fix a population profile  $\vec{p}$ . Denote the total population by  $p := \sum_{i \in N} p_i$ . The behavior of a house monotone solution on inputs with profile  $\vec{p}$  and arbitrary house sizes can be expressed through what we call an *infinite seat sequence*, an infinite sequence  $\alpha = \alpha_1, \alpha_2, \dots$  over the states  $N$ . We will also define *finite seat sequences*, which are sequences  $\alpha = \alpha_1, \dots, \alpha_p$  of length  $p$  over the states. Either sequence represents that, for any house size  $h$  (in the case of a finite seat sequence  $h \leq p$ ), the sequence apportionments  $a_i(h) := |\{1 \leq h' \leq h \mid \alpha_{h'} = i\}|$  seats to each state  $i$ . We can naturally express the quota axiom for seat sequences:  $\alpha$  satisfies *quota* if, for all  $h$  ( $h \leq p$  if  $\alpha$  is finite) and all states  $i$ , we have  $a_i(h) \in \{\lfloor h p_i/p \rfloor, \lceil h p_i/p \rceil\}$ .

The main obstacle in obtaining a house monotone method via cumulative rounding is that we can only apply cumulative rounding to a finite number  $T$  of copies, whereas the quota axiom must hold for all house sizes  $h \in \mathbb{N}_+$ . However, it turns out that for our purposes of satisfying quota, we can treat the allocation of seats  $1, 2, \dots, p$  independently from the allocation of seats  $p+1, \dots, 2p$ , the allocation of seats  $2p+1, \dots, 3p$ , and so forth. The reason is that, when  $h$  is a multiple  $kp$  of  $p$  (for some  $k \in \mathbb{N}_+$ ), each state  $i$ 's standard quota is an integer  $k p_i$ . Thus, any solution that satisfies quota is forced to choose exactly the allocation  $(k p_1, \dots, k p_n)$  for house size  $h$ . At this point, the constraints for satisfying quota and house monotonicity reset to what they were at  $h = 1$ . We make this precise in the following lemma, proved in [Appendix B](#) of the full version,

**Lemma 8.5** *An infinite seat sequence  $\alpha$  satisfies quota iff it is the concatenation of infinitely many finite seat sequences  $\beta^1, \beta^2, \beta^3, \dots$  of length  $p$  each satisfying quota, i.e.,*

$$\alpha = \beta_1^1, \beta_2^1, \dots, \beta_p^1, \beta_1^2, \beta_2^2, \dots, \beta_p^2, \beta_1^3, \dots.$$

This lemma allows us to apply cumulative rounding to only  $T = p$  many copies of a star graph. Then, cumulative rounding produces a random matching that encodes a finite seat sequence satisfying quota, and [Lemma 8.5](#) shows that the infinite repetition of this finite sequence describes an infinite seat sequence satisfying quota. This implies the existence of an apportionment method satisfying all three axioms we aimed for. The formal proof is in [Appendix B](#) of the full version.

**Theorem 8.6** *There exists an apportionment method  $F$  that satisfies house monotonicity, quota, and ex ante proportionality.*

**Implications for deterministic methods** Our construction also increases our understanding of *deterministic* apportionment solutions satisfying house monotonicity and quota: Indeed, the possible roundings of the bipartite graph constructed for cumulative rounding turn out to correspond one-to-one to the finite seat sequences satisfying quota. Together with [Lemma 8.5](#), this gives a characterization of all seat sequences that satisfy quota, providing a geometric (and graph-theoretic) alternative to the characterizations by Balinski and Young [[BY79](#)] and Still [[Sti79](#)].

[[BY79](#)] Balinski and Young (1979): *Quota-tone Apportionment Methods*.

[[Sti79](#)] Still (1979): *A Class of New Methods for Congressional Apportionment*.

**Theorem 8.7** For each population vector  $\vec{p}$ , we can construct a bipartite graph  $(A \cup B, E)$  such that the set  $S$  of all finite seat sequences satisfying quota for  $\vec{p}$  is in one-to-one correspondence to the corner points of the polytope of all perfect fractional matchings on  $(A \cup B, E)$ . Together with [Lemma 8.5](#) this characterizes the set of infinite seat sequences satisfying quota as the set of infinite sequences over  $S$ .

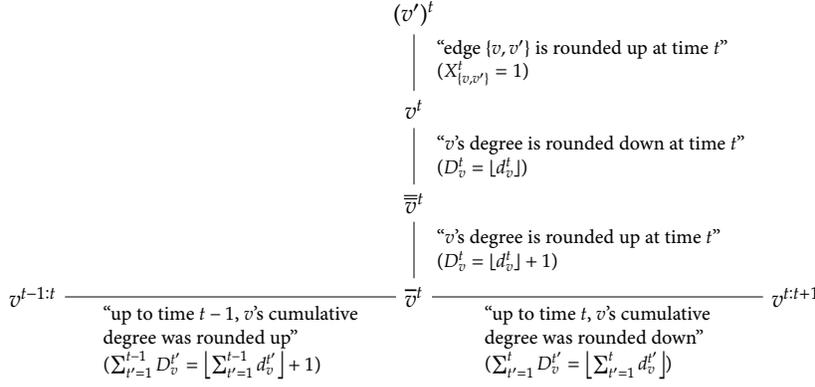
Since a fractional matching assigning each edge  $\{a, b_i\}$  a weight of  $p_i/p > 0$  lies in the interior of this polytope of perfect fractional matchings, one immediate consequence of this characterization (equivalently, of ex ante proportionality in [Theorem 8.6](#)) is that, for each state  $i$  and  $h \in \mathbb{N}_+$ , there is a house monotone and quota-compliant solution that assigns the  $h$ -th seat to  $i$ . To our knowledge, this result is not obvious based on the earlier characterizations. More generally, the polytope characterization might be useful in answering questions such as “For a set of pairs  $(h_1, i_1), (h_2, i_2), \dots, (h_t, i_t)$ , is there a population-monotone and quota-compliant solution that assigns the  $h_j$ -th seat to state  $i_j$  for all  $1 \leq j \leq t$ ?” To answer this question, one can remove the nodes  $a^{h_j}$  and  $b_{i_j}^{h_j}$  from the graph (simulating that they got matched) and check whether the remaining graph still permits a perfect matching, say, with the help of Hall’s marriage theorem [[Hal35](#)].

[[Hal35](#)] Hall (1935): *On Representatives of Subsets.*

**Computation** Before we prove the cumulative rounding result in [Section 8.5](#), let us quickly discuss computational considerations of our house-monotone apportionment method. While it is possible to run dependent rounding on the constructed graph (for a given population profile  $\vec{p}$ ), the running time would scale in  $O(p^2 n^2)$ , and the quadratic dependence on the total population  $p$  might be prohibitive. In practice, we see two ways to avoid this computational cost:

First, one might often not require a solution that is house monotone on all possible house sizes  $h \in \mathbb{N}_+$ ; instead, it might suffice to rule out Alabama paradoxes for house sizes up to an upper bound  $h_{max}$ . In this case, it suffices to apply cumulative rounding on  $h_{max}$  many copies of the graph, leading to a much more manageable running time of  $O(h_{max}^2 n^2)$ .

A second option would be to apply cumulative rounding on all  $p$  copies of the graph, but to stop pipage rounding once all edge weights in the first  $h$  copies of the graph are integral, even if edge weights for higher house sizes are still fractional. This would allow to return an apportionment for inputs  $\vec{p}, h$  while randomly determining not a single house-monotone solution, but a conditioned distribution  $F^c$  over house-monotone solutions, all of which agree on the apportionment for  $\vec{p}$  and  $h$ . Since all solutions are house monotone, the expected number of seats for a party will always monotonically increase in  $h$  across the conditioned distribution. Should it become necessary to determine apportionments for larger house sizes, one can simply continue the cumulative-rounding process where it left off. Since the pipage rounding used to prove [Theorem 8.4](#) leaves open which cycles or maximal paths get rounded next, it seems likely that one can deliberately choose cycles/paths such that the apportionment for the first  $h$  seats is determined in few rounds.



**Figure 8.3:** Interpretation of each edge being rounded up in the constructed graph, for arbitrary nodes  $v, v' \in A \cup B$  and  $1 \leq t \leq T$ . The correctness of this characterization will be shown along the proof of [Theorem 8.4](#), specifically in the sections on degree preservation and cumulative degree preservation.

## 8.5 Proof of Cumulative Rounding

We will now prove [Theorem 8.4](#) on cumulative rounding. Our proof will construct a weighted bipartite graph including  $T$  many copies of  $(A \cup B, E)$ , connected by appropriate additional edges and nodes, and then applying dependent rounding to this constructed graph. The additional edges and vertices ensure that if too many edges adjacent to some node  $v$  are rounded up in one copy of the graph, then this is counterbalanced by rounding down edges adjacent to  $v$  in another copy.

**Construction 8.1** Let  $(A \cup B, E)$ ,  $T$ , and  $\{w_{e,t}^t\}$  be given as in [Theorem 8.4](#). We construct a new weighted, undirected, and bipartite graph as follows: For each node  $v \in A \cup B$  and for each  $t = 1, \dots, T$ , create four nodes  $v^t$ ,  $\bar{v}^t$ ,  $\tilde{v}^t$ , and  $v^{t:t+1}$ ; furthermore, create a node  $v^{0:1}$  for each node  $v$ . For each  $\{a, b\} \in E$  and  $t = 1, \dots, T$ , connect the nodes  $a^t$  and  $b^t$  with an edge of weight  $w_{\{a,b\}}^t$ . Additionally, for each node  $v \in A \cup B$  and each  $t = 1, \dots, T$ , insert edges with the following weights:

$$\begin{array}{c}
 v^t \\
 | \\
 1 - d_v^t + \lfloor d_v^t \rfloor \\
 \bar{v}^t \\
 | \\
 d_v^t - \lfloor d_v^t \rfloor \\
 \tilde{v}^t \\
 | \\
 1 - \sum_{v'=1}^t d_{v'}^t + \lfloor \sum_{v'=1}^t d_{v'}^t \rfloor \\
 v^{t-1:t} \text{ --- } \bar{v}^t \text{ --- } v^{t:t+1}
 \end{array}$$

Before we go into the proof, we give in [Figure 8.3](#) an interpretation for what it means for each edge in the constructed graph to be rounded up. One can easily verify that, under the (premature) assumption that cumulative rounding satisfies marginal distribution, degree preservation, and cumulative degree preservation, the edge weights coincide with the probabilities of each interpretation's event. We want to stress that it is not obvious that these descriptions will indeed be consistent for any dependent rounding of the constructed graph, and we will not make use of these descriptions in the proof of [Theorem 8.4](#). Instead, the characterizations will follow from intermediate results in the proof. We give these interpretations here to make the construction seem less mysterious. We begin the

formal analysis of the construction with a sequence of simple observations about the constructed graph (proofs are in [Appendix C](#) of the full version).

**Lemma 8.8** *The graph produced by [Construction 8.1](#) is bipartite.*

**Lemma 8.9** *All edge weights lie between 0 and 1.*

**Lemma 8.10** *For each node  $v \in A \cup B$ , the following table gives the fractional degrees of various nodes in the constructed graph, all of which are integer:*

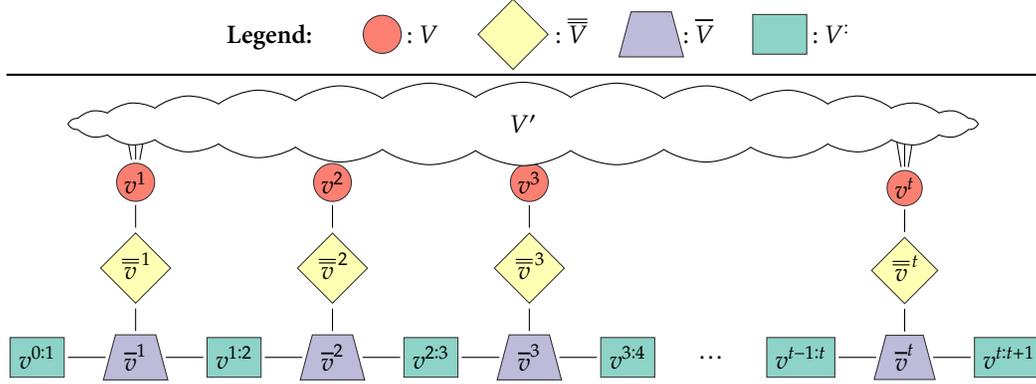
nodes	fractional degree
$v^t$ ( $\forall 1 \leq t \leq T$ )	$\lfloor d_v^t \rfloor + 1$
$\bar{v}^t$ ( $\forall 1 \leq t \leq T$ )	$\lfloor \sum_{t'=1}^t d_v^{t'} \rfloor - \lfloor \sum_{t'=1}^{t-1} d_v^{t'} \rfloor - \lfloor d_v^t \rfloor + 1$
$\bar{\bar{v}}^t$ ( $\forall 1 \leq t \leq T$ )	1
$v^{t:t+1}$ ( $\forall 1 \leq t \leq T-1$ )	1
$v^{0:1}$	0

*Proof of [Theorem 8.4](#).* We define cumulative rounding as the random process that follows [Construction 8.1](#) and then applies dependent rounding ([Theorem 8.3](#)) to the constructed graph, which is valid since the graph is bipartite and all edge weights lie in  $[0, 1]$  ([Lemmas 8.8](#) and [8.9](#)). For an edge  $e$  in the constructed graph, let  $\hat{X}_e$  be the random variable indicating whether dependent rounding rounds it up or down. For any edge  $\{a, b\} \in E$  in the underlying graph and some  $1 \leq t \leq T$ , we define the random variable  $X_{\{a,b\}}^t$  to be equal to  $\hat{X}_{\{a^t, b^t\}}$ . Recall that we defined  $D_v^t = \sum_{v \in e \in E} X_e^t$ .

To prove the theorem, we have to bound the running time of this process, and provide the four guaranteed properties: marginal distribution, degree preservation, negative correlation, and cumulative degree preservation, out of which the last property takes by far the most work.

**Running time.** Without loss of generality, we may assume that each vertex  $v \in A \cup B$  is incident to at least one edge, since, otherwise, we could remove this vertex in a preprocessing step. From this, it follows that  $|E| \in \Omega(|A| + |B|)$ . Constructing the graph takes  $O(T|E|)$  time, which will be dominated by the time required for running dependent rounding on the constructed graph. The constructed graph has  $(1 + 4T)(|A| + |B|) \in O(T(|A| + |B|))$  nodes and  $T|E| + 4T(|A| + |B|) \in O(T|E|)$  edges. Since the running time of dependent rounding scales in the product of the number of vertices and the number of edges, our procedure runs in  $O(T^2(|A| + |B|)|E|)$  time, as claimed.

**Marginal distribution.** For an edge  $\{a, b\} \in E$  and  $1 \leq t \leq T$ ,  $\mathbb{E}[X_{\{a,b\}}^t] = \mathbb{E}[\hat{X}_{\{a^t, b^t\}}] = w_{\{a,b\}}^t$ , where the last equality follows from the marginal-distribution property of dependent rounding.



**Figure 8.4:** Illustration of the counting argument for proving cumulative degree preservation. Edges in the figure are edges from the constructed graph, a superset of the edges in the rounded graph. Node color and shape indicate the set that a node belongs to, as indicated in the legend.

**Degree preservation.** Fix a node  $v \in A \cup B$  and  $1 \leq t \leq T$ . By [Lemma 8.10](#), the fractional degree of  $v^t$  is  $\lfloor d_v^t \rfloor + 1$ , and thus, by degree preservation of dependent rounding, exactly  $\lfloor d_v^t \rfloor + 1$  edges adjacent to  $v^t$  must be rounded up. The only of these edges that does not count into  $D_v^t$  is  $\{\bar{v}^t, v^t\}$ ; depending on whether this edge is rounded up or down,  $D_v^t$  is either  $\lfloor d_v^t \rfloor$  or  $\lfloor d_v^t \rfloor + 1$ . If  $d_v^t$  is not integer, the latter number equals  $\lceil d_v^t \rceil$ , which proves degree preservation. Else, if  $d_v^t$  is an integer, the edge weight of  $\{\bar{v}^t, v^t\}$  is 1. Dependent rounding always rounds up edges with weight 1, which means that  $D_v^t$  is definitely  $\lfloor d_v^t \rfloor$  in this case. Thus, degree preservation holds in either case.

**Negative correlation.** Negative correlation for  $v \in A \cup B$ ,  $S \subseteq \{e \in E \mid v \in e\}$ , and  $1 \leq t \leq T$  directly follows from the negative-correlation property of dependent rounding for the node  $v^t$  and the edge set  $S' := \{(v^t, (v')^{t'}) \mid \{v, v'\} \in S\}$ .

**Cumulative degree preservation.** Fix a node  $v \in A \cup B$  and  $1 \leq t \leq T$ . We will consider the “rounded version” of the constructed graph, i.e., the unweighted bipartite graph over the nodes of the constructed graph in which exactly those edges are present that got rounded up by the randomized rounding process. We define five sets of nodes in the rounded graph ([Figure 8.4](#)):

$$\begin{aligned} V &:= \{v^{t'} \mid 1 \leq t' \leq t\} & V' &:= \{(v')^{t'} \mid v' \in (A \cup B) \setminus \{v\}, 1 \leq t' \leq t\} \\ \bar{V} &:= \{\bar{v}^{t'} \mid 1 \leq t' \leq t\} & \bar{V}' &:= \{\bar{v}'^{t'} \mid 1 \leq t' \leq t\} \\ V^* &:= \{v^{t:t+1} \mid 0 \leq t' \leq t\} \end{aligned}$$

For any set of nodes  $V_1$  in the rounded graph, we denote its neighborhood by  $N(V_1)$ , and we will write  $\text{deg}(V_1)$  for the sum of degrees of  $V_1$  in the rounded graph. For any two sets of nodes  $V_1, V_2$ , we write  $\text{cut}(V_1, V_2)$  to denote the number of edges between  $V_1$  and  $V_2$  in the rounded graph.

Note that  $\sum_{t'=1}^t D_v^{t'}$ , which we must bound, equals  $\text{cut}(V, V')$ . We will bound this quantity by repeatedly using the following fact, which we refer to *pivoting*: For pairwise disjoint sets of nodes  $V_0, V_1, V_2$ , if  $N(V_0) \subseteq V_1 \cup V_2$ , then  $\text{deg}(V_0) = \text{cut}(V_0, V_1) + \text{cut}(V_0, V_2)$ . Since [Lemma 8.10](#) gives us a clear view of the fractional degrees of nodes in the constructed graph, and since, by degree preservation, a

node's degree in the rounded graph must equal the fractional degree whenever the latter is an integer, this property allows us to express cuts in terms of other cuts. Figure 8.4 illustrates which of these sets border on each other, and helps in following along with the derivation.

$$\begin{aligned}
& \sum_{v'=1}^t D_v^{t'} \\
&= \text{cut}(V, V') \\
&= \text{deg}(V) - \text{cut}(V, \bar{V}) && (\text{pivot } V_0 = V, V_1 = V', V_2 = \bar{V}) \\
&= t + \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor - \text{cut}(V, \bar{V}) && (\text{deg}(V) = t + \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor \text{ by Lemma 8.10}) \\
&= t + \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor - \text{deg}(\bar{V}) + \text{cut}(\bar{V}, \bar{V}) && (\text{pivot } V_0 = \bar{V}, V_1 = V, V_2 = \bar{V}) \\
&= \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor + \text{cut}(\bar{V}, \bar{V}) && (\text{deg}(\bar{V}) = t \text{ by Lemma 8.10}) \\
&= \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor + \text{deg}(\bar{V}) - \text{cut}(\bar{V}, V') && (\text{pivot } V_0 = \bar{V}, V_1 = \bar{V}, V_2 = V') \\
&= \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor - \text{cut}(\bar{V}, V') + \sum_{v'=1}^t (\lfloor \sum_{v''=1}^{t'} d_v^{t''} \rfloor - \lfloor \sum_{v''=1}^{t'-1} d_v^{t''} \rfloor - \lfloor d_v^{t'} \rfloor + 1) \\
& && (\text{Lemma 8.10}) \\
&= \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor - \text{cut}(\bar{V}, V') + \lfloor \sum_{v''=1}^t d_v^{t''} \rfloor - \sum_{v'=1}^t \lfloor d_v^{t'} \rfloor + t && (\text{telescoping sum}) \\
&= \lfloor \sum_{v'=1}^t d_v^{t'} \rfloor + t - \text{cut}(\bar{V}, V').
\end{aligned}$$

To bound  $\text{cut}(\bar{V}, V')$  in the last expression, observe that  $N(V' \setminus \{v^{t:t+1}\}) \subseteq \bar{V}$ , from which it follows that  $\text{cut}(\bar{V}, V' \setminus \{v^{t:t+1}\}) = \text{deg}(V' \setminus \{v^{t:t+1}\}) = t - 1$ . Thus,  $\text{cut}(\bar{V}, V') = t - 1 + \mathbb{1}\{\hat{X}_{\{\bar{v}^t, v^{t:t+1}\}}\}$ , and we resume the above equality

$$= \lfloor \sum_{v'=1}^t d_v^{t'} \rfloor + t - (t - 1 + \mathbb{1}\{\hat{X}_{\{\bar{v}^t, v^{t:t+1}\}}\}) = \lfloor \sum_{v'=1}^t d_v^{t'} \rfloor + 1 - \mathbb{1}\{\hat{X}_{\{\bar{v}^t, v^{t:t+1}\}}\}.$$

If  $\sum_{v'=1}^t d_v^{t'}$  is not an integer, the above shows that  $\sum_{v'=1}^t D_v^{t'}$  is either the floor or ceiling of  $\sum_{v'=1}^t d_v^{t'}$ , establishing cumulative degree preservation. Else, if  $\sum_{v'=1}^t d_v^{t'}$  is integer, note that the weight of the edge  $\{\bar{v}^t, v^{t:t+1}\}$  in the constructed graph is 1. Since dependent rounding always rounds such edges up,  $\sum_{v'=1}^t D_v^{t'} = \lfloor \sum_{v'=1}^t d_v^{t'} \rfloor$ . This establishes cumulative degree preservation, the last of the properties guaranteed by the theorem.  $\square$

## 8.6 Other Applications of Cumulative Rounding

Our exploration of house monotone randomized apportionment led us to the more general technique of cumulative rounding, which we believe to be of broader interest. We next illustrate this by discussing additional applications.

### 8.6.1 Sortition of the European Commission

The European Commission is one of the main institutions of the European Union, in which it plays a role comparable to that of a government. The commission consists of one commissioner from each of the 27 member states, and each commissioner is charged with a specific area of responsibility. Since the number of EU member states has nearly doubled in the past 20 years, so has the size of the commission. Besides making coordination inside the commission less efficient,

the enlargement of the commission has led to the creation of areas of responsibility much less important than others. Since the important portfolios are typically reserved for the largest member states, smaller states have found themselves with limited influence on central topics being decided in the commission.

To remedy this imbalance, Buchstein and Hein [BH09] propose to reduce the number of commissioners to 15, meaning that only a subset of the 27 member states would send a commissioner at any given time. Which states would receive a seat would be determined every 5 years by a weighted lottery (“sortition”), in which states would be chosen with degressive proportional weights. Degressive means that smaller states get non-proportionately high weight; such weights are already used for apportioning the European parliament. The authors argue that by the law of large numbers, political representation on the commission would be essentially proportional to these weights in politically relevant time spans.

However, a follow-up simulation study by Buchstein et al. [BHJ13] challenges this assertion on two fronts: (1) First, the authors find that their implementation of a weighted lottery chooses states with probabilities that deviate from proportionality to the weights in a way that is not analytically tractable (see [BH83, p. 24]). (2) Second, and more gravely, their simulations undermine “a central argument in favor of legitimacy” in the original proposal, namely, that “in the long term, the seats on the commission would be distributed approximately like the share of lots” [BHJ13, own translation]. From a mathematical point of view, the authors had overestimated the rate of concentration across the independent lotteries. Instead, in the simulation, it takes 30 lotteries (150 years) until there is a probability of 99% that all member states have sent at least one commissioner.

These serious concerns could be resolved by using cumulative rounding to implement the weighted lotteries. Specifically, we would again construct a star graph with a special node  $a$  and one node  $b_i$  for each state  $i$ , setting  $T$  to the desired number of consecutive lotteries. For each  $1 \leq t \leq T$ , each edge  $\{a, b_i\}$  would be weighted by  $\frac{15 w_i}{\sum_{j \in N} w_j}$  where  $w_j$  is state  $j$ 's degressive weight.<sup>12</sup> Degree preservation on  $a$  would ensure that in each lottery  $t$ , exactly 15 distinct states are selected. By marginal distribution, the selection probabilities would be exactly proportional to the degressive weights, resolving issue (1). Furthermore, cumulative degree preservation on the state nodes would eliminate issue (2). If we take the effective selection probabilities of Buchstein et al. [BHJ13] as the states' weights, even the smallest states  $i$  would have an edge weight  $w_{\{a, b_i\}}^t \approx 0.187$ . Then, cumulative quota prevents any state from getting rounded down in  $11 = \lceil 2/0.187 \rceil$  consecutive lotteries: Indeed, fixing any  $0 \leq t_0 \leq T - 11$ ,

$$\sum_{t'=1}^{t_0+11} D_{b_i}^{t'} \geq \lfloor (t_0 + 11) 0.187 \rfloor \geq \lfloor t_0 0.187 \rfloor + 2 \geq \lceil t_0 0.187 \rceil + 1 \geq \sum_{t'=1}^{t_0} D_{b_i}^{t'} + 1,$$

which means that state  $i$  must have been selected at least once between time  $t_0 + 1$  and  $t_0 + 11$ . In political terms, this means that 55, not 150, years would be enough to *deterministically* ensure that each member state send a commissioner at least once in this period.

[BH09] Buchstein and Hein (2009): *Randomizing Europe*.

[BHJ13] Buchstein et al. (2013): *Die „EU-Kommissionslotterie“*.

[BH83] Brewer and Hanif (1983): *An Introduction to Sampling with Unequal Probabilities*.

12: This assumes that each state's weight is at most  $1/15 \sum_{j \in N} w_j$ , which is in fact not the case for the largest member states [BHJ13]. Therefore, proportionality to the weights is incompatible with Buchstein and Hein's requirement that each state may not send more than a single commissioner. To obtain proportionality, there are three solutions: increasing the number of commissioners, allowing a state to receive multiple commissioners (which can be expressed in cumulative rounding by splitting the state into multiple copies), or adjusting the weights. If desired, cumulative rounding can accommodate weights that change across lotteries according to population projections, which Buchstein et al. do for some of their experiments.

### 8.6.2 Repeated Allocation of Courses to Faculty Or Shifts to Workers

A common paradigm in fair division is to first create a fractional assignment between agents and resources, and to then implement this fractional assignment in expectation, through randomized rounding. Below, we describe a setting of allocating courses to faculty members in a university department, in which implementing a fractional assignment using cumulative rounding is attractive.

For a university department, denote its set of faculty members by  $A$  and the set of possible courses to be taught by  $B$ . For each faculty member  $a$  and course  $b$ , let there be a weight  $w_{\{a,b\}} \in [0, 1]$  indicating how frequently course  $b$  should be taught by  $a$  on average. These numbers could be derived using a process such as probabilistic serial [BM01], the Hylland-Zeckhauser mechanism [HZ79], or the mechanisms by Budish et al. [BCKM13], which would transform preferences of the faculty over which courses to teach into such weights.<sup>13</sup> We will allow arbitrary fractional degrees on the faculty side (so one person can teach multiple courses) while assuming that the fractional degree of any course  $b$  is at most 1.

When applying cumulative rounding to this graph (using the same edge weights in each period) for consecutive semesters  $1 \leq t \leq T$ , we observe the following properties.

- ▶ Marginal distribution implies that, in each semester, faculty member  $a$  has a probability  $w_{\{a,b\}}$  of teaching course  $b$ .
- ▶ Degree preservation on the course side means that a course is never taught by two different faculty members in the same semester.
- ▶ Degree preservation on the faculty side implies that a faculty member  $a$ 's teaching load does not vary by more than 1 between semesters; it is either the floor or the ceiling of  $a$ 's expected teaching load.
- ▶ Cumulative degree preservation on the course side ensures that courses are offered with some regularity. For example, if a course's fractional degree is  $1/2$ , it will be taught exactly once in every academic year (either in Fall or in Spring).
- ▶ Cumulative degree preservation on the faculty side allows for non-integer teaching load. For example, a faculty member with fractional degree 1.5 will have a "2-1" teaching load, i.e., they will teach 3 courses per year, either 2 in the Fall and one in the Spring or vice versa.

The same approach is applicable for matching workers to shifts.

One could also use cumulative rounding to repeatedly round a fractional assignment of general chores, such as the ones computed by the online platform spliddit.org [GP14]. In this case, a caveat is that (cumulative) degree preservation only ensures that the *number* of assigned chores is close to its expected number per time period, not necessarily the *cost* of the assigned chores. However, if many chores are allocated per time step, and if costs are additive, then an agent's per-timestep cost is well-concentrated, which follows from the negative-correlation property that permits the application of Chernoff concentration bounds [PS97].

[BM01] Bogomolnaia and Moulin (2001): *A New Solution to the Random Assignment Problem*.

[HZ79] Hylland and Zeckhauser (1979): *The Efficient Allocation of Individuals to Positions*.

[BCKM13] Budish et al. (2013): *Designing Random Allocation Mechanisms*.

13: Although these mechanisms are formulated for goods, they can be applied to bads when the number of bads allocated to each agent is fixed, as it is when allocating courses to faculty or shifts to workers.

[GP14] Goldman and Procaccia (2014): *Spliddit*.

[PS97] Panconesi and Srinivasan (1997): *Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds*.

## 8.7 Discussion

Though our work is motivated by the application of apportioning seats at random, the technical questions we posed and addressed are fundamental to the theoretical study of apportionment. In a sense, any deterministic apportionment solution is “unproportional” — after all, its role is to decide which agents receive more or fewer seats than their standard quota. By searching for randomized methods satisfying *ex ante* proportionality, we ask whether these unproportional solutions can be combined (through random choice) such that these deviations from proportionality cancel out to achieve perfect proportionality, and whether this remains possible when we restrict the solutions to those satisfying subsets of the axioms population monotonicity, house monotonicity, and quota. Naturally, this objective pushes us to better understand the whole space of solutions satisfying these subsets of axioms, including the space’s more extreme elements. Therefore, it is in hindsight not surprising that our work led to new insights for deterministic apportionment: a more robust impossibility between population monotonicity and quota (Theorem 8.1), an exploration of solutions generalizing the divisor solutions (Theorem 8.2), and a geometric characterization of house monotone and quota compliant solutions (Theorem 8.7).

Concerning the cumulative rounding technique introduced in this paper, we have only scratched the surface in exploring its applications. In particular, we hope to investigate whether cumulative rounding can extend existing algorithmic results that use dependent rounding, and whether it can be used to construct new approximation algorithms. For both of these purposes, the negative-correlation property, which we have not used much so far, will hopefully turn out to be valuable.

Despite their advantageous properties, randomized mechanisms have in the past often met with resistance by practitioners and the public [KPS18], but, as we described in the first part of this thesis, we see signs of a shift in attitudes in the area of *citizens’ assemblies*. These citizens’ assemblies proudly point to their random selection — and to complex selection algorithms such as the one we developed in Chapter 2 — as a source of legitimacy. If citizens’ assemblies continue to become more prominent, randomness will be associated by the public with neutrality and fairness, not with haphazardness, and randomized apportionment methods might receive serious consideration.

[KPS18] Kurokawa et al. (2018): *Leximin Allocations in the Real World*.

# PROPOSAL FOR REFUGEE RESETTLEMENT

# Online Refugee Placement

# 9

## 9.1 Introduction

There are 27 million refugees around the world [UNH22]. The United Nations High Commissioner for Refugees (UNHCR) considers over 1.4 million of them to be in need of *resettlement*, that is, permanent relocation from a temporary country of asylum to the country of resettlement [UNH20]. Resettlement is mainly targeted at the most vulnerable refugees, such as children at risk, survivors of violence and torture, and those with urgent medical needs. Dozens of countries around the world resettle refugees; in 2019, for example, around 108 000 refugees were resettled [UNHnd]. Still, the number of refugees in need of resettlement far exceeds the number that is actually resettled in every year.

Historically, most countries taking in resettled refugees have paid little attention to where inside the country these refugees are placed. This policy might be worth reconsidering, however, since there is ample evidence that the initial local resettlement destination dramatically affects the outcomes of refugees [ÆFG11; ÅF09; ÅÖZ10; ÅR07; BFH+18; Dam14; MHH19]. One specific variable impacted by community placement is whether and when resettled refugees find employment. Employment plays a key role in the successful integration of a refugee by “promoting economic independence, planning for the future, meeting members of the host society, providing opportunity to develop language skills, restoring self-esteem and encouraging self-reliance” [ASo8].

Since promoting employment is so crucial, the American resettlement agency HIAS began in 2017 to match refugees to communities using the matching software Annie™ MOORE (Matching and Outcome Optimization for Refugee Empowerment), which is designed to maximize the total number of refugees who obtain employment soon after arrival [AAM+21]. Each week, the US government assigns a new batch of refugees to HIAS, and Annie™ suggests which community each refugee in the batch should be placed in. Before this work, Annie™ made its suggestions using a greedy algorithmic approach, that is, each batch of arrivals was allocated by separately maximizing the expected employment of this batch (subject to the remaining community capacities and ensuring that refugees have access to necessary services). Allocating affiliate capacity in such a greedy way will likely lead to suboptimal employment, however: A placement algorithm could achieve better employment by weighing in each placement decision whether a slot of capacity is more beneficial when used by a refugee in the current batch or when saved up for some refugee potentially arriving later in the year.

In this chapter, we improve the optimization approach of Annie™ by intentionally incorporating the dynamic nature of the matching problem. For this, we design two closely related algorithms — one based on stochastic programming and another based on Walrasian equilibrium — that optimize the dynamic matching of refugees to communities in the United States. Our focus is to study these algorithms in a rich model that captures all of the relevant practical features of the

9.1 Introduction . . . . .	133
9.1.1 Related Work . . . . .	134
9.1.2 Chapter Outline . . . . .	136
9.2 Institutional Background . . . . .	136
9.3 Model . . . . .	137
9.4 Online Bipartite Matching . . . . .	138
9.4.1 Algorithmic Approach . . . . .	138
9.4.2 Empirical Evaluation . . . . .	141
9.5 Non-Unit Cases . . . . .	144
9.6 Batching . . . . .	145
9.6.1 Empirical Evaluation . . . . .	146
9.7 Unknown Arrival Numbers . . . . .	147
9.7.1 Relying on Capacities . . . . .	147
9.7.2 Misestimation & Uncertainty . . . . .	150
9.7.3 Better Arrival Knowledge . . . . .	152
9.8 Implementation . . . . .	153
9.9 Discussion . . . . .	155

[UNH22] UNHCR (2022): *Global Trends: Forced Displacement in 2021*.

[UNH20] UNHCR (2020): *Global Resettlement Needs 2021*.

[UNHnd] UNHCR (n.d.): *Refugee Data Finder*.

[ÆFG11] Åslund et al. (2011): *Peers, Neighborhoods, and Immigrant Student Achievement*.

[ÅF09] Åslund and Fredriksson (2009): *Peer Effects in Welfare Dependence Quasi-Experimental Evidence*.

[ÅÖZ10] Åslund et al. (2010): *How Important Is Access to Jobs?*

[ÅR07] Åslund and Rooth (2007): *Do When and Where Matter?*

[BFH+18] Bansak et al. (2018): *Improving Refugee Integration through Data-Driven Algorithmic Assignment*.

[Dam14] Damm (2014): *Neighborhood Quality and Labor Market Outcomes*.

[MHH19] Martén et al. (2019): *Ethnic Networks Can Foster the Economic Integration of Refugees*.

[ASo8] Ager and Strang (2008): *Understanding Integration*.

[AAM+21] Ahani et al. (2021): *Placement Optimization in Refugee Resettlement*.

refugee resettlement process, including indivisible families of refugees, batching, and unknown numbers of refugee arrivals. We evaluate the performance of our algorithms on HIAS data from 2014 until 2019. We show that both algorithms achieve over 98 percent of the hindsight-optimal employment in all years whereas the greedy baseline achieves only around 90 percent. We then describe how we implemented our algorithms within Annie™ to create Annie™ 2.0.

### 9.1.1 Related Work

This chapter extends a line of work initiated by Bansak et al. [BFH+18], which aims to increase refugees' employment outcomes through data- and optimization-driven placement. This approach consists of two components: using machine learning to estimate the probability that a given refugee placed at a given community would find employment, and using mathematical programming to perform the optimization. Ahani et al. [AAM+21] adopted a similar approach to develop Annie™; they also pointed out the practical relevance of indivisible families and the possibility of batching. Both papers seek to maximize employment with respect to a current batch of refugees, without considering future arrivals; it is in this sense that we think of deployed algorithms as *greedy*, and that is indeed our benchmark in this chapter.

Our dynamic refugee placement problem generalizes the classic *edge-weighted online bipartite matching* problem, but most algorithms in the theoretical literature are not promising for our application since they are optimized for overly pessimistic arrival scenarios. Whereas competitive analysis was quite successful for unweighted online bipartite matching [KVV90], no constant-factor approximation algorithm is possible for the weighted setting if arrivals are adversarial [FHTZ20]. In the random-order arrival model, a  $1/e$ -approximation is possible [KRTV13], but the algorithm is impractical; in particular, it leaves the first 37% of arrivals entirely unmatched. Even if arrivals are drawn i.i.d. from a known distribution, Manshadi et al. [MGS12] show that no online algorithm can obtain a better approximation ratio than 0.823, far below the performance of even the greedy baseline in our setting. Since this impossibility is based on contrived arrival distributions, many papers additionally assume that arrivals belong to finitely many types determining their edge weights. In this setting, constructing matchings that are optimal up to lower-order terms (with high probability) is not difficult [AHL13], and multiple papers obtain such results, often in generalizations of edge-weighted online bipartite matching [AHL12; AHL13; VB21]. What limits the applicability of these algorithms to our setting, however, is that these algorithms require the distribution over types explicitly in their input, and are often constructed based on the assumption that multiple arrivals of each type will occur in a single run of the algorithm. By contrast, we estimate employment scores based on 20 independent features, which means the number of refugee “types” is too large to enumerate and we do not expect to see identical refugees.

Our algorithmic approach can be seen as an instantiation of the *Bayes Selector*, an algorithmic paradigm that has yielded impressive theoretical and empirical results across various problems with stochastic online arrivals [BGV20; FB19; SFCS22; VB21; VBG21]. Conceptually, the Bayes Selector takes in a prediction of

[BFH+18] Bansak et al. (2018): *Improving Refugee Integration through Data-Driven Algorithmic Assignment*.

[AAM+21] Ahani et al. (2021): *Placement Optimization in Refugee Resettlement*.

[KVV90] Karp et al. (1990): *An Optimal Algorithm for On-Line Bipartite Matching*.

[FHTZ20] Fahrback et al. (2020): *Edge-Weighted Online Bipartite Matching*.

[KRTV13] Kesselheim et al. (2013): *An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions*.

[MGS12] Manshadi et al. (2012): *Online Stochastic Matching*.

[AHL13] Alaei et al. (2013): *The Online Stochastic Generalized Assignment Problem*.

[AHL12] Alaei et al. (2012): *Online Prophet-Inequality Matching with Applications to Ad Allocation*.

[VB21] Vera and Banerjee (2021): *The Bayesian Prophet*.

[BGV20] Banerjee et al. (2020): *Constant Regret in Online Allocation*.

[FB19] Freund and Banerjee (2019): *Good Prophets Know When the End Is Near*.

[SFCS22] Sinclair et al. (2022): *Hindsight Learning in MDPs with Exogenous Inputs*.

[VBG21] Vera et al. (2021): *Online Allocation and Pricing*.

future arrivals and then performs the action (in our setting: chooses the affiliate for the current arrival) that seems most likely to coincide with the action taken by an optimal benchmark. Under some regularity conditions on the arrivals, algorithms following this methodology have *constant regret*, that is, the expected difference between the algorithm's performance and that of the optimal benchmark does not grow with the size of the problem. The prediction of future arrivals often takes other shapes, but can be a sampled trajectory of arrivals as in our work [BGV20]. In most papers, the choice of action is based on how *often* the optimal benchmark would take an action in the simulated future rather than, as in our work, on the marginal effect of an action on the optimal value. Very recently, however, Sinclair et al. [SFCSS22] analyze the same variant of the Bayes Selector (the “hindsight planning policy”) as our Equation (9.1), and show that it gives constant regret for the problem of stochastic online bin packing. Even though we do not provide theoretical guarantees in this chapter, the success of the Bayes Selector across related settings partially explains our good empirical performance.

Shadow prices have been used to guide decisions in online settings in a variety of contexts, including advertising [DH09; MSVV07; VVS10], revenue management [TVV04], worker assignment [HV12; JKK21], and resource allocation [AWZ20]. Agrawal et al. [AWY14] develop a dynamic learning approach where prices are calculated in a similar manner to ours; while they update their match scores upon every doubling of the arrival history, we update our match scores upon every batch. Ho and Vaughan [HV12] extend the advertising context of Devanur and Hayes [DH09] to assign workers to tasks when match scores are initially unknown and must be learned. Like Ho and Vaughan [HV12], Johari et al. [JKK21] also consider the worker-to-job context, but learn scores while matching via an explore-then-exploit approach. In our setting, our scores are known in advance independent of arrivals [AAM+21].

In independent and concurrent work, Bansak [Ban20] also considers dynamic refugee resettlement; the algorithm obtaining the highest employment in that study is equivalent to our two-stage stochastic programming formulation in the simplest setting. Our model is much richer as we include non-unit family sizes, incompatibilities between families and communities, and allow for uncertain arrival numbers. A second limitation of their work is that their best algorithm is prohibitively slow. This lack of computational efficiency pushes them to consider algorithms with worse employment outcomes, and it limits their empirical evaluation to a single month of arrivals. By contrast, we leverage multiple algorithmic insights to speed up the algorithm by two orders of magnitude without substantially trading off employment. This speed-up allows us to empirically evaluate our algorithms for realistic matching periods, which last for an entire fiscal year. We compare the running times of both algorithms in Section 9.6. Finally, very recent work by Bansak and Paulson [BP22] extends the earlier work by Bansak [Ban20] by incorporating a secondary objective that seeks to consume capacity at similar rates across affiliates, improving case wait times across affiliates without sacrificing much employment.

[BGV20] Banerjee et al. (2020): *Constant Regret in Online Allocation*.

[SFCSS22] Sinclair et al. (2022): *Hindsight Learning in MDPs with Exogenous Inputs*.

[DH09] Devanur and Hayes (2009): *The Adwords Problem*.

[MSVV07] Mehta et al. (2007): *Adwords and Generalized Online Matching*.

[VVS10] Vee et al. (2010): *Optimal Online Assignment with Forecasts*.

[TVV04] Talluri et al. (2004): *The Theory and Practice of Revenue Management*.

[HV12] Ho and Vaughan (2012): *Online Task Assignment in Crowdsourcing Markets*.

[JKK21] Johari et al. (2021): *Matching While Learning*.

[AWZ20] Asadpour et al. (2020): *Online Resource Allocation with Limited Flexibility*.

[AWY14] Agrawal et al. (2014): *A Dynamic Near-Optimal Algorithm for Online Linear Programming*.

[AAM+21] Ahani et al. (2021): *Placement Optimization in Refugee Resettlement*.

[Ban20] Bansak (2020): *A Minimum-Risk Dynamic Assignment Mechanism Along with an Approximation, Heuristics, and Extension from Single to Batch Assignments*.

[BP22] Bansak and Paulson (2022): *Outcome-Driven Dynamic Refugee Assignment with Allocation Balancing*.

### 9.1.2 Chapter Outline

In Section 9.2, we provide an overview of the US refugee resettlement process. In Section 9.3, we outline our model of dynamic refugee matching. In Section 9.4, we propose our two algorithms and show that they obtain near-optimal employment in a baseline setting that ignores the indivisibility of families, batching, and uncertainty about the total number of arrivals. In the next three sections, we layer on complexity toward the setting encountered in practice: indivisible families (Section 9.5), batching (Section 9.6), and unknown arrival numbers (Section 9.7). In these sections, we demonstrate that indivisible families and batching do not substantially change our algorithms' employment performance, and that employment remains high unless the number of arrivals widely deviates from the numbers announced by the government. In Section 9.8, we then explain how we implemented our approach within Annie™ and conclude in Section 9.9.

## 9.2 Institutional Background

The federal Office of Refugee Resettlement was created by the Refugees Act in 1980. The Act established funding rules and authorized the President of the United States to set annual capacities for resettlement. The resettlement process is managed by the US Refugee Admissions Program (USRAP) of the US Department of State, in conjunction with a number of federal agencies across federal departments as well as the International Organization for Migration and the UNHCR.

Applications for the resettlement program take place from outside of the US, typically in refugee camps. The US government conducts security checks, medical screening, and performs cultural orientation, which can take upwards of 18 months [Jon15]. After clearance, USRAP decentralizes the process of welcoming refugees to nine NGOs known as *resettlement agencies*, of which one is HIAS. Each agency works with their own network of local *affiliates*, each supported by local offices as well as religious entities like churches, synagogues, or mosques, which serve as community liaisons for refugees. Each agency typically works with dozens of affiliates, though the number of affiliates can fluctuate over time. Some affiliates lack services to host certain kinds of refugees. For example, certain affiliates do not have translators for non-English-speaking refugees or lack support for single-parent families.

Agencies have no influence on which refugees are cleared for resettlement by USRAP or on when the refugees might arrive. Resettlement agencies meet on a weekly or fortnightly basis in order to allocate among themselves the refugees that have been cleared by USRAP. Refugees are usually resettled with members of their family. Such an indivisible group of refugees is referred to as a *case*. As a family can split when its members are fleeing their home country, some refugees who are applying for resettlement might already have existing relatives or connections in the US. Such cases *with US ties* are automatically resettled near their existing ties. All other refugees, referred to as *free cases*, can be resettled by any agency into any of the agency's affiliates.

[Jon15] Jones (2015): *Home Away From Home*.

Each affiliate has an assigned annual capacity for the number of refugees (rather than cases) it can admit in a given fiscal year.<sup>1</sup> These capacities are approved by USRAP and, in theory, agencies cannot exceed them. In practice, capacities can be slightly adjusted towards the end of the year or, as in recent years, substantially revised in the course of the year. Since capacities limit the number of refugees *arriving* in a fiscal year rather than *allocated* in it, and since there is typically a delay of multiple months between the two events, the Department of State tells the resettlement agencies an estimated arrival date for each cleared case. Agencies are assessed annually by USRAP on their performance in finding employment for refugees within 90 days of their arrival. Data on 90-day employment is therefore diligently collected by the affiliates and monitored by the agencies.

### 9.3 Model

An instance of the matching problem first defines a set  $L$  of *affiliates*, and each affiliate  $\ell$  has a capacity  $c_\ell \in \mathbb{N}_{\geq 0} \cup \{\infty\}$  of how many refugees it can host. We call a collection  $\{c_\ell\}_{\ell \in L}$  of capacities for all affiliates a *capacity profile*  $c$ . To describe changes in capacity, it will be useful to manipulate the capacity profiles as vectors. Specifically, we write  $c - e_\ell$  to describe the capacity profile obtained from  $c$  by reducing the capacity of affiliate  $\ell$  by 1.

On the other side of the matching problem is a set  $N = \{1, \dots, n\}$  of *cases*. Each case  $i$  represents an indivisible family of  $s_i \in \mathbb{N}_{\geq 1}$  refugees. Furthermore, each case  $i$ , for each affiliate  $\ell$ , has an *employment score*  $u_{i,\ell}$ , which indicates the expected number of case members that will find employment if the case is allocated to  $\ell$ . Typically, these employment scores  $u_{i,\ell}$  are real numbers in  $[0, s_i]$ , but we will also allow to set  $u_{i,\ell} = -\infty$  to express that case  $i$  is not compatible with affiliate  $\ell$ . We will refer to the combination of a case's size and vector of employment scores as the *characteristics* of the case. To ensure that the matching problem is always feasible, we will assume that  $L$  contains a special affiliate  $\perp$  that represents leaving a case unmatched, where  $u_{i,\perp} = 0$  for all cases  $i$  and  $c_\perp = \infty$ .<sup>2</sup>

We use the employment scores developed by Ahani et al. [AAM+21], and we give details on data preprocessing and training in Appendix A of the full version. Throughout this chapter, we consider these employment scores as ground truth, which means that we evaluate algorithms directly based on the employment scores. An evaluation of how accurately the employment scores predict employment outcomes is outside of the scope of this chapter, and has already been performed by Ahani et al.

The goal of the matching problem is to allocate cases to affiliates such that the *total employment*, that is, the sum of employment scores, is maximized, subject to not exceeding capacities. For a set  $I \subseteq N$  and a capacity profile  $c = \{c_\ell\}_{\ell \in L}$ , define  $\text{MATCHING}(I, c)$  as the matching integer linear program (ILP) below, where variables  $x_{i,\ell}$  indicate whether case  $i \in I$  is matched to affiliate  $\ell \in L$ :

$$\begin{aligned} & \text{maximize} && \sum_{i \in I} \sum_{\ell \in L} u_{i,\ell} x_{i,\ell} \\ & \text{subject to} && \sum_{\ell \in L} x_{i,\ell} = 1 && \forall i \in I \end{aligned}$$

1: Each fiscal year ranges from October 1 of the previous calendar year to September 30. For example, fiscal year 2017 ranges from October 1, 2016 to September 30, 2017.

2: For example, allowing cases to be unmatched is necessary since an arriving case might only be compatible with affiliates whose capacity is already exhausted. When these situations occur in practice, such cases do not remain unmatched; instead, capacities can be increased or case-affiliate incompatibilities overruled manually by the arrivals officer. For our sequence of models, we report the fraction of matched refugees in Appendix D.7 of the full version, and find that our algorithms do not lead to fewer refugees being matched than in the greedy baseline. To lower the number of unmatched refugees at the cost of reducing employment, one can add a constant reward per refugee to the  $u_{i,\ell}$  with  $\ell \neq \perp$ .

[AAM+21] Ahani et al. (2021): *Placement Optimization in Refugee Resettlement*.

$$\begin{aligned} \sum_{i \in I} s_i x_{i,\ell} &\leq c_\ell && \forall \ell \in L \\ x_{i,\ell} &\in \{0, 1\} && \forall i \in I, \ell \in L. \end{aligned}$$

Let  $\text{OPT}(I, c)$  denote the optimal objective value of  $\text{MATCHING}(I, c)$ . The *linear programming (LP) relaxation* of  $\text{MATCHING}(I, c)$  is obtained by replacing the constraint  $x_{i,\ell} \in \{0, 1\}$  by  $0 \leq x_{i,\ell} \leq 1$  for all  $i \in I, \ell \in L$ . For a fixed matching, we define the *match score* of a case  $i$  as its employment score  $u_{i,\ell_i}$  at the affiliate  $\ell_i$  where it is allocated; we will also refer to its *match score per refugee*,  $u_{i,\ell_i}/s_i$ .

Finally, cases arrive *online*, that is, they arrive one by one and, when case  $i$  arrives, the decision of which affiliate to place it in must be made irrevocably, before the characteristics of the subsequent arrivals  $i + 1, \dots, n$  are known.<sup>3</sup> Thus, although an online matching algorithm must still produce a matching whose indicator variables  $x_{i,\ell}$  satisfy the constraints of  $\text{MATCHING}(N, c)$ , the total employment  $\sum_{i \in N, \ell \in L} u_{i,\ell} x_{i,\ell}$  typically will not attain the benchmark  $\text{OPT}(N, c)$  of the *optimal matching in hindsight*. While we will not commit to a specific model of how the characteristics of arriving cases are generated, these arrivals should be thought of as *stochastic* rather than worst-case, and the distribution of case characteristics as changing slowly enough that sampling from recent arrivals is a reasonable proxy for the distribution of future arrivals.

Throughout the following sections, we will consider a sequence of models, which incorporate an increasing number of features of the real-world refugee allocation problem: in [Section 9.4](#), we consider traditional online bipartite matching, which results from requiring  $s_i = 1$  in the above model; from [Section 9.5](#) onward, we allow cases to have arbitrary size; from [Section 9.6](#) onward, we also allow cases to arrive in batches rather than one by one; in [Section 9.7](#), we no longer assume that the total number  $n$  of arriving cases is known to the algorithm.

3: From [Section 9.6](#) onward, cases will instead arrive in batches, which can be allocated simultaneously.

## 9.4 Online Bipartite Matching ( $s_i = 1$ )

In this section, we will consider the special case of online bipartite (weighted) matching. We stress that this classic problem does not capture key features of the refugee-allocation problem in practice, which we will add in later sections. Instead, online bipartite matching allows us to more cleanly draw connections to theoretical arguments, which help motivate our algorithm design. Later in the chapter, we will empirically show that the approach continues to work well in richer and more realistic settings.

Formally, this section considers the model defined in the previous section, with the restriction that all cases consist of single refugees, that is, that  $s_i = 1$  for all  $i \in N$ . Under this assumption, it is well-known that the optimum matching for the ILP  $\text{MATCHING}(I, c)$  can be found by solving its LP relaxation.

### 9.4.1 Algorithmic Approach

To motivate our algorithmic approach, we begin by describing why matching systems currently deployed in practice lead to suboptimal employment. These systems assign cases *greedily*, which — putting aside batching for now — means

that an arriving case  $i$  is matched to the affiliate  $\ell$  with highest employment score  $u_{i,\ell}$  among those that have at least  $s_i$  remaining capacity. The main problem with greedy assignment is that it exhausts the capacity of the most desirable affiliates too early. In particular, we observe on the real data that a large fraction of cases have their highest employment score in the same affiliate  $\ell^*$ , but that the size of the employment advantage of affiliate  $\ell^*$  over the second-best affiliate varies. Since it only considers the highest-employment affiliate for each case, greedy assignment will fill the entire capacity of  $\ell^*$  early in the year, including with some cases that benefit little from this assignment. Consequently, cases that would particularly profit from being placed in  $\ell^*$  but arrive later in the year no longer fit within the capacity.

Intuitively, the decision to match a case  $i$  to an affiliate  $\ell$  has two effects: the immediate increase of the total employment by  $u_{i,\ell}$  but also an opportunity cost for consuming  $\ell$ 's capacity, which might prevent profitable assignments for later arrivals. Since greedy assignment only considers the former effect, it leaves employment on the table.

A better approach is *two-stage stochastic programming*, which allocates an arriving case  $i$  to the affiliate  $\ell$  maximizing the sum of the immediate employment  $u_{i,\ell}$  and the expected optimal employment obtainable by matching the future arrivals subject to the remaining capacity. That is, if, at the time of  $i$ 's arrival, the remaining capacities are given by  $c$ , two-stage stochastic programming allocates  $i$  to the affiliate

$$\operatorname{argmax}_{\ell \in L: c_\ell \geq s_i} u_{i,\ell} + \mathbb{E} \left[ \operatorname{OPT}(\{i+1, \dots, n\}, c - s_i \cdot e_\ell) \right],$$

where the expectation is taken over the characteristics of cases  $j = i+1, \dots, n$ . Since adding a constant term does not change the  $\operatorname{argmax}$ , this can be rewritten as

$$\begin{aligned} &= \operatorname{argmax}_{\ell \in L: c_\ell \geq s_i} u_{i,\ell} - \mathbb{E} \left[ \operatorname{OPT}(\{i+1, \dots, n\}, c) \right] + \mathbb{E} \left[ \operatorname{OPT}(\{i+1, \dots, n\}, c - s_i \cdot e_\ell) \right] \\ &= \operatorname{argmax}_{\ell \in L: c_\ell \geq s_i} u_{i,\ell} - \mathbb{E} \left[ \operatorname{OPT}(\{i+1, \dots, n\}, c) - \operatorname{OPT}(\{i+1, \dots, n\}, c - s_i \cdot e_\ell) \right]. \end{aligned} \quad (9.1)$$

Using our assumption that  $s_i = 1$ , this can be simplified to

$$= \operatorname{argmax}_{\ell \in L: c_\ell \geq 1} u_{i,\ell} - \mathbb{E} \left[ \operatorname{OPT}(\{i+1, \dots, n\}, c) - \operatorname{OPT}(\{i+1, \dots, n\}, c - e_\ell) \right].$$

Note that the expectation that is subtracted in either of the last two lines is exactly the expected opportunity cost of reducing the capacity of  $\ell$  by placing case  $i$  there. This motivates our algorithmic approach: in every time step, we first compute a *potential*  $p_\ell$  for each affiliate  $\ell$ . Then, rather than myopically maximizing the utility of the match as does greedy assignment, our algorithm PM (“potential match”) myopically maximizes the utility of the current match minus the potential of the capacity used, as shown in [Algorithm 2](#). (Note that an affiliate  $\ell$  can always be defined in [Line 5](#) as, by assumption,  $c_\perp = \infty$ .)

We estimate the expected value of the opportunity cost by averaging over a fixed number  $k$  of *trajectories*, each of which consists of randomly sampled characteristics of all arrivals  $i+1$  through  $n$ . As the characteristics of arriving

**Algorithm 2:** PM(Potential)**Parameter:** a subroutine Potential to determine affiliate potentials

---

```

1 initialize the capacities  $c_\ell$  for each affiliate  $\ell$ ;
2 for  $t = 1, \dots, n$  do
3   observe the case size  $s_t$  and the employment scores  $\{u_{t,\ell}\}_\ell$ ;
4   call Potential() to define a potential  $p_\ell$  for each affiliate  $\ell$ ;
5    $\ell \leftarrow \operatorname{argmax}_{\ell \in L: c_\ell \geq s_t} u_{t,\ell} - s_t p_\ell$ ;
6   allocate case  $t$  to  $\ell$  and set  $c_\ell \leftarrow c_\ell - s_t$ ;

```

---

refugees change over time, and as these changes tend to be gradual, we draw these arrival characteristics uniformly with replacement from the arrivals in the six months prior to the current allocation decision. In [Appendix D.3](#) of the full version, we evaluate different lengths of this sampling window.

For each sampled trajectory, it remains to calculate the potential, which we would like to equal the opportunity cost  $\operatorname{OPT}(\{i+1, \dots, n\}, c) - \operatorname{OPT}(\{i+1, \dots, n\}, c - e_\ell)$ . Clearly, this could be computed by solving  $O(|L|)$  matching LPs, which is what the flagship algorithm by Bansak [[Ban20](#)] does.

Instead, we make use of a celebrated result in matching theory [[Leo83](#)] to compute the opportunity costs for *all* affiliates with remaining capacity as the shadow prices of a *single* LP:

**Fact 9.1** Fix a matching-problem instance, in which all cases  $i$  have size  $s_i = 1$ . In the LP relaxation of  $\operatorname{MATCHING}(N, c)$ , let  $\{p_\ell\}_{\ell \in L}$  denote the unique element-wise maximal set of shadow prices for the constraints  $\sum_{i \in N} s_i x_{i,\ell} \leq c_\ell$ . Then, for each  $\ell$  with  $c_\ell \geq 1$ ,

$$p_\ell = \operatorname{OPT}(\{i+1, \dots, n\}, c) - \operatorname{OPT}(\{i+1, \dots, n\}, c - e_\ell).$$

This suggests the procedure Pot1 for computing potentials, which is shown in [Algorithm 3](#).

We also develop a second method Pot2 for computing potentials, which is based on a slightly different LP and has different theoretical underpinnings:

- ▶ whereas the matching LP for Pot1 does not include the current batch of arrivals, the current batch is included in the LP for Pot2,
- ▶ whereas Pot1 uses the element-wise maximal set of shadow prices, Pot2 uses the element-wise minimal one, and
- ▶ whereas Pot1 is theoretically derived from two-stage stochastic programming, Pot2 is motivated by a connection to Walrasian equilibria.

For conciseness, we defer the formal definition of Pot2 and its connection to the Walrasian equilibrium to [Appendix B](#) of the full version.

[[Ban20](#)] Bansak (2020): *A Minimum-Risk Dynamic Assignment Mechanism Along with an Approximation, Heuristics, and Extension from Single to Batch Assignments*.

[[Leo83](#)] Leonard (1983): *Elicitation of Honest Preferences for the Assignment of Individuals to Positions*.

### 9.4.2 Empirical Evaluation

We evaluate the employment of our potential-based matching algorithm on real yearly arrivals at HIAS. For each fiscal year, we consider all refugees who arrived in this period, and we consider them in the order in which they were received for allocation by HIAS. For the capacities, we use the year's *final*, i.e. most revised, capacities.<sup>4</sup> We also immediately take into account that affiliates are restricted in which nationalities, languages, and family sizes they can accommodate, as well as in whether they can host single parents and the constraints on tied cases.

---

#### Algorithm 3: Pot1( $k$ )

---

**Parameter:**  $k \in \mathbb{N}_{\geq 1}$ , the number of trajectories per potential computation

**Input:** remaining capacities  $c$ , the index  $t$  of the last observed case, characteristics of cases arriving in the past 6 months

**Output:** a set of potentials  $p_\ell$  for all affiliates  $\ell$

```

1 for  $j = 1, \dots, k$  do
2   for each  $i = t + 1, \dots, n$ , set  $s_i$  and  $\{u_{i,\ell}\}_\ell$  to the size and employment
   scores of a random, recently arrived case;
3   solve the following bipartite-matching LP:

```

$$\begin{aligned}
 & \text{maximize} && \sum_{i=t+1}^n \sum_{\ell \in L} u_{i,\ell} x_{i,\ell} \\
 & \text{subject to} && \sum_{\ell \in L} x_{i,\ell} = 1 && \forall i = (t+1), \dots, n \\
 & && \sum_{i=t+1}^n s_i x_{i,\ell} \leq c_\ell && \forall \ell \in L \quad (*) \\
 & && 0 \leq x_{i,\ell} && \forall i = (t+1), \dots, n, \forall \ell \in L.
 \end{aligned}$$

```

4   for each  $\ell$ , set  $p_\ell^j$  to be the maximal shadow price5 of the constraint (*);
5   set  $p_\ell \leftarrow (\sum_{j=1}^k p_\ell^j)/k$  for all  $\ell$ ;
6   return  $\{p_\ell\}_{\ell \in L}$ ;

```

---

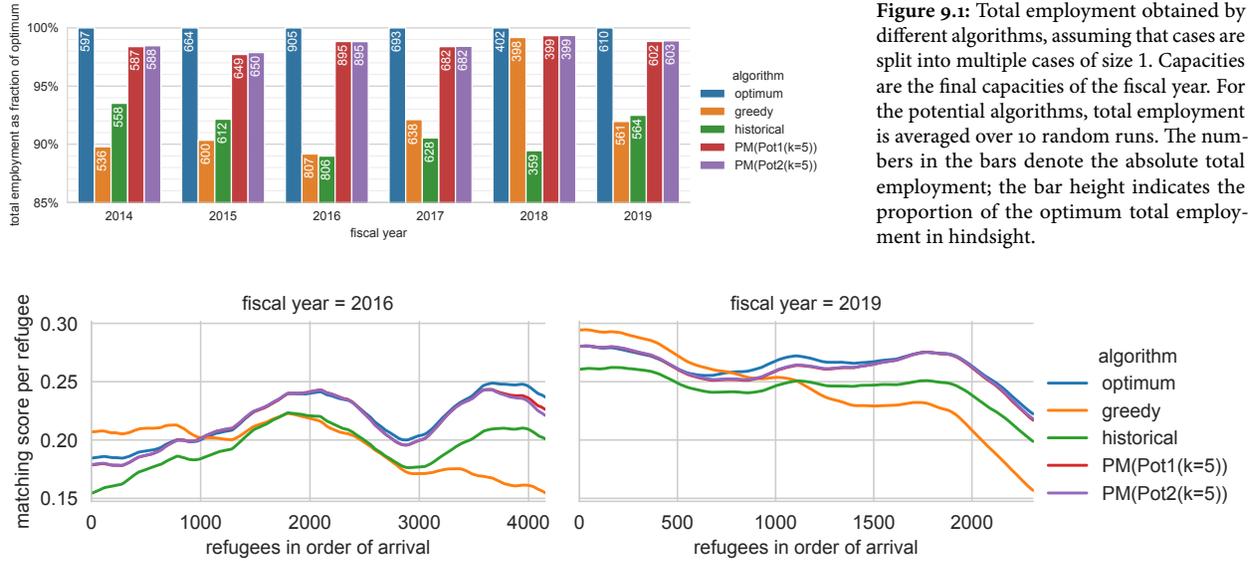
The main way in which this experiment deviates from reality is the assumption (made throughout this section) that cases have unit size. To satisfy this assumption, we split each case of size  $s_i > 1$  into  $s_i$  identical single-refugee cases with a  $1/s_i$  fraction of the original employment scores. In subsequent sections, we will repeat the experiments without this modification.

We study 6 fiscal years, from 2014 to 2019. As affiliates closed and opened across these years, the number of affiliates varies between 16 and 24 (not counting the unmatched affiliate  $\perp$ ). Finally, the number of arriving refugees (respectively, cases) varies between 1 670 (resp., 640) and 4 150 (resp., 1 630) across fiscal years.

As shown in [Figure 9.1](#), even the greedy baseline obtains a total employment of between 89% and 92% of  $\text{OPT}(N, c)$ , the optimum matching in hindsight. (One outlier is the year 2018, which we discuss below.) Nevertheless, the greedy algorithm leads to between 50 and 100 fewer refugees finding employment every year compared to what would have been possible in the optimum matching. Our potential algorithms close a large fraction of this gap, obtaining between 98% and 99% of the optimal total employment, both for algorithms based on

4: When the number of refugees resettled in the fiscal year exceeds the official capacity, we use the number of resettled refugees instead. In these situations, HIAS negotiated an increase in capacity that is not always recorded in our data.

5: One way of finding the maximal shadow price is to first solve the dual LP to find its objective value, then adding a constraint that constrains the objective of the dual LP to be equal to this optimal objective value, and to finally maximize the sum of dual variables  $p_\ell$  over this new restricted LP.



**Figure 9.1:** Total employment obtained by different algorithms, assuming that cases are split into multiple cases of size 1. Capacities are the final capacities of the fiscal year. For the potential algorithms, total employment is averaged over 10 random runs. The numbers in the bars denote the absolute total employment; the bar height indicates the proportion of the optimum total employment in hindsight.

**Figure 9.2:** Evolution of the per-refugee match score in order of arrival, for fiscal years 2016 and 2019 in the experiment of Figure 9.1 (split cases, final capacities). Consecutive match scores are smoothed using triangle smoothing with width 500.

Pot1 and for those based on Pot2. Since experiments in this model take much longer to run than those in subsequent models, we defer a comparison between the two potential methods and between values of  $k$  to Section 9.6.1, where we can run the potential algorithms a sufficient number of times to discern smaller differences.

The fiscal year 2018 stands out from the others due to the fact that the greedy algorithm performs on par with the potential algorithms, at 99% of the hindsight-optimal total employment. This is easily explained by the fact that the capacities are much looser than in other fiscal years: whereas, in all other fiscal years between 2014 and 2019, the number of arriving refugees amounts to between 84% (2019) and 97% (2016) of the final total capacity across all affiliates, this fraction is only 48% in 2018. Since capacity is so abundant, the optimal matching will match a large fraction of cases to their maximum-score affiliate, and the greedy matching is close to optimal.

We also compare to the employment obtained by the allocation chosen by HIAS (“historical”). This comparison gives the historical matching a slight advantage, as HIAS sometimes overrides the incompatibility between an affiliate and a case, which we do not allow any other algorithm to do.<sup>6</sup>

In Figure 9.2, we investigate how the match score changes over the course of two fiscal years, 2016 and 2019, chosen to contain one year in which the greedy and historical baselines perform relatively poorly (2016) and one in which they perform well (2019). As the match score of subsequently arriving refugees can greatly differ, these graphs are heavily smoothed over time. If arrivals were drawn from a time-invariant distribution, we would expect the curves for the optimum matching in hindsight to be level, since how much employment the optimum matching can extract from a case would be independent of the case’s arrival time. Instead, we see that the employment prospects of arrivals fluctuate noticeably over time; in particular, the early refugees in fiscal year 2016 and the late refugees

<sup>6</sup>: In these cases, we estimate the employment achieved by the case using the regression rather than using  $u_{i,\ell} = -\infty$ .

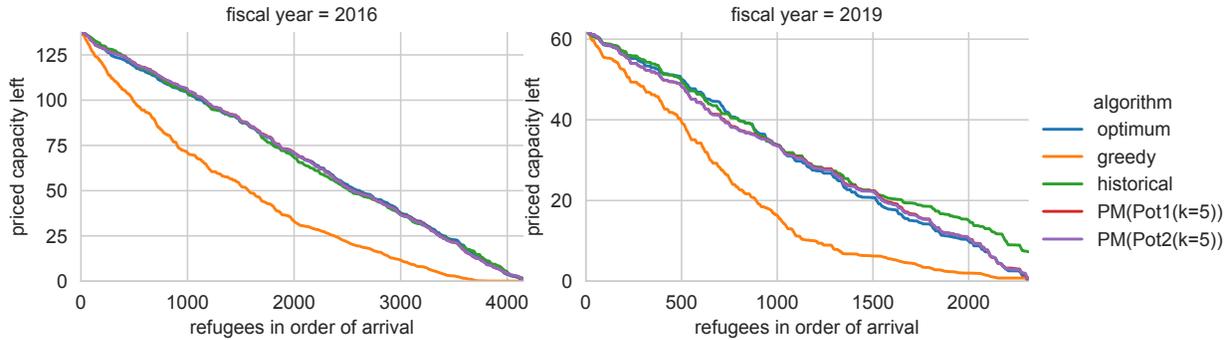


Figure 9.3: Remaining priced capacity at the time of arrival of different refugees, for fiscal years 2016 and 2019 in the experiment of Figure 9.1 (split cases, final capacities).

in fiscal year 2019 seem to have worse employment prospects than other refugees in the plot.

The curves for both potential algorithms are nearly indistinguishable from one another, which shows that the algorithms make very similar decisions. In 2016, these curves start out closely tracking the curve of the optimal-hindsight matching, but fall behind for the last cases, which we observe in most fiscal years. The similarity of the curves over most of the year indicates that our approach of sampling trajectories from past arrivals is nearly as useful as the optimum algorithm's perfect knowledge of future arrivals and that it leads to a similar trade-off in extracting immediate employment versus preserving capacity for later arrivals. Of course, the imperfect knowledge of the future incurs a small loss towards the end of the fiscal year, likely because the amount of capacity reserved per affiliate does not perfectly match the demand, which explains the gap in total employment between the hindsight optimum and the potential algorithms. This typical end-of-year effect is not very pronounced in fiscal year 2019, likely because the final arrivals of fiscal year 2019 have lower employment probabilities than what would be expected based on past arrivals. Instead, the potential algorithms fall behind the optimum algorithm for some period in the middle of the year, perhaps because they are reserving capacity for late arrivals which the optimum already knows to hold little promise.

The most striking curve is that of the greedy algorithm, which lies above those of all other algorithms in the first quarter of arrivals, but then falls clearly below the other curves in the second half. This observation can be explained by the effect we predicted in the motivation of our potential approach: the greedy algorithm extracts small additional gains in employment early in the arrival period, at the cost of prematurely consuming the capacity of the most desirable affiliates. Then, the lack of capacity limits the match scores of later arrivals, resulting in an overall unfavorable trade-off. This effect can be directly seen in Figure 9.3, in which we visualize the amount of capacity remaining in the most valuable affiliates. Specifically, looking at all arrivals of the fiscal year, we compute the shadow prices of the matching LP. At any point in time, we can then weight the remaining capacity by these prices to obtain a *priced capacity*. In Figure 9.3, we see that the optimum-hindsight matching and the potential algorithms use up the priced capacity at a roughly constant pace and essentially consume it all. By contrast, the greedy algorithm uses up the capacity very quickly, such that at the

median refugee, only 22% (2016) or 17% (2019) of the priced capacity is left.

The historical matching made by HIAS does not have such obvious defects, but still falls short in terms of total employment. In both reference years, the average employment moves in parallel with the optimum matching, meaning that HIAS does not overly focus on extracting employment at certain parts of the fiscal year at the expense of others. However, the average employment consistently lies below that of the optimum and of the potential algorithms. We see in [Figure 9.3](#) that, in 2019, HIAS started consuming the priced capacity at a near-constant pace very similar to that of the optimum algorithm. Around the median arrival, however, the historical matching slowed down its capacity consumption and ended up not consuming all priced capacity, which explains some loss in total employment. One reason for this behavior might be that HIAS staff treat the last 9% of the capacity as a reserve that they are more reluctant to use. In a year such as 2019, in which the overall arrivals were only 84% of the total capacity, this heuristic might have actually kept much of the reserve capacity free, including in the affiliates that could have generated higher employment. By contrast, the total arrivals in 2016 amounted to 97% of the overall capacity, which could explain why nearly all priced capacity was consumed in this year. Despite using up priced capacity in a similar pattern as the optimum matching in 2016, the historical assignment achieved lower matching scores throughout the year. This indicates that the low employment of the historical matching is not just due to a reluctance to use the entire capacity, but that the priced capacity is furthermore inefficiently allocated.

## 9.5 Non-Unit Cases ( $s_i \geq 1$ )

The most pressing aspect of refugee matching that we have ignored thus far is that many cases do not consist of individual refugees. Instead, they consist of an entire family of refugees, which has to be resettled to the same affiliate.

To accommodate cases consisting of multiple family members, we will from now drop the assumption that the  $s_i$  are 1. The main effect of this change is that the LP relaxation of the ILPs  $\text{MATCHING}(I, c)$  can now be a strict relaxation. Indeed, the LP relaxation might allow for higher objective values because it allows fractional solutions.<sup>7</sup> As a result, our dual prices will no longer *exactly* compute the marginal value of a unit of capacity. In any case, to retain the exact connection to stochastic programming in [Equation \(9.1\)](#), PM would have to subtract the opportunity cost of  $s_i$  units of capacity from  $u_{i,\ell}$ , which might exceed  $s_i$  times the opportunity cost of a single unit of capacity.

However, as the capacity of most affiliates is much larger than the size of a typical case, both approximations can be expected to be relatively close, which is what we find empirically: we repeat the experiment of the previous section, but without splitting up cases into individual refugees. The results are nearly indistinguishable, which supports our decision to use LP relaxations even in the setting with indivisible cases. The full figures are deferred to [Appendix D.1](#) of the full version.

7: One can always find a fractional solution that splits cases into  $1/s_i$  fractions similarly to what we did in the evaluation of [Section 9.4.2](#).

## 9.6 Batching

A second aspect that we have not considered thus far is that HIAS does not actually process arriving cases one by one, but in batches containing one or multiple cases. Most of these batches result from the weekly meetings between the resettlement agencies, but smaller batches with urgent cases are allocated between the weekly meetings.

The fact that cases arrive in batches does not make the problem harder; after all, a matching algorithm that does not support batching can still be used by presenting the cases of each batch to the algorithm one by one. As we will argue, however, batching represents an opportunity to improve on this strategy: there is a (limited) opportunity to increase total employment and a (substantial) opportunity to reduce running time.

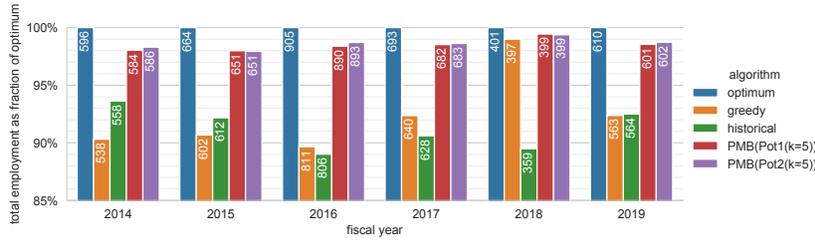
Concerning total employment, using a non-batching algorithm in a batching setting is wasteful since it ignores potentially valuable information. Specifically, when the earliest cases of the batch are allocated, a non-batching algorithm presumes that the characteristics of the other cases in the batch are not yet known. Arguably, as the sizes of batches tend to be much smaller than the total number of cases  $n$ , the amount by which accounting for this information can increase total employment is likely to be limited.

As for running time, given that the matching algorithm receives no new information between the first and last case of a batch, it seems reasonable not to recompute potentials within a batch. As there tend to be 5 to 10 times more cases than batches and as the computation of potentials is the bottleneck in the running time of the potential algorithms, this promises to substantially speed up the algorithm.

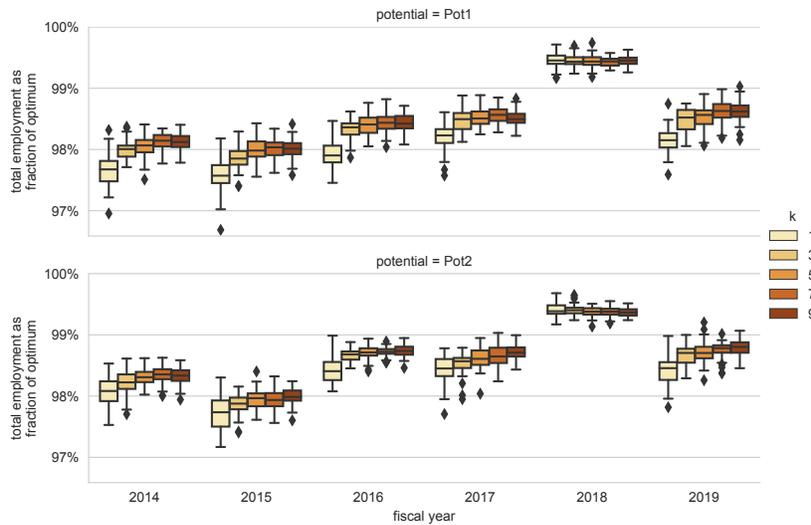
In adapting our algorithm PM to batching, we will not change how we compute the potentials  $p_\ell$ . However, the algorithm now allocates all cases in the batch at once, still with the objective of optimizing the immediate utility of the assignment less the sum of potentials consumed. Thus, our extended algorithm PMB (“potential match with batching”, Algorithm 4 in Appendix C of the full version) allocates the current batch according to the solution to a matching ILP, in which the utility of matching case  $i$  to affiliate  $\ell$  is set to  $u_{i,\ell} - s_i p_\ell$ . Note that, if all batches have size  $b = 1$ , this algorithm coincides with our previous algorithm PM. Moreover, PMB also generalizes the greedy algorithm previously implemented in Annie™, which can be recovered by setting all potentials  $p_\ell$  to zero.

We can now compare the running time of our algorithms to the flagship algorithm by Bansak [Ban20], which is very closely related, but does not use dual prices to compute opportunity costs and handles batching in a way that does not improve running time. The computational bottleneck in both of our algorithms and theirs is the computation of bipartite-matching LPs over the trajectories of simulated future arrivals. Whereas we compute a single such program per batch of arrivals, Bansak solves  $|L| \cdot b$  many such LPs per batch, where  $|L|$  is the number of affiliates and  $b$  is the number of cases in the batch. In our dataset, a typical value of  $|L| \cdot b$  is around 150, so these speed-ups are substantial.

[Ban20] Bansak (2020): *A Minimum-Risk Dynamic Assignment Mechanism Along with an Approximation, Heuristics, and Extension from Single to Batch Assignments.*



**Figure 9.4:** Total employment, where cases are not split and arrive in batches. Capacities are the final fiscal year capacities. In contrast to Figure 9.1, cases are treated as indivisible, cases arrive in batches, and the batching variants of greedy and the potential algorithms are used. For the potential algorithms, the mean employment across 50 random runs is shown.



**Figure 9.5:** Distribution of the total employment obtained by instantiating PMB with different potential methods and different  $k$ , in the experiment of Figure 9.4 (whole cases, batching, final capacities) and over 50 random runs per algorithm.

## 9.6.1 Empirical Evaluation

We repeat the experiment measuring the total employment obtained by the algorithms, this time with the greedy algorithm and the potential algorithms allocating cases in batches. As shown in Figure 9.4, the results again look very close to those in the restricted setting of online bipartite matching, confirming that our algorithmic approach generalizes well not only to non-unit case sizes but also to batching as it is used in practice.

Since processing entire cases in batches is much faster than processing cases (or individual refugees) one by one, we are now in a position to run each potential algorithm many times and analyze the distribution of total employments. As shown in Figure 9.5, the total employment produced by each potential algorithm is sharply concentrated, especially when the algorithms use  $k \geq 3$  trajectories to compute duals.

Running each algorithm many times enables us to compare the relative performance of the potential algorithms. Across both ways of computing potentials, and all fiscal years (with the exception of 2018, where everything is very close together), we see a clear tendency that averaging the potentials across more trajectories improves the employment outcome. These effects are somewhat limited, though, as going from a single trajectory to nine trajectories improves the median employment by less than half a percent of the hindsight optimum. As is to be expected, increasing  $k$  exhibits diminishing returns.

For  $k$  held constant, we observe that the Pot2 variants quite consistently outperform the Pot1 variants; again with the exception of 2018, in which a small inversion of this trend can be seen. While all potential algorithms perform very well, based on these results, we recommend the Pot2 potentials with a relatively large  $k$  for practical implementation. Of course, increasing  $k$  increases the running time of the matching algorithm. However, since a resettlement agency computes only one set of potentials per day, the algorithm runs in few seconds even for  $k = 9$  (see [Appendix D.2](#) of the full version).

To additionally support our observation that the potential algorithms outperform the greedy algorithm and the historical matching, we repeat the experiment from [Figure 9.4](#) for additional arrival sequences derived from the historical data. As we show in [Appendix D.5](#) of the full version, we obtain similar employment performance as in [Figure 9.4](#) if the arrival sequence for each year is reversed, or if we consider shifted yearly arrival periods from, say, April to the March of the following year rather than fiscal years (from October to September). In [Section 9.7.2](#), we also evaluate the algorithms on bootstrapped arrivals. While we discuss more specific observations there, the potential algorithms perform similarly well or slightly better in that setting, consistently at 99% of the hindsight optimum.

## 9.7 Uncertainty in the Number of Future Arrivals

Given that our algorithm PMB supports non-unit sized cases and batching, it might seem that we are ready to replace the greedy algorithm in Annie™ by our potential algorithm. However, our algorithm crucially relies on one piece of input that the greedy algorithm did not need, namely, the total number of cases arriving in the fiscal year. This number determines the length of the sampled trajectories, which can greatly impact the shadow prices and, thus, how the algorithm allocates cases.

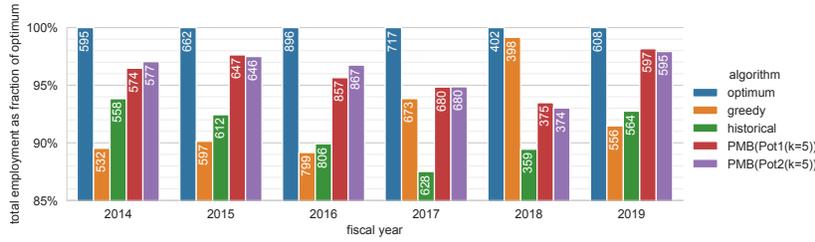
In principle, the information given to resettlement agencies should provide a fairly precise estimate of how many cases are expected to arrive. Indeed, before the start of each fiscal year, the US Department of State announces how many refugees it intends to resettle in that fiscal year, and resettlement agencies are instructed to prepare for a certain fraction of this total number. In fact, HIAS sets its affiliate capacities to sum up to 110% of this number of announced refugees, which is intended to give local affiliates a good idea of how many refugees they will receive while affording the resettlement agency some freedom in its allocation decisions.

### 9.7.1 Relying on Capacities

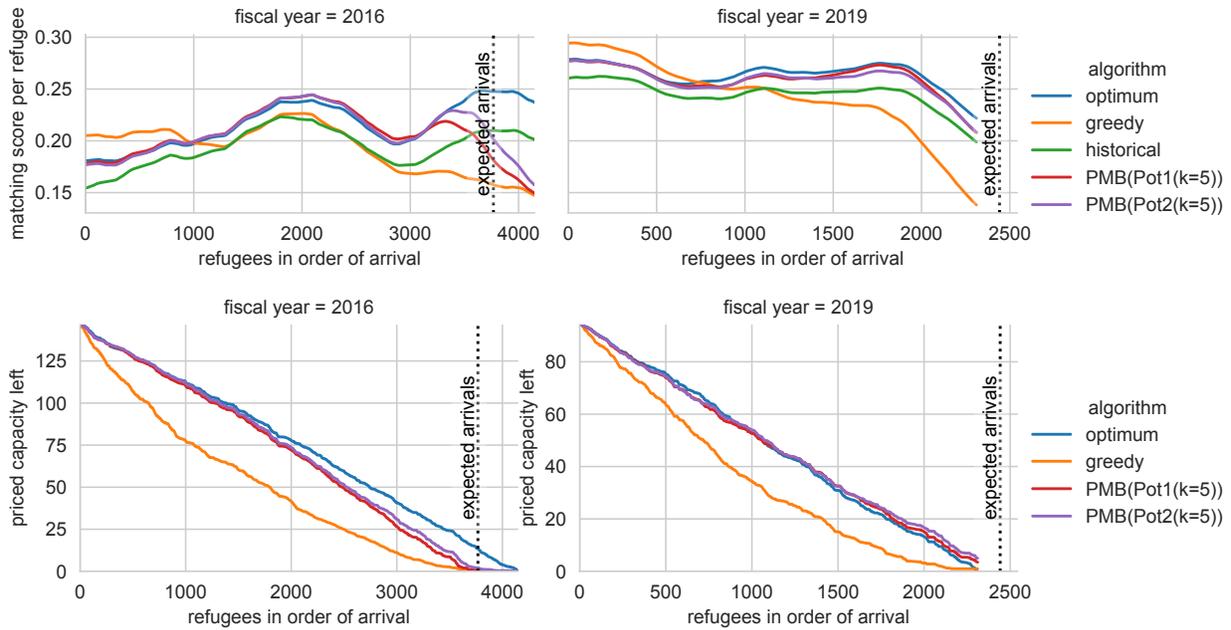
It is thus natural to run our potential algorithms under the assumption that the number of arriving refugees will be  $1/(110\%) \approx 91\%$  of the total announced capacity.<sup>8</sup> The result of this strategy is shown in [Figure 9.6](#). Since these experiments use the initial, unrevised capacities, the employment scores of the hindsight optimum and the greedy algorithm may differ from those in previous experiments, which used the most revised capacities.<sup>9</sup> In all fiscal years other than 2017 and 2018, the

8: To convert the number of remaining refugees into a number of cases, we divide by the average case size of recent arrivals (over the years, this average size fluctuates between 2.4 and 2.6). While the number of refugees who have arrived is below 91% of the total capacity, this gives us a total number of cases  $n$  for the algorithms. Once the number of arrivals exceeds 91% of the total capacity, we make the algorithms assume that the current case is the last to arrive, that is, all subsequently sampled trajectories have length zero.

9: This means that the comparison to the historical algorithm is not quite on equal terms, since the latter is constrained by a different set of capacities. In all fiscal years except for 2017 and 2018, the final capacities are element-wise larger than the original capacities.



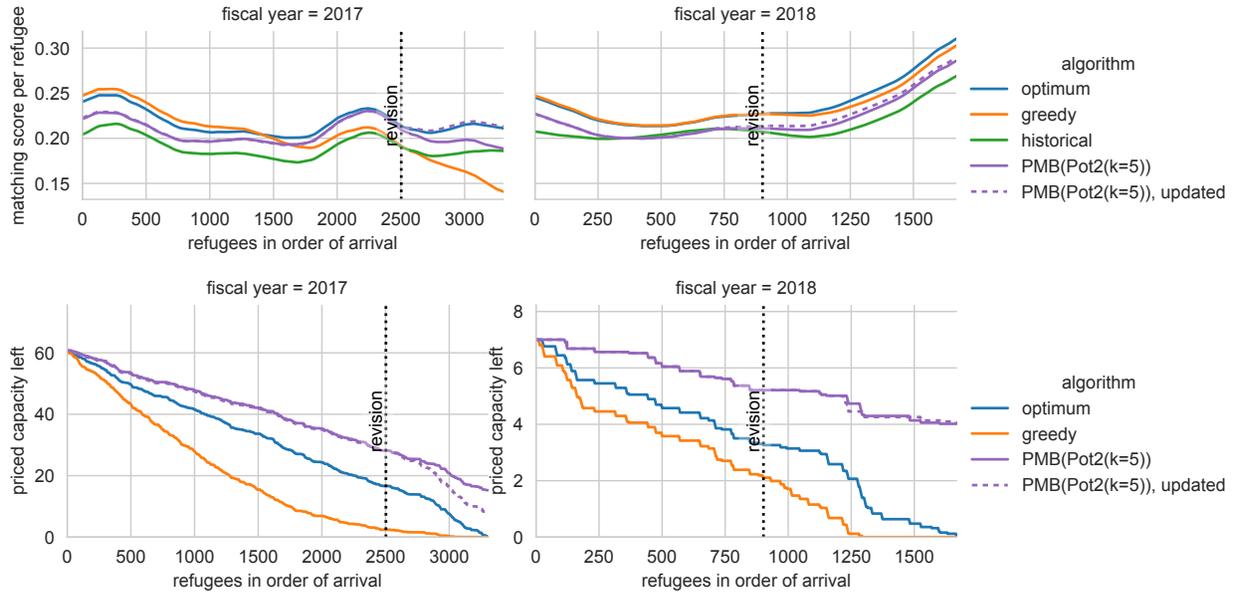
**Figure 9.6:** Total employment, where cases are not split up and arrive in batches. The potential algorithms no longer have access to the true number of arriving cases but assume that the arriving refugees amount to 91% of the total capacity. Capacities are the *initial* capacities of the fiscal year (except for historical). For the potential algorithms, the mean employment across 50 random runs is shown.



**Figure 9.7:** Evolution of the per-refugee match score and remaining priced capacity in order of arrival, for fiscal years 2016 and 2019 and one run per algorithm in the experiment of Figure 9.6 (whole cases, batches, initial capacities, potential algorithms do not know  $n$ ). Dotted line show how many refugees the potential algorithms expect. Smoothing as in Figure 9.2. Priced capacity is not shown for historical since it uses different capacities.

imprecise knowledge of future arrivals deteriorates the approximation ratio of the potential algorithms, but the potential algorithms continue to clearly outperform the greedy baseline overall, and they outperform the historical matching in every single year.

Setting aside the outlier years of 2017 and 2018 for the moment, we investigate the fiscal years 2016 and 2019, in which arrivals were otherwise highest and lowest relative to the announced capacity. In fiscal year 2016, the total arrivals were particularly large relative to the initial capacity: the arrival numbers added up to 100% of the initial capacity rather than 91%, which means that our potential algorithms expected around 3 770 refugees to arrive rather than the 4 150 that ended up arriving. As a result, the potential algorithms consume the priced capacity at an approximately constant rate, consuming it all around the expected number of expected refugees (Figure 9.7, bottom left). Up to this point, the potential algorithms are more generous in consuming capacity than would be ideal given the actual number of arriving cases, which is why the potential algorithms obtain a slightly higher average employment over the first three quarters of arrivals (Figure 9.7, top left) than the optimal matching in hindsight. For refugees arriving after the 3 770 expected refugees, however, the capacity



**Figure 9.8:** Evolution of the per-refugee match score and remaining priced capacity in order of arrival, for fiscal years 2017 and 2018 in the experiment of Figure 9.6 (whole cases, batches, initial capacities, potential algorithms do not know  $n$ ). Dashed line shows evolution if potential algorithm updates its expected arrival number at time of capacity revision (dotted line).

in the best affiliates is used up, which is why the averaged employment sharply drops after this point.<sup>10</sup>

In 2019, by contrast, fewer refugees arrived than expected, only 86% of the total capacity. At the bottom right of Figure 9.7, it is visible that the potential algorithms consume priced capacity at a slightly lower rate than the optimal algorithm in hindsight, as they aim to use up the capacity around 2 440 refugees rather than the 2 310 who ended up arriving. This effect is reflected in the average employment rates (top right), which lie below that of the optimal algorithm throughout most of the year.<sup>11</sup>

The fiscal years of 2017 and 2018 stand out due to the fact that the total number of arriving refugees fell far short of the announced number reflected in the approved capacities: in 2017, arrivals amounted to 65% of the approved capacities, while they amounted to only 46% in 2018. Both of these years fall into the beginning of the Trump administration, which not only sharply reduced the announced intake of resettled refugees, but furthermore abruptly halted the intake of refugees from six predominantly Muslim countries starting from early 2017.

As the potential algorithm depicted in Figure 9.8 severely overestimates how many cases will arrive, it holds back much more priced capacity than would be optimal (bottom, solid lines). This causes the potential algorithms to extract less employment throughout the year than the optimal algorithm (top, solid lines). As observed in Section 9.4.2, the capacities in 2018 are so loose that the greedy algorithm performs close to optimal.

In these two years, the US Department of State eventually reacted by correcting the expected arrivals downward and instructing the resettlement agencies to reduce their capacities. In fiscal year 2017, this revision came quite late and ended up underestimating the arrivals: where the arrivals amounted to only 65% of

10: Note that, due to the triangle smoothing, the drop starts dragging down the curve 500 arrivals before its actual start.

11: The drop in employment probabilities at the end of the fiscal year affects all algorithms including the hindsight optimum and must therefore be caused by an anomaly in arrival characteristics.

the initial capacities, they exceeded the revised total capacity at a level of 103%, rather than amounting to the 91% that was intended. Even if imperfect, this signal that arrivals are much lower than originally announced is still useful to the potential algorithms. Indeed, in [Figure 9.8](#), the dashed curve corresponds to a potential algorithm that still starts out expecting 91% of the initial capacities to arrive, but expects only 91% of the revised capacities to arrive from the point on where they were announced (vertical line). While this information comes late, the algorithm in fiscal year 2017 uses the new information to burn through the remaining priced capacity more aggressively (bottom left), which allows for higher employment among refugees arriving after the revision of arrival numbers (top left). As a result, the employment reaches 97% of the optimum in hindsight, exceeding the value of 95% without the updated information that we showed in [Figure 9.6](#).

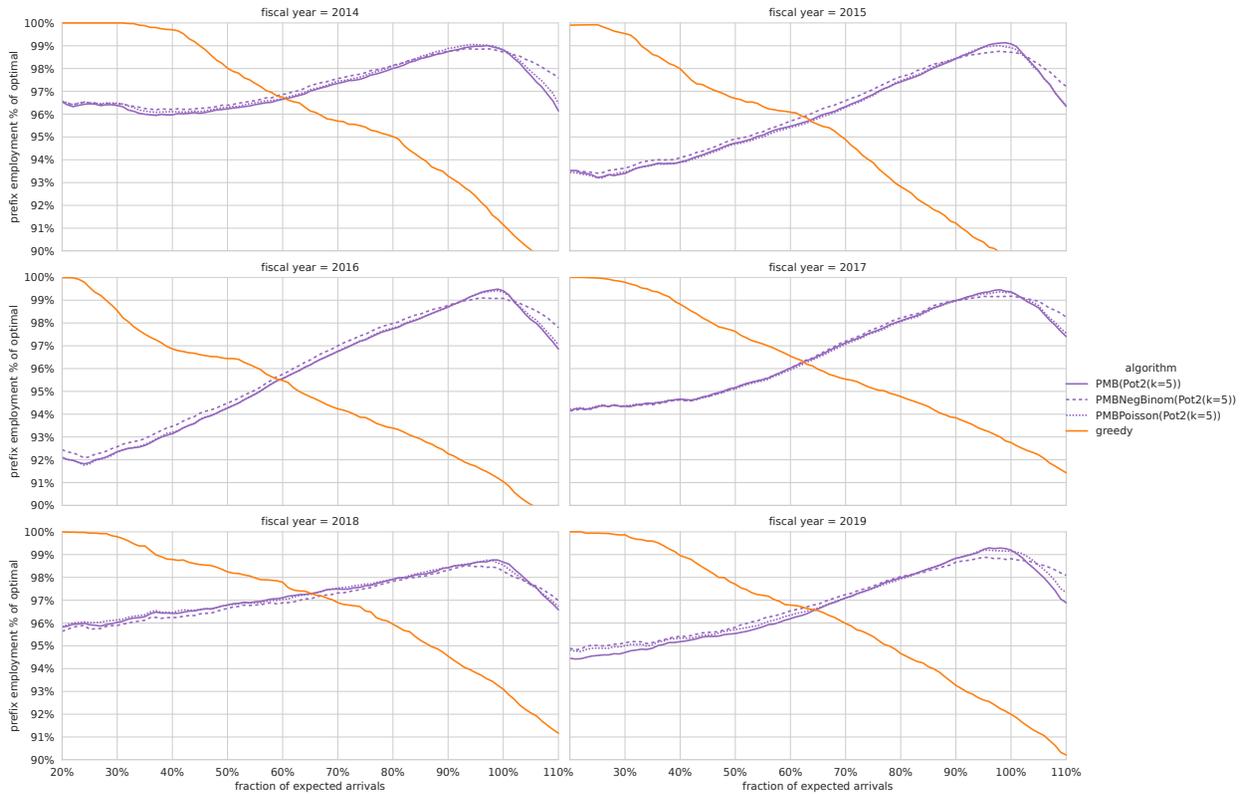
By contrast, the revision in fiscal year 2018 did not yield much useful information; whereas the arrivals amounted to 46% of the initial capacities, they still amounted to 48% of the revised capacities. This seems to indicate that, even after half of the fiscal year's refugees had already been allocated, the administration overestimated the number of arriving refugees by a factor of two. Because the revision barely changed the number of expected arrivals, giving the potential algorithm access to this revised information does not have much effect ([Figure 9.8](#), right).

While we have considered the informational value of revisions above, our experiments have not considered that these revisions actually reduced the allowable capacities. Although we include a variant of the experiment in [Appendix D.6](#) of the full version, it is difficult to meaningfully compare the employment achieved by different algorithms if the parameters of the matching problem are changed so drastically during the matching period. One particular challenge is that, while the amount of reduction was extraneously decided, HIAS was involved in deciding which capacities to decrease, which was done in a way that depended on previous allocation decisions.<sup>12</sup> Since we only know the revised capacities that were agreed upon, not the counterfactual revision of capacities that would be made, the greedy algorithm and the potential algorithms might have already exceeded a reduced capacity before it was announced. This means that the experiment rewards algorithms for greedily using up the capacity in the best affiliates before the revision, which we do not expect to be a good policy in practice. More generally, a substantial change in capacities is an exceptional situation, outside of our model, and cannot be addressed by our algorithm alone without manual intervention.

12: While the sum of capacities did not change much in fiscal year 2018, the capacities of some affiliates were substantially decreased and those of others were substantially increased.

### 9.7.2 Arrival Misestimation on Bootstrapped Data and Incorporating Uncertainty

To obtain more systematic insights into the robustness of potential algorithms to misestimated arrival numbers, we study bootstrapped case arrivals, which allows us to simulate varying numbers of arrivals. The results of this experiment are displayed in [Figure 9.9](#). As a baseline, consider the greedy algorithm, which obtains optimal employment when the number of arrivals is much lower than the total capacity (say, 25% of the expected arrivals, which is  $\frac{25\%}{110\%} \approx 23\%$  of the capacity), but becomes more and more suboptimal the more refugees arrive.



**Figure 9.9:** Employment achieved by different algorithms as a function of how many refugees arrive. Refugee arrivals are bootstrapped over each fiscal year’s historical arrivals, and the number of arriving refugees are given as a fraction of the historical arrivals. Capacities are 110% of historical matching. Employment is measured as a ratio of the optimal hindsight employment for the same set of arriving refugees. Curves are averaged over 10 arrival sequences.

By contrast, the potential algorithms perform best (around 99% of the optimal employment) when the number of arriving refugees matches what the algorithm expects. On average, this number is around half of a percentage point higher than in the corresponding non-bootstrapped experiments (Figure 9.4). Such an increase is to be expected as the bootstrapping setup ensures that the algorithm draws trajectories from the same distribution from which the arrivals are generated. In particular, the real arrival sequence used for Figure 9.4 might contain a drift in refugee characteristics or a seasonality not captured by our algorithm, and the lack of these features in the bootstrapped experiment allows for slightly higher employment. It is just as noticeable, however, that this increase is *only* half a percentage point, revealing that a drift of arrival characteristics and seasonality does not account for most of the remaining optimality gap of our algorithm.

The further the actual arrival number deviates from this expectation, the further the relative employment performance of the potential algorithm decreases. Noticeably, the performance more quickly deteriorates when the arrival numbers exceed the expectation, versus falling short. This sharp decline makes sense for two reasons. First, the algorithms aim to exploit all useful capacity exactly at the expected number of refugee arrivals; thus, only a subset of the affiliates remain available for subsequent arrivals. Second, once the number of arrivals exceeds the expectation, the trajectories in the potential algorithms add no cases beyond those that have already arrived, which means that the algorithm serves

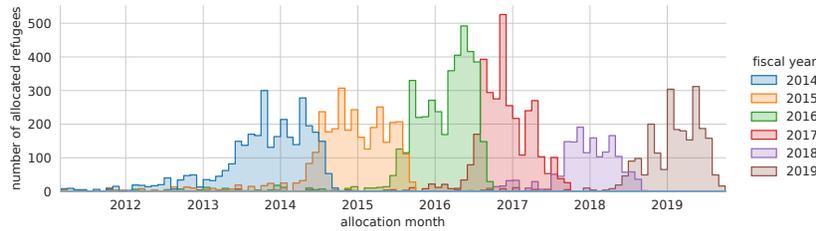
subsequent arrivals greedily. In the six fiscal years we observe, arrivals below the expectation seem like a more urgent problem than arrivals above the expectation, but over-arrivals might well become a problem under different political circumstances or when applying potential algorithms to other matching settings.

A natural way to make the potential algorithms more robust to inaccurate arrival estimates is to treat arrival estimates not as exact predictions but as subject to some uncertainty. Concretely, we adapt the potential algorithms by sampling trajectories of different lengths, each drawn from a “prior” distribution whose mean is the arrival estimate, conditioning this distribution such that trajectory lengths are never less than the number of refugees who have already been allocated. Conceivably, these adapted trajectories could generate potentials that are robust across a wider range of arrival numbers, and the adapted algorithm could therefore lead to higher employment when the official arrival numbers are inaccurate. The most obvious distribution is perhaps a Poisson distribution. As shown by the dotted line in [Figure 9.9](#), using Poisson trajectories hardly changes the employment outcomes for any of the experiments relative to the baseline of fixed trajectory sizes. This is most likely due to the low variance of the Poisson distribution. For a quite typical mean of 3 000 arriving refugees, 95% of the probability mass lies within a distance of only 3.6% of the mean. For this reason, we also try a distribution with overdispersion, specifically a negative binomial distribution parameterized to have its mean equal to the expected arrivals and its standard deviation equal to 10% of the expected arrivals. For example, if again 3 000 arrivals are expected, 95% of the probability mass deviates up to 20% from the mean. As the figure shows, negative-binomial trajectories lead to decent improvements in employment when more refugees arrive than expected. When fewer refugees arrive than expected, using random trajectory lengths helps more often than not, though with different degrees of success. Overall, negative-binomial arrivals seem to make the potential algorithms marginally more robust to misestimated arrival numbers, though not by enough to make misestimation less of an overall concern. Additionally, this additional robustness comes at a nonnegligible cost when arrival estimates are accurate.

### 9.7.3 Better Knowledge of Future Arrivals

In [Section 9.7.1](#), we demonstrated that, even without outside supervision, our potential algorithms lead to substantial employment increases over the baselines, unless the announced capacities miss the eventual arrival numbers by an extreme margin. Even in these typical years, however, more accurate arrival predictions could increase the total employment on the order of percentage points of the hindsight optimum. Obviously, more accurate information about arrivals would be even more useful in years like 2017 and 2018, in which the official information is unreliable.

One approach would be to use time-series prediction to estimate the number of arrivals. For instance, when the US Department of State revised the capacities for the fiscal year 2018 in January 2018 (several months into the fiscal year), the announcement that 2.5 times more refugees were still to come than had already arrived might have raised some doubts. However, the graph of monthly arrivals



**Figure 9.10:** Monthly number of allocated refugees, disaggregated by fiscal year of arrival.

in Figure 9.10 shows that late increases in arrival rates may actually happen as they did in fiscal year 2016.<sup>13</sup>

A fundamental challenge that any data-driven approach faces is that there is very little data to learn from. Indeed, while HIAS has data on hundreds of thousand of refugees, they only have data on 15 fiscal years, which is, moreover, incomplete and smaller-scale in earlier years. Thus, there is a limited foundation to learn about how arrival patterns change between years. This task becomes especially difficult given that arrival numbers are heavily influenced by external events such as elections, the emergence of humanitarian disasters, and changes in immigration policy, which cannot be deduced from past arrival patterns. Thus, while a time-series prediction approach might lead to marginal improvements over naïvely expecting 91% of the capacity to arrive, past arrival numbers are unlikely to give enough information to accurately predict future arrival numbers.

Fortunately, resettlement agencies such as HIAS already possess much richer information and insights into the dynamics of refugee arrivals than a pure data approach would consider. In fiscal year 2017, for example, HIAS foresaw a worsening climate for refugee resettlement immediately after the November 2016 election<sup>14</sup> and was aware of concrete plans to drastically reduce refugee intake in January 2017,<sup>15</sup> both before these changes were reflected in arrival numbers and before the capacities were officially updated in March 2017. Similarly, HIAS continuously monitors domestic politics and international crises for their potential impact on resettlement, and moreover it has some limited insight into the resettlement pipeline, which allows it to prepare for changes in arrivals. We therefore believe that, rather than building a sophisticated tool for predicting arrivals in a fully autonomous manner, it is preferable to allow HIAS staff to override our prediction with more advanced information.

## 9.8 Implementation in Annie™ MOORE

To enable HIAS to benefit from dynamic allocation via potentials, we have integrated new features into its matching software Annie™ MOORE. A crucial design requirement is that HIAS staff must be able to override the allocation recommendations of Annie™ when they are aware of requirements outside of our model. From an interface-design perspective, the challenge is to visualize the effect of such overrides on total employment, enabling HIAS staff to make informed trade-offs. In the original, static model, this was easy enough: as the quality of a matching was just the total employment of the current batch, the interface labeled each case–locality match with its associated employment score, and staff could drag the case to other localities to see the respective employment scores.

13: In fiscal year 2016, the number of arrivals after January 2016 was 1.6 times larger than the number that had arrived so far. In the fiscal year of 2015, the number of refugees arriving after January 2015 was only 75% of that arriving before.

14: <https://www.hias.org/news/press-releases/hias-calls-president-elect-trump-respect-longstanding-refugee-policy>

15: <https://www.hias.org/news/press-releases/trumps-planned-action-refugees-betrayal-american-values>

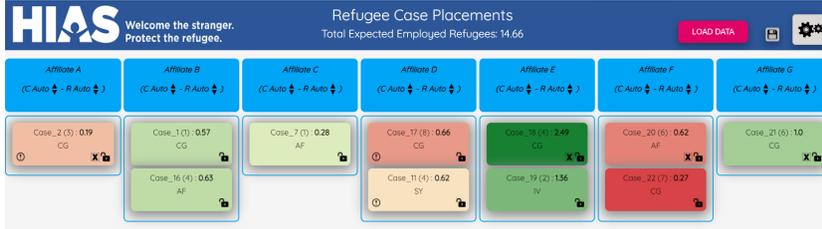


Figure 9.11: Updated Annie™ Interface. Family tiles now show both original numerical employment scores of families in affiliates, as well as the *adjusted* employment score by its shading. Green indicates positive adjusted scores, red negative scores, and darker colors represent greater magnitudes.

In a dynamic setting, however, presenting only the employment scores may unintentionally encourage HIAS staff to greedily use capacity in their overrides, at the expense of future arrivals.

As we illustrate in Figure 9.11, the new interface of Annie™ augments the original interface with information about affiliate potentials, thereby taking future arrivals into account. Specifically, the background color of the tile for case  $i$  encodes the *adjusted* employment score, that is, the original employment score  $u_{i,\ell}$  less the potential  $s_i p_\ell$  of the capacity consumed in affiliate  $\ell$ .<sup>16</sup> The fact that the algorithm PMB always maximizes the sum of adjusted employment scores in its allocation of the current batch means that the algorithm is explainable in terms of the information presented to the user. In the interface, the *green* color spectrum indicates positive adjusted employment scores (meaning that the employment score of the case outweighs the loss in future employment), while the *red* color spectrum highlights negative adjusted scores (where a placement reduces future employment by more than its employment score). Darker colors signify greater magnitudes.

In overriding the allocation recommended by Annie™, HIAS staff should be able to quickly find alternative placements for a case that do not reduce immediate and future employment by more than necessary. To support this workflow, our interface shows the adjusted employment scores of a case across all affiliates at a glance: as shown in Figure 9.12, upon dragging a particular case tile from its current placement, all other case tiles temporarily fade in appearance, and the shading of every affiliate tile temporarily assumes the adjusted employment score relative to the selected case. By hovering a selected case tile over a new affiliate, the original (numeric) employment score and the adjusted match score (background color of the case tile) dynamically update. Moreover, incompatibilities with affiliates due to nationality, language, family size, and single parent households can be seen via an exclamation mark in the lower left corner of the affiliate tile. After dropping the case tile in a new affiliate, the background color for each affiliate returns to its original blue shade, and all affiliate-tile exclamation marks disappear.

On a separate screen (not shown), Annie™ enables the entry of a prediction for total refugee arrivals, as mentioned in Section 9.7.3. This estimate can be critical to inform the process of estimating proper shadow prices, as at times HIAS is in a better position to give more accurate case arrival predictions than officially announced capacities.

16: The employment scores of cases in affiliates are prominently retained in a text label.



**Figure 9.12:** Moving a family tile. Other case tiles fade, and affiliate tiles are colored as per their adjusted employment scores in shades of green (positive) or red (negative). Exclamation marks indicate incompatibilities.

## 9.9 Discussion

We have developed and implemented algorithms for dynamically allocating refugees in a way that promotes refugees' prospects of finding employment. These algorithms outperform the baselines, even when taking into account how refugee placement in practice deviates from a classic matching setting.

While we have tested the algorithms as an autonomous system, the success of Annie™ in increasing employment outcomes in practice will depend on how it performs in interaction with HIAS resettlement staff. In Section 9.7.3, we already saw that the allocation decisions of Annie™ can greatly profit from human decision makers providing better estimates of future arrivals. Human input is equally crucial in dealing with uncertainty in several other places: for example, HIAS staff might intervene by correcting the arrival year of a case should the Department of State's estimate seem off, or they might increase some affiliate capacities late in the year if they anticipate that these capacities can be increased. By allowing all parameters of the matching problem to be changed, Annie™ allows HIAS resettlement staff to improve the matching using all available information.

Ideally, the human-in-the-loop system consisting of the matching algorithm and HIAS staff can combine the strengths of both of its parts: On the one hand, the algorithms in Annie™ capitalize on subtle patterns in employment data and manage capacity more effectively over the course of the fiscal year. On the other hand, the expert knowledge of HIAS staff enables the system to handle the uncertainty that is inherent in a matching problem involving the actions of multiple government agencies, dozens of affiliates, and thousands of refugees. In light of the administration's recent increase of the total resettlement capacity from 15 000 to 125 000,<sup>17</sup> we foresee both parts playing a crucial role: the increasing scale of the problem will make data-based algorithms more effective, and human guidance will be necessary to navigate the evolving environment of a rapidly growing operation.

17: <https://www.hias.org/news/press-releases/refugee-cap-fy2022-set-125000>

## CONCLUSIONS

Throughout the chapters of this thesis, we have already drawn conclusions from individual pieces of research. We conclude by reflecting more globally on the work included, by answering two questions: How can computer science guide democratic innovations, and which future directions can advance the study of democracy in computer science?

10.1 How Can Computer Science Guide Democratic Innovations? . . . . .	157
10.2 Future Directions for the Study of Democracy in Computer Science . . . . .	159

## 10.1 How Can Computer Science Guide Democratic Innovations?

Why, of all sciences, should computer science study democracy, and what does it fundamentally have to offer? In the introduction, we answered a narrow version of this question, by identifying three different modes of our work: designing algorithms, analyzing processes, and identifying alternative processes. Here, we posit three broader answers (neither disjoint nor exhaustive) to these guiding questions, on which we elaborate below:

1. Computer scientists can bring expertise to practitioners' algorithms.
2. Computer scientists can provide a precise language for the desiderata of democratic processes.
3. Computer scientists can support the daily operations of practitioners.

### Answer 1: Bringing Algorithmic Expertise to Practitioners' Algorithms

Whether computer scientists are involved or not, democratic practitioners develop and use algorithms. In some cases, such as the baselines in [Chapters 2](#) and [5](#), these algorithms are already implemented as code and publicly accessible.<sup>1</sup> In other cases, these algorithms are harder to spot, which was the case for four additional selection algorithms that we learned of over the course of our discussions with other sortition organizations.<sup>2</sup> These organizations did not think of their sampling procedures as algorithms, and typically implemented them not in code but using spreadsheets, dice, and flexibility for manual intervention. Nonetheless, the goal of making combinatoric decisions in an impartial way pushed each organization towards a distinct algorithmic approach.

A first contribution that computer scientists can make to these naturally emerging algorithms is the methodology of algorithmic analysis. In our experience, questions that are natural from a computer science perspective played little role in the discussion between practitioners prior to our involvement: *Which properties does an algorithm satisfy? Which inputs does the algorithm perform worst<sup>3</sup> on? Which algorithms are better than others?* In both [Chapters 2](#) and [5](#), our work started by explaining shortcomings of the existing algorithms to practitioners, which built the foundation for our development of new algorithms.

1: In both cases, this availability was made possible by the Sortition Foundation's unusual technical expertise and its commitment to open sourcing their tools.

2: See [supplementary information 12](#) of Flanigan et al. [FGG+21].

3: "Worst" and "better" need not refer to running time, but might also refer to the quality of the solution, for example some notion of the solution's fairness.

As a second contribution, where limitations of the existing algorithms call for the development of a new one, computer scientists can employ a large toolbox of algorithmic techniques. In this thesis, we mostly applied techniques from optimization (integer linear programming, column generation, submodular optimization, linear programming duality), but most techniques taught in courses on algorithm design are likely to be new to practitioners and therefore potentially valuable.

## Answer 2: Providing a Precise Language for Desiderata

The field of political theory, where the desiderata of democratic innovations are primarily discussed, formulates these goals in abstract terms that leave many details unspecified. For example, Carson and Martin [CM99] categorize the benefits of sortition into “(1) promotion of equality, (2) representativeness, (3) efficiency, and (4) protection against conflict and domination.” Trying to preserve these benefits gives some direction in designing a selection algorithm (see [Appendix A.1](#)), but political theory does not make these desiderata so concrete as to, say, make them measurable. In the few cases where political theory discusses practical details, this is often in the context of a critique: For instance, political theorists might point out that certain desiderata *cannot* be realized in practice, but typically without suggesting what practitioners *should* do in the face of these impossibilities.

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

Due to computer science’s roots both in mathematics and engineering, computer scientists tend to formulate desiderata in a way that is highly different from and complementary to political theory. First, since (theoretical) computer scientists analyze processes through mathematics, the desiderata they study are mathematical statements whose satisfaction can be objectively determined.<sup>4</sup> For example, in [Chapters 2](#) and [3](#), we operationalized “promotion of equality” with the more specific interpretation of “each agent has an equal marginal probability of being selected to the panel.” Second, computer science’s origins in engineering might explain why computer scientists tend to take on a constructive rather than critical stance: when computer scientists show that a desirable property is currently violated (or even impossible to satisfy), they are expected to *build* an alternative process that satisfies the property (or an approximate version of it). For instance, whereas multiple political theorists have pointed out that equal chances of selection cannot be achieved in practical sortition, we were the first to suggest a notion of “as close to equal chances as possible” in [Chapter 2](#).

4: This point is related to the role of computing as a “formalizer” in promoting social change in the categorization by Abebe et al. [ABK+20].

Of course, the choice of objectives is up to political theorists and practitioners, not computer scientists. But work in computer science can act as a catalyst for discussions among those former two groups: by confronting practitioners with the choice between different algorithms, and by explaining the properties underlying the computer science work, practitioners and political theorists are nudged to more precisely state what they see as the goal.

## Answer 3: Supporting Practitioners’ Daily Operations

By offering hands-on support to practitioners, computer scientists can contribute a technical expertise that is typically outside the main focus of democratic orga-

nizations. Throughout the work on this thesis, this support of course included helping practitioners use our website and tools, which has immediate benefits for both sides. But our support extended to tasks not directly related to research, such as building websites and forming diverse focus groups for screening materials.

Besides the immediate benefits of such close collaboration to practitioners, it allows researchers to learn about practical problems and practitioners to learn the range of what researchers can do. Both of these learning processes help identify opportunities for computer science to support and inform practitioners' efforts. For example, the following three questions emerged in our collaborations and are prime targets for future work:

- ▶ How can we incorporate the fact that some panel members drop out of the assembly and must be replaced into our conception of fairness?
- ▶ How can panel organizers trade off the cost of the recruitment process with the level of representativeness and fairness?
- ▶ If a panel organizer prefers to use a selection algorithm that can be executed without a computer, which one should they use?

## 10.2 Future Directions for the Study of Democracy in Computer Science

Going forward, we identify three priorities for directing our field towards interesting mathematics, practical impact, and coherence: deliberation, incentivizing participation, and engaging with normativity.

### Direction 1: Deliberation as the Frontier of Computational Social Choice

Traditionally, the field of computational social choice<sup>5</sup> studies the *aggregation* of individual preferences through voting. The success of deliberative democracy, however, raises the question whether *deliberation*, not aggregation, should be seen as the ideal of democracy — and how computational social choice can reflect this shift.

Recent papers [CD20; FGMS17; GL16; PP15; ZLT21] have taken first steps on this path by proposing procedural models of deliberation. So far, however, models of deliberation are “toy models” that are only loosely inspired by deliberation; as a result, these models can hardly claim to be stand-ins for deriving insights about real-world deliberation. One way to develop more realistic models is to root them firmly in the deliberation literature. A good starting point for this agenda could be a paper by Dryzek and List [DL03], which disentangles the “informational, argumentative, reflective and social” effects of deliberation, which is a much broader perspective on deliberation than the one that underlies existing models. There is also an urgent need to develop theoretical models based on empirical observations of deliberation, since such work can identify the most important conditions for promoting the success of deliberation in practice. Once

5: And, more generally, its parent discipline of social choice.

[CD20] Chung and Duggan (2020): *A Formal Theory of Democratic Deliberation*.

[FGMS17] Fain et al. (2017): *Sequential Deliberation for Social Choice*.

[GL16] Goel and Lee (2016): *Towards Large-Scale Deliberative Decision-Making*.

[PP15] Perote-Peña and Piggins (2015): *A Model of Deliberative and Aggregative Democracy*.

[ZLT21] Zvi et al. (2021): *Iterative Deliberation via Metric Aggregation*.

[DL03] Dryzek and List (2003): *Social Choice Theory and Deliberative Democracy*.

such models are available, problems such as our group-allocation problem in [Chapter 5](#) can and should be revisited with much greater depth.

Besides modeling deliberative processes, the success of deliberation raises questions about how computational social choice conceives of agents: *What drives real people towards successful deliberation, and how can we represent this drive in our agent models?* Currently, the agents studied in computational social choice have fully-formed preferences, know the precise effects of possible outcomes, are purely self-interested, and pursue this interest in a coldly calculating manner. If real-world agents indeed had these characteristics, we would not expect deliberation to have much effect on group decision making. By contrapositive, the impact of deliberation should make us look for what our models are missing — new forms of partial information and bounded rationality, altruism, or social norms?

## Direction 2: Designing Processes in Which Voting Feels Worthwhile

One of the worrying signs for the state of democracy is a global decrease in voter turnout over the past decades [[KB21](#)]. How can democratic innovations encourage larger turnout? One possible lever is an agent's *expected benefit of voting*, which is the probability that the agent's vote changes the outcome times the expected change in the agent's utility conditioned on their vote changing the outcome. It is theoretically and empirically well-supported that the expected benefit influences turnout [[AK75](#); [FOSY78](#); [RO68](#)], but increasing the benefit of voting is also an appealing goal in its own right: over time, voters should feel that participation in the democratic process “pays off.” In single-winner plurality elections (say, most national presidential elections), the expected benefit of voting tends to be low because the probability of casting a pivotal vote is minuscule, even more so when one candidate has a stable lead in support.

Other forms of elections offer more opportunities for votes to change the outcome, and can thus potentially make voting more rewarding. Participatory budgeting seems like a particularly promising candidate since its outcome structure (i.e., sets of budget-feasible projects) allows to satisfy both majority and minority interests with part of the budget. Recent work on voting rules that provide strong proportionality guarantees in participatory budgeting [[PPS21](#)] is exciting from this angle because these rules ensure a form of responsiveness: if a project (say, renovating a local school) would not be funded by default, but a modest number of agents join the voting process from the political sidelines to support this project, their participation must be rewarded by the project getting funded. In the best case, such responsiveness might incentivize marginalized communities to turn out at higher rates and thus decrease the demographic distortions in voter turnout [[LN14](#)]. It seems highly attractive to strengthen these proportionality guarantees (especially towards the elusive notion of *core stability* [[PPS21](#)]), but also to think whether the perspective of encouraging participation inspires other axioms for participatory budgeting.

Underlying these technical directions, the fundamental objective is a political one: to create spaces where constituents can experience democracy benefitting their daily lives. In a time of democratic cynicism [[OEC22](#); [WSC19](#)], in which adherents

[[KB21](#)] Kostelka and Blais (2021): *The Generational and Institutional Sources of the Global Decline in Voter Turnout.*

[[AK75](#)] Ashenfelter and Kelley (1975): *Determinants of Participation in Presidential Elections.*

[[FOSY78](#)] Frohlich et al. (1978): *A Test of Downsian Voter Rationality.*

[[RO68](#)] Riker and Ordeshook (1968): *A Theory of the Calculus of Voting.*

[[PPS21](#)] Peters et al. (2021): *Proportional Participatory Budgeting with Additive Utilities.*

[[LN14](#)] Leighley and Nagler (2014): *Who Votes Now?*

[[OEC22](#)] OECD (2022): *Building Trust to Reinforce Democracy.*

[[WSC19](#)] Wike et al. (2019): *Many Across the Globe Are Dissatisfied With How Democracy Is Working.*

of technocracy challenge the possibility of effective democratic governance [Bel15; Kha17], proving the opposite is vital to defending democracy.

[Bel15] Bell (2015): *The China Model*.

[Kha17] Khanna (2017): *Technocracy in America*.

### Direction 3: Embracing Normativity

As computer scientists, we are trained to derive factual statements such as “in model X, algorithm Y satisfies property Z” or “on dataset X, algorithm A runs faster on average than algorithm B,” which philosophers refer to as *positive* statements. But when working on democracy, motivating our work or defining desirable properties is inherently *normative*, i.e., dependent on subjective notions of what is desirable for society. For example, a reader who is generally opposed to citizens’ assemblies, or who has a very different conception of the objectives of sortition, might not be convinced of the value of our work on selection algorithms. But, if motivations are subjective, how can our field avoid being a collection of arbitrary directions, just up to the authors’ personal beliefs?

One pragmatic tool for making directions less arbitrary is writing the motivation of the paper for a broad audience of democrats, which is ideologically and geographically unspecific. This approach is already the norm in the field. We have aimed to follow this approach in our work as well; for instance, we aimed to present citizens’ assemblies such that our work seems relevant not only to, say, proponents of far-reaching reforms towards sortition, but also to readers interested in citizens’ assemblies as a more modest instrument of citizen consultation.

A more far-reaching approach to avoid arbitrary directions we want to propose is for computer scientists to directly engage with the related normative literature. Disciplines such as political theory, in which normative statements play a large role, have long developed conventions that regulate how new positions must be defended and connected to existing thought to avoid arbitrariness. By anchoring itself to normative research, computer science can profit from these conventions. Admittedly, we have not done so for most of the work in this thesis, but have aimed to root our approach in Appendix A to political theory in Appendix A.1, which we see as a model for our future work.

In fact, the need of connecting computer science research to normative literature applies far beyond the study of democracy. Popular movements towards *artificial intelligence for social good* or *mechanism design for social good* by definition require normative justification for how a piece of work in this area does indeed serve the “social good.” Recently, there has been an increased focus on negative social consequences of information technology; for example, the machine learning conference NeurIPS instructed authors to discuss “potential negative societal impacts” of their work in 2022 [Neu22]. We believe that claims of *positive* social impact deserve equal attention, and more systematic attention than is currently the norm.

[Neu22] NeurIPS (2022): *NeurIPS Paper Checklist Guidelines*.

# Bibliography

- [AAM+21] Narges Ahani, Tommy Andersson, Alessandro Martinello, Alexander Teytelboym, and Andrew C. Trapp. *Placement Optimization in Refugee Resettlement*. In: *Operations Research* (2021).  (cit. on pp. 8, 133–135, 137).
- [AB16] Christopher H. Achen and Larry M. Bartels. *Democracy for Realists: Why Elections Do Not Produce Responsive Government*. Princeton Studies in Political Behavior. Princeton University Press, 2016 (cit. on p. 1).
- [AB18] Allianz Vielfältige Demokratie and Bertelsmann Foundation. *Citizens' participation using sortition*. 2018.   (cit. on p. 14).
- [ABC+17] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. *Justified Representation in Approval-Based Committee Voting*. In: *Social Choice and Welfare* 48.2 (2017), pp. 461–485.  (cit. on pp. 7, 96, 97, 99, 100, 108).
- [ABK+20] Rediet Abebe, Solon Barocas, Jon Kleinberg, Karen Levy, Manish Raghavan, and David G. Robinson. *Roles for Computing in Social Change*. In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2020, pp. 252–260.  (cit. on p. 158).
- [ABKU94] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. *Balanced Allocations*. In: *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*. 1994, pp. 593–602.  (cit. on pp. 7, 76, 80, 86).
- [ABM19] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. *Fair Mixing: The Case of Dichotomous Preferences*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2019, pp. 753–781.  (cit. on pp. 101–103).
- [ACE22] ACE Electoral Knowledge Network. *Electoral System (Chamber 1)*. 2022. [https://aceproject.org/epic-en/CDTable?question=ES005&view=country&set\\_language=en](https://aceproject.org/epic-en/CDTable?question=ES005&view=country&set_language=en) (visited on 06/21/2022).  (cit. on pp. 7, 93).
- [ADF17] Faez Ahmed, John P. Dickerson, and Mark Fuge. *Diverse Weighted Bipartite B-Matching*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 35–41 (cit. on p. 53).
- [ÅEFG11] Olof Åslund, Per-Anders Edin, Peter Fredriksson, and Hans Grönqvist. *Peers, Neighborhoods, and Immigrant Student Achievement: Evidence from a Placement Policy*. In: *American Economic Journal: Applied Economics* 3.2 (2011), pp. 67–95.  (cit. on p. 133).

- [AEH+18] Haris Aziz, Edith Elkind, Shenwei Huang, Martin Lackner, Luis Sánchez-Fernández, and Piotr Skowron. *On the Complexity of Extended and Proportional Justified Representation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2018 (cit. on pp. 107, 108).
- [ÅFo9] Olof Åslund and Peter Fredriksson. *Peer Effects in Welfare Dependence Quasi-Experimental Evidence*. In: *Journal of human resources* 44.3 (2009), pp. 798–825.  (cit. on p. 133).
- [AGG+15] Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, and Toby Walsh. *Computational Aspects of Multi-Winner Approval Voting*. In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2015, pp. 107–115 (cit. on p. 107).
- [AGP+21] Narges Ahani, Paul Gözl, Ariel D. Procaccia, Alexander Teytelboym, and Andrew C. Trapp. *Dynamic Placement in Refugee Resettlement*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2021, p. 5.  (cit. on p. 11).
- [AHL12] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. *Online Prophet-Inequality Matching with Applications to Ad Allocation*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2012, pp. 18–35.  (cit. on p. 134).
- [AHL13] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. *The Online Stochastic Generalized Assignment Problem*. In: *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 2013, pp. 11–25.  (cit. on p. 134).
- [AK75] Orley Ashenfelter and Stanley Kelley. *Determinants of Participation in Presidential Elections*. In: *The Journal of Law and Economics* 18.3 (1975), pp. 695–733.  (cit. on p. 160).
- [ALM+19] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. *Strategyproof Peer Selection Using Randomization, Partitioning, and Apportionment*. 2019 (cit. on pp. 113, 114).
- [Aló06] Carlos Alós-Ferrer. *A Simple Characterization of Approval Voting*. In: *Social Choice and Welfare* 27.3 (2006), pp. 621–625.  (cit. on p. 93).
- [AN20] Mohammad Akbarpour and Afshin Nikzad. *Approximate Random Allocation Mechanisms*. In: *The Review of Economic Studies* 87.6 (2020), pp. 2473–2510.  (cit. on p. 114).
- [ÅÖZ10] Olof Åslund, John Östh, and Yves Zenou. *How Important Is Access to Jobs? Old Question—Improved Answer*. In: *Journal of Economic Geography* 10.3 (2010), pp. 389–422.  (cit. on p. 133).
- [ÅRo7] Olof Åslund and Dan-Olof Rooth. *Do When and Where Matter? Initial Labour Market Conditions and Immigrant Earnings*. In: *The Economic Journal* 117.518 (2007), pp. 422–448.  (cit. on p. 133).

- [ASo4] Alexander A. Ageev and Maxim I. Sviridenko. *Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee*. In: *Journal of Combinatorial Optimization* 8.3 (2004), pp. 307–328.  (cit. on p. 114).
- [ASo8] Alastair Ager and Alison Strang. *Understanding Integration: A Conceptual Framework*. In: *Journal of Refugee Studies* 21.2 (2008), pp. 166–191.  (cit. on p. 133).
- [AWY14] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. *A Dynamic Near-Optimal Algorithm for Online Linear Programming*. In: *Operations Research* 62.4 (2014), pp. 876–890.  (cit. on p. 135).
- [AWZ20] Arash Asadpour, Xuan Wang, and Jiawei Zhang. *Online Resource Allocation with Limited Flexibility*. In: *Management Science* 66.2 (2020), pp. 642–666.  (cit. on p. 135).
- [BA99] Albert-László Barabási and Réka Albert. *Emergence of Scaling in Random Networks*. In: *Science* 286.5439 (1999), pp. 509–512.  (cit. on pp. 66, 76, 77).
- [BAL+22] Vanessa A. Boese, Nazifa Alizada, Martin Lundstedt, Kelly Morrison, Natalia Natsika, Yuko Sato, Hugo Tai, and Staffan I. Lindberg. *Democracy Report 2022: Autocratization Changing Nature? Varieties of Democracy Institute*. 2022. [https://v-dem.net/media/publications/dr\\_2022.pdf](https://v-dem.net/media/publications/dr_2022.pdf)  (cit. on p. 1).
- [Ban19] Nikhil Bansal. *On a Generalization of Iterated and Randomized Rounding*. In: *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*. 2019, pp. 1125–1135.  (cit. on pp. 24, 27).
- [Ban20] K. Bansak. *A Minimum-Risk Dynamic Assignment Mechanism Along with an Approximation, Heuristics, and Extension from Single to Batch Assignments*. 2020.  (cit. on pp. 135, 140, 145).
- [BBCV11] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. *Viscous Democracy for Social Networks*. In: *Communications of the ACM* 54.6 (2011), pp. 129–137.  (cit. on p. 68).
- [BBD00] P. S. Bradley, K. P. Bennett, and Ayhan Demiriz. *Constrained K-Means Clustering*. In: *Microsoft Research, Redmond* (2000), pp. 1–8 (cit. on p. 48).
- [BBPS21] Florian Brandl, Felix Brandt, Dominik Peters, and Christian Stricker. *Distribution Rules Under Dichotomous Preferences: Two Out of Three Ain't Bad*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2021, pp. 158–179.  (cit. on p. 102).
- [BCo8] Salvador Barberà and Danilo Coelho. *How to Choose a Non-Controversial List with  $k$  Names*. In: *Social Choice and Welfare* 31.1 (2008), pp. 79–96.  (cit. on pp. 7, 97).
- [BCC96] Joaquín Bautista, Ramon Companys, and Albert Corominas. *A Note on the Relation between the Product Rate Variation (PRV) Problem and the Apportionment Problem*. In: *Journal of the Operational Research Society* 47.11 (1996), pp. 1410–1414 (cit. on p. 114).

- [BCKM13] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. *Designing Random Allocation Mechanisms: Theory and Applications*. In: *American economic review* 103.2 (2013), pp. 585–623.  (cit. on p. 130).
- [BCKO17] Eric Budish, Gérard P. Cachon, Judd B. Kessler, and Abraham Othman. *Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation*. In: *Operations Research* 65.2 (2017), pp. 314–336.  (cit. on p. 15).
- [Bec12] Sven Becker. *Web Platform Makes Professor Most Powerful Pirate*. 2012. <http://www.spiegel.de/international/germany/liquid-democracy-web-platform-makes-professor-most-powerful-pirate-a-818683.html>  (cit. on p. 65).
- [Bel15] Daniel Bell. *The China Model: Political Meritocracy and the Limits of Democracy*. Princeton University Press, 2015 (cit. on p. 161).
- [BFo7] Steven J. Brams and Peter C. Fishburn. *Approval Voting*. 2nd edition. Springer, 2007 (cit. on p. 93).
- [BF81] József Beck and Tibor Fiala. “Integer-making” Theorems. In: *Discrete Applied Mathematics* 3.1 (1981), pp. 1–8.  (cit. on pp. 4, 24, 27).
- [BFGP22] Yushi Bai, Uriel Feige, Paul Gözl, and Ariel D. Procaccia. *Fair Allocations for Smoothed Utilities*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2022.  (cit. on p. 11).
- [BFH+18] Kirk Bansak, Jeremy Ferwerda, Jens Hainmueller, Andrea Dillon, Dominik Hangartner, Duncan Lawrence, and Jeremy Weinstein. *Improving Refugee Integration through Data-Driven Algorithmic Assignment*. In: *Science* 359.6373 (2018), pp. 325–329.  (cit. on pp. 133, 134).
- [BFJL17] Markus Brill, Rupert Freeman, Svante Janson, and Martin Lackner. *Phragmén’s Voting Methods and Justified Representation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (2017).  (cit. on p. 107).
- [BG21] Avrim Blum and Paul Gözl. *Incentive-Compatible Kidney Exchange in a Slightly Semi-Random Model*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2021.  (cit. on p. 11).
- [BG22] Yushi Bai and Paul Gözl. *Envy-Free and Pareto-Optimal Allocations for Agents with Asymmetric Random Valuations*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2022 (cit. on p. 11).
- [BGG+22] Jake Barrett, Kobi Gal, Paul Gözl, Rose Hong, and Ariel D. Procaccia. *Now we’re talking: Better deliberation groups through sub-modular optimization*. 2022 (cit. on p. 11).
- [BGL+12] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. *When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings*. In: *Algorithmica* 63.4 (2012), pp. 733–762.  (cit. on p. 114).

- [BGL19] Daan Bloembergen, Davide Grossi, and Martin Lackner. *On Rational Delegations in Liquid Democracy*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2019, pp. 1796–1803 (cit. on p. 68).
- [BGP+22] Markus Brill, Paul Gözl, Dominik Peters, Ulrike Schmidt-Kraepelin, and Kai Wilker. *Approval-Based Apportionment*. In: *Mathematical Programming* (2022).  (cit. on p. 11).
- [BGP19] Gerdus Benadè, Paul Gözl, and Ariel D. Procaccia. *No Stratification Without Representation*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2019, pp. 281–314.  (cit. on p. 11).
- [BGV20] Siddhartha Banerjee, Itai Gurvich, and Alberto Vera. *Constant Regret in Online Allocation: On the Sufficiency of a Single Historical Trace*. 2020 (cit. on pp. 134, 135).
- [BH09] Hubertus Buchstein and Michael Hein. *Randomizing Europe: The Lottery as a Decision-Making Procedure for Policy Creation in the EU*. In: *Critical Policy Studies* 3.1 (2009), pp. 29–57.  (cit. on pp. 8, 113, 129).
- [BH83] K. R. W. Brewer and Muhammad Hanif. *An Introduction to Sampling with Unequal Probabilities*. Vol. 15. Lecture Notes in Statistics. Springer, 1983 (cit. on pp. 111, 129).
- [BHJ13] Hubertus Buchstein, Michael Hein, and Jakob Jünger. *Die „EU-Kommissionslotterie“: Eine Simulationsstudie*. In: *Die Versprechen der Demokratie*. 2013, pp. 190–227.  (cit. on p. 129).
- [Bin20] Reuben Binns. *On the Apparent Conflict between Individual and Group Fairness*. In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2020, pp. 514–524.  (cit. on p. 24).
- [Bir46] Garrett Birkhoff. *Three Observations on Linear Algebra*. In: *Universidad Nacional de Tucumán, Revista A* 5 (1946), pp. 147–151 (cit. on p. 114).
- [BK14] Steven J. Brams and D. Marc Kilgour. *Satisfaction Approval Voting*. In: *Voting Power and Procedures*. Springer, 2014, pp. 323–346 (cit. on p. 95).
- [BKNS14] Jan Behrens, Axel Kistner, Andreas Nitsche, and Björn Swierczek. *The Principles of LiquidFeedback*. 1st edition. Interaktive Demokratie e.V, 2014 (cit. on pp. 66, 68).
- [BKP19] Steven J. Brams, D. Marc Kilgour, and Richard F. Potthoff. *Multi-winner Approval Voting: An Apportionment Approach*. In: *Public Choice* 178.1 (2019), pp. 67–93.  (cit. on p. 95).
- [BLS18] Markus Brill, Jean-François Laslier, and Piotr Skowron. *Multi-winner Approval Rules as Apportionment Methods*. In: *Journal of Theoretical Politics* 30.3 (2018), pp. 358–382.  (cit. on pp. 95, 98, 101).
- [BM01] Anna Bogomolnaia and Hervé Moulin. *A New Solution to the Random Assignment Problem*. In: *Journal of Economic Theory* 100.2 (2001), pp. 295–328.  (cit. on p. 130).

- [BMo4] Anna Bogomolnaia and Hervé Moulin. *Random Matching under Dichotomous Preferences*. In: *Econometrica* 72.1 (2004), pp. 257–279.  (cit. on p. 18).
- [BMSo5] Anna Bogomolnaia, Hervé Moulin, and Richard Stong. *Collective Choice under Dichotomous Preferences*. In: *Journal of Economic Theory* 122.2 (2005), pp. 165–184.  (cit. on p. 102).
- [BP14] Bastian Bullwinkel and Lothar Probst. *Innerparteiliche Willensbildung und Entscheidungsprozesse durch digitale Partizipation: Ein Praxistest des Konzepts der Liquid Democracy*. In: *Zeitschrift für Parlamentsfragen* (2014), pp. 382–401.  (cit. on p. 6).
- [BP21] Florian Brandl and Dominik Peters. *Approval Voting under Dichotomous Preferences: A Catalogue of Characterizations*. 2021 (cit. on p. 93).
- [BP22] Kirk Bansak and Elisabeth Paulson. *Outcome-Driven Dynamic Refugee Assignment with Allocation Balancing*. 2022.  (cit. on p. 135).
- [BRo4] Béla Bollobás and Oliver M. Riordan. *Mathematical Results on Scale-Free Random Graphs*. In: *Handbook of Graphs and Networks*. Ed. by Stefan Bornholdt and Hans Georg Schuster. Wiley, 2004, pp. 1–34.  (cit. on p. 80).
- [Bri18] Markus Brill. *Interactive Democracy*. In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2018, pp. 1183–1187 (cit. on p. 92).
- [BS14] Terry Bouricius and David Schechter. *An Idealized Design for Government. Part 2: Executive Branch Accountability*. 2014. <https://web.archive.org/web/20180916144903/http://stwj.systemswiki.org/?p=1717> (cit. on p. 2).
- [BS18] David E. Broockman and Christopher Skovron. *Bias in Perceptions of Public Opinion among Political Elites*. In: *American Political Science Review* 112.3 (2018), pp. 542–563.  (cit. on p. 1).
- [BT18] Markus Brill and Nimrod Talmon. *Pairwise Liquid Democracy*. In: *IJCAI*. 2018, pp. 137–143 (cit. on p. 68).
- [BT96] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996 (cit. on p. 14).
- [BYo1] Michel L. Balinski and H. Peyton Young. *Fair Representation: Meeting the Ideal of One Man, One Vote*. 2nd edition. Brookings Institution Press, 2001 (cit. on pp. 7, 93, 95, 102, 103, 111, 112, 116, 118).
- [BY75] Michel L. Balinski and H. Peyton Young. *The Quota Method of Apportionment*. In: *The American Mathematical Monthly* 82.7 (1975), pp. 701–730.  (cit. on pp. 104, 114).
- [BY79] Michel L. Balinski and H. Peyton Young. *Quotatone Apportionment Methods*. In: *Mathematics of Operations Research* 4.1 (1979), pp. 31–38.  (cit. on pp. 120, 123).

- [BZ16] Christian Blum and Christina Isabel Zuber. *Liquid Democracy: Potentials, Problems, and Perspectives*. In: *Journal of Political Philosophy* 24.2 (2016), pp. 162–182.  (cit. on pp. 6, 67).
- [CD20] Hun Chung and John Duggan. *A Formal Theory of Democratic Deliberation*. In: *American Political Science Review* 114.1 (2020), pp. 14–35.  (cit. on pp. 53, 63, 159).
- [CG17] Zoé Christoff and Davide Grossi. *Binary Voting with Delegable Proxy: An Analysis of Liquid Democracy*. In: *Electronic Proceedings in Theoretical Computer Science* 251 (2017), pp. 134–150.  (cit. on p. 68).
- [CG98] Jaime Carbonell and Jade Goldstein. *The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries*. In: *Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 1998, pp. 335–336.  (cit. on p. 53).
- [Cit20] Citizens’ Panel on COVID-19. *Final report of the citizens’ panel on COVID-19*. 2020. <https://joinofbyfor.org/wp-content/uploads/2020/11/Final-Report-of-the-Citizens-Panel-on-COVID-19.pdf>  (cit. on pp. 10, 13).
- [CJMW19] Yu Cheng, Zhihao Jiang, Kamesh Munagala, and Kangning Wang. *Group Fairness in Committee Selection*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2019, pp. 263–279.  (cit. on pp. 99, 114).
- [CKL+07] Jiangzhuo Chen, Robert D. Kleinberg, László Lovász, Rajmohan Rajaraman, Ravi Sundaram, and Adrian Vetta. *(Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows*. In: *Journal of the ACM (JACM)* 54.4 (2007), p. 16.  (cit. on pp. 66, 69, 72–74, 86).
- [Cli20] Climate Assembly UK. 2020. <https://www.climateassembly.uk/report/read/final-report.pdf>  (cit. on pp. 29, 41).
- [CM19] Ioannis Caragiannis and Evi Micha. *A Contribution to the Critique of Liquid Democracy*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 116–122 (cit. on p. 68).
- [CM99] Lyn Carson and Brian Martin. *Random Selection in Politics*. Praeger, 1999 (cit. on pp. 13, 14, 34, 158, 184–187).
- [CMS19] Rosa Camps, Xavier Mora, and Laia Saumell. *The Method of Eneström and Phragmén for Parliamentary Elections by Means of Approval Voting*. 2019.  (cit. on p. 95).
- [Cou19] Dimitri Courant. *Sortition and Democratic Principles: A Comparative Analysis*. In: *Legislature by Lot: Transformative Designs for Deliberative Governance*. The Real Utopias Project. Verso, 2019 (cit. on p. 184).
- [CRSo6] Jiangzhuo Chen, Rajmohan Rajaraman, and Ravi Sundaram. *Meet and Merge: Approximation Algorithms for Confluent Flows*. In: *Journal of Computer and System Sciences* 72.3 (2006), pp. 468–489.  (cit. on p. 72).

- [CVZ09] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. *Dependent Randomized Rounding for Matroid Polytopes and Applications*. 2009.  (cit. on pp. 40, 42).
- [CVZ10] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. *Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures*. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2010, pp. 575–584.  (cit. on p. 27).
- [Dah90] Robert Alan Dahl. *After the Revolution? Authority in a Good Society*. Yale University Press, 1990 (cit. on p. 13).
- [Dal50] Tore Dalenius. *The Problem of Optimum Stratification*. In: *Scandinavian Actuarial Journal* 1950.3-4 (1950), pp. 203–213.  (cit. on p. 37).
- [DALU21] Virginie Do, Jamal Atif, Jérôme Lang, and Nicolas Usunier. *Online Selection of Diverse Committees*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 154–160.  (cit. on p. 15).
- [Dam14] Anna Piil Damm. *Neighborhood Quality and Labor Market Outcomes: Evidence from Quasi-Random Neighborhood Assignment of Immigrants*. In: *Journal of Urban Economics* 79 (2014), pp. 139–166.  (cit. on p. 133).
- [DBC+19] John S. Dryzek, André Bächtiger, Simone Chambers, Joshua Cohen, James N. Druckman, Andrea Felicetti, James S. Fishkin, David M. Farrell, Archon Fung, Amy Gutmann, Hélène Landemore, Jane Mansbridge, Sofie Marien, Michael A. Neblo, Simon Niemeyer, Maija Setälä, Rune Slothuus, Jane Suiter, Dennis Thompson, and Mark E. Warren. *The Crisis of Democracy and the Science of Deliberation*. In: *Science* 363.6432 (2019), pp. 1144–1146.  (cit. on pp. 1, 2).
- [DD10] Gil Delannoi and Oliver Dowlen, eds. *Sortition: Theory and Practice*. Imprint Academic, 2010 (cit. on p. 13).
- [Dem19] Democracy R&D. *About the network*. 2019. <https://democracyrd.org/about/>  (cit. on p. 13).
- [DH09] Nikhil R. Devanur and Thomas P. Hayes. *The Adwords Problem: Online Keyword Matching with Budgeted Bidders under Random Permutations*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2009, pp. 71–78.  (cit. on p. 135).
- [DL03] John S. Dryzek and Christian List. *Social Choice Theory and Deliberative Democracy: A Reconciliation*. In: *British Journal of Political Science* 33.1 (2003), pp. 1–28 (cit. on p. 159).
- [Dow08] Oliver Dowlen. *The Political Potential of Sortition: A Study of the Random Selection of Citizens for Public Office*. Imprint Academic, 2008 (cit. on p. 187).
- [Dow09] Oliver Dowlen. *Sorting Out Sortition: A Perspective on the Random Selection of Political Officers*. In: *Political Studies* 57.2 (2009), pp. 298–315.  (cit. on p. 186).

- [Dro81] H. R. Droop. *On Methods of Electing Representatives*. In: *Journal of the Statistical Society of London* 44.2 (1881), pp. 141–202.  (cit. on p. 101).
- [Dud15] Conal Duddy. *Fair Sharing under Dichotomous Preferences*. In: *Mathematical Social Sciences* 73 (2015), pp. 1–5.  (cit. on p. 102).
- [EE19] Stephen Elstub and Oliver Escobar. *Handbook of Democratic Innovation and Governance*. Edward Elgar Publishing, 2019 (cit. on p. 1).
- [EFSS17] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. *Properties of Multiwinner Voting Rules*. In: *Social Choice and Welfare* 48.3 (2017), pp. 599–632.  (cit. on p. 97).
- [EGP19] Bruno Escoffier, Hugo Gilbert, and Adèle Pass-Lanneau. *The Convergence of Iterative Delegations in Liquid Democracy in a Social Network*. In: *Algorithmic Game Theory*. Ed. by Dimitris Fotakis and Evangelos Markakis. Springer, 2019, pp. 284–297.  (cit. on p. 68).
- [El-19] Sherif El-Helaly. *The Mathematics of Voting and Apportionment: An Introduction*. Compact Textbooks in Mathematics. Springer, 2019 (cit. on p. 116).
- [End10] Ulle Endriss. *Lecture Notes on Fair Division*. 2010 (cit. on p. 18).
- [End13] Ulle Endriss. *Reduction of Economic Inequality in Combinatorial Domains*. In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2013 (cit. on p. 18).
- [Eng89] Fredrik Engelstad. *The Assignment of Political Office by Lot*. In: *Social Science Information* 28.1 (1989), pp. 23–50.  (cit. on pp. 14, 184–186).
- [Faind] Fair Outcomes. *Fair Proposals*. n.d. <https://www.fairproposals.com/>  (cit. on p. 15).
- [FB19] Daniel Freund and Siddhartha Banerjee. *Good Prophets Know When the End Is Near*. Social Science Research Network, 2019.  (cit. on p. 134).
- [FGG+18] James Fishkin, Nikhil Garg, Lodewijk Gelauff, Ashish Goel, Kamesh Munagala, Sukolsak Sakshuwong, Alice Siu, and Sravya Yandamuri. *Deliberative Democracy with the Online Deliberation Platform*. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 2018 (cit. on p. 53).
- [FGG+21] Bailey Flanigan, Paul Gözl, Anupam Gupta, Brett Hennig, and Ariel D. Procaccia. *Fair Algorithms for Selecting Citizens' Assemblies*. In: *Nature* 596.7873 (2021), pp. 548–552.  (cit. on pp. 11, 157).
- [FGGP20] Bailey Flanigan, Paul Gözl, Anupam Gupta, and Ariel D. Procaccia. *Neutralizing Self-Selection Bias in Sampling for Sortition*. In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cit. on p. 11).

- [FGM16] Brandon Fain, Ashish Goel, and Kamesh Munagala. *The Core of the Participatory Budgeting Problem*. In: *Proceedings of the Conference on Web and Internet Economics (WINE)*. 2016, pp. 384–399.  (cit. on p. 101).
- [FGMS17] Brandon Fain, Ashish Goel, Kamesh Munagala, and Sukolsak Sakshuwong. *Sequential Deliberation for Social Choice*. In: *Proceedings of the Conference on Web and Internet Economics (WINE)*. Ed. by Nikhil R. Devanur and Pinyan Lu. 2017, pp. 177–190.  (cit. on pp. 53, 63, 159).
- [FHTZ20] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. *Edge-Weighted Online Bipartite Matching*. In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 412–423.  (cit. on p. 134).
- [FW80] Steven Fortune, John Hopcroft, and James Wyllie. *The Directed Subgraph Homeomorphism Problem*. In: *Theoretical Computer Science* 10.2 (1980), pp. 111–121.  (cit. on p. 73).
- [Fin16] Hannah Fingerhut. *Already-low voter satisfaction with choice of candidates falls even further*. Pew Research Center. 2016. <https://www.pewresearch.org/fact-tank/2016/09/12/already-low-voter-satisfaction-with-choice-of-candidates-falls-even-further/>  (cit. on p. 1).
- [Fin71] D. J. Finney. *Probit Analysis*. 3rd edition. Cambridge University Press, 1971 (cit. on p. 28).
- [Fis09] James S. Fishkin. *When the People Speak: Deliberative Democracy and Public Consultation*. Oxford University Press, 2009 (cit. on pp. 2, 14, 185).
- [Fis18] James S. Fishkin. *Democracy When the People Are Thinking: Revitalizing Our Politics Through Public Deliberation*. Oxford University Press, 2018 (cit. on pp. 13, 14, 184, 185).
- [Fis78] Peter C. Fishburn. *Axioms for Approval Voting: Direct Proof*. In: *Journal of Economic Theory* 19.1 (1978), pp. 180–185.  (cit. on p. 93).
- [Fis79] Peter C. Fishburn. *Symmetric and Consistent Aggregation with Dichotomous Voting*. In: *Aggregation and Revelation of Preferences*. Ed. by Jean-Jacques Laffont. North-Holland, 1979, pp. 201–218 (cit. on p. 93).
- [FKP21] Bailey Flanigan, Gregory Kehne, and Ariel D. Procaccia. *Fair Sortition Made Transparent*. In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 25720–25731 (cit. on pp. 15, 21).
- [FL05] James S. Fishkin and Robert C. Luskin. *Experimenting with a Democratic Ideal: Deliberative Polling and Public Opinion*. In: *Acta politica* 40.3 (2005), pp. 284–298.  (cit. on p. 13).
- [FMS18] Brandon Fain, Kamesh Munagala, and Nisarg Shah. *Fair Allocation of Indivisible Public Goods*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2018, pp. 575–592.  (cit. on p. 99).

- [FOSY78] Norman Frohlich, Joe A. Oppenheimer, Jeffrey Smith, and Oran R. Young. *A Test of Downsian Voter Rationality: 1964 Presidential Voting*. In: *American Political Science Review* 72.1 (1978), pp. 178–197.  (cit. on p. 160).
- [GB20] Ernesto Ganuza and Gianpaolo Baiocchi. *The Power of Ambiguity: How Participatory Budgeting Travels the Globe*. In: *Journal of Deliberative Democracy* 8.2 (2020).  (cit. on p. 1).
- [GHG+21] Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp M. Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelman, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, and Yuji Shinano. *MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library*. In: *Mathematical Programming Computation* (2021).  (cit. on p. 186).
- [GKMP18] Paul Gözl, Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia. *The Fluid Mechanics of Liquid Democracy*. In: *Proceedings of the Conference on Web and Internet Economics (WINE)*. 2018 (cit. on p. 11).
- [GKP19] Paul Gözl, Anson Kahng, and Ariel D. Procaccia. *Paradoxes in Fair Machine Learning*. In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2019 (cit. on pp. 11, 24).
- [GKPS06] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. *Dependent Rounding and Its Applications to Approximation Algorithms*. In: *Journal of the ACM (JACM)* 53.3 (2006), pp. 324–360.  (cit. on pp. 8, 112–114, 120, 121).
- [GL16] Ashish Goel and David T. Lee. *Towards Large-Scale Deliberative Decision-Making: Small Groups and the Importance of Triads*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2016, pp. 287–303.  (cit. on pp. 53, 159).
- [GN14] Ani Guerdjikova and Klaus Nehring. *Weighing Experts, Weighing Sources: The Diversity Value*. 2014 (cit. on p. 103).
- [GP14] Jonathan Goldman and Ariel D. Procaccia. *Spliddit: Unleashing Fair Division Algorithms*. In: *ACM SIGecom Exchanges* 13.2 (2014), pp. 41–46.  (cit. on pp. 15, 130).
- [GP19] Paul Gözl and Ariel D. Procaccia. *Migration as Submodular Optimization*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2019.  (cit. on p. 11).
- [GPP22] Paul Gözl, Dominik Peters, and Ariel D. Procaccia. *In This Apportionment Lottery, the House Always Wins*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2022.  (cit. on p. 11).
- [GR20] Paul Gözl and Gili Rusak. *Panelot*. 2020. <http://www.panelot.org/> (cit. on pp. 10, 18, 60).
- [Gre15] James Green-Armytage. *Direct Voting and Proxy Voting*. In: *Constitutional Political Economy* 26.2 (2015), pp. 190–220.  (cit. on p. 67).

- [Gri04] Geoffrey Grimmett. *Stochastic Apportionment*. In: *The American Mathematical Monthly* 111.4 (2004), pp. 299–307.  (cit. on pp. 7, 111, 113, 115, 119).
- [GS07] Pranava R. Goundan and Andreas S. Schulz. *Revisiting the Greedy Approach to Submodular Set Function Maximization*. In: *Optimization Online* 1984 (2007), pp. 1–25 (cit. on p. 58).
- [GW18] John Gastil and Erik Olin Wright. *Legislature by Lot: Envisioning Sortition within a Bicameral System*. In: *Politics & Society* 46.3 (2018), pp. 303–330.  (cit. on p. 2).
- [Hal35] P. Hall. *On Representatives of Subsets*. In: *Journal of the London Mathematical Society* s1-10.1 (1935), pp. 26–30.  (cit. on p. 124).
- [HC20] Lily Hu and Yiling Chen. *Fair Classification and Social Welfare*. In: *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAcT)*. 2020, pp. 535–545.  (cit. on p. 24).
- [HE91] D. Holt and D. Elliot. *Methods of Weighting for Unit Non-Response*. In: *The Statistician* 40.3 (1991), p. 333.  (cit. on p. 24).
- [HG21] Brett Hennig and Paul Gözl. *StratifySelect*. 2021. <https://github.com/sortitionfoundation/stratification-app>  (cit. on pp. 9, 18).
- [HJ16] John Haslegrave and Jonathan Jordan. *Preferential Attachment with Choice*. In: *Random Structures & Algorithms* 48.4 (2016), pp. 751–766.  (cit. on p. 81).
- [HMM10] Andrew J. Healy, Neil Malhotra, and Cecilia Hyunjung Mo. *Irrelevant Events Affect Voters' Evaluations of Government Performance*. In: *Proceedings of the National Academy of Sciences* 107.29 (2010), pp. 12804–12809.  (cit. on p. 1).
- [HP98] Dorit S. Hochbaum and Anu Pathria. *Analysis of the Greedy Approach in Problems of Maximum K-Coverage*. In: *Naval Research Logistics* 45.6 (1998), pp. 615–627.  (cit. on p. 55).
- [HT52] D. G. Horvitz and D. J. Thompson. *A Generalization of Sampling Without Replacement From a Finite Universe*. In: *Journal of the American Statistical Association* 47.260 (1952), p. 663.  (cit. on p. 37).
- [HV12] Chien-Ju Ho and Jennifer Vaughan. *Online Task Assignment in Crowdsourcing Markets*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 1. 2012, pp. 45–51.  (cit. on p. 135).
- [HZ79] Aanund Hylland and Richard Zeckhauser. *The Efficient Allocation of Individuals to Positions*. In: *Journal of Political Economy* 87.2 (1979), pp. 293–314.  (cit. on p. 130).
- [IB21] Jonas Israel and Markus Brill. *Dynamic Proportional Rankings*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 261–267.  (cit. on p. 97).
- [Iri19a] Irish Citizens' Assembly Project. *Irish citizens' assembly (2019–2020)*. 2019. <http://www.citizenassembly.ie/work/>  (cit. on p. 13).
- [Iri19b] Irish Citizens' Assembly Project. *Procurement and Costs*. 2019. <https://web.archive.org/web/20190331155441/https://www.citizenassembly.ie/en/about-the-citizens-assembly/background/procurement-and-costs/> (cit. on p. 46).

- [Jac17] Vincent Jacquet. *Explaining Non-Participation in Deliberative Mini-Publics*. In: *European Journal of Political Research* 56.3 (2017), pp. 640–659. [🔗](#) (cit. on pp. 4, 5).
- [Jac19] Vincent Jacquet. *The Role and the Future of Deliberative Mini-Publics: A Citizen Perspective*. In: *Political Studies* 67.3 (2019), pp. 639–657. [🔗](#) (cit. on p. 13).
- [Jan18a] Svante Janson. *Phragmén's and Thiele's Election Methods*. 2018. [🔗](#) (cit. on p. 98).
- [Jan18b] Svante Janson. *Thresholds Quantifying Proportionality Criteria for Election Methods*. 2018. [🔗](#) (cit. on p. 101).
- [JKK21] Ramesh Johari, Vijay Kamble, and Yash Kanoria. *Matching While Learning*. In: *Operations Research* 69.2 (2021), pp. 655–681. [🔗](#) (cit. on p. 135).
- [JMW20] Zhihao Jiang, Kamesh Munagala, and Kangning Wang. *Approximately Stable Committee Selection*. In: *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*. 2020, pp. 463–472. [🔗](#) (cit. on p. 99).
- [JÖ19] S. Janson and A. Öberg. *A Piecewise Contractive Dynamical System and Phragmén's Election Method*. In: *Bulletin de la Société Mathématique de France* 147.3 (2019), pp. 395–441. [🔗](#) (cit. on p. 95).
- [Jon15] Irving Jones. *Home Away From Home*. In: *State Magazine* (2015), pp. 17–20 (cit. on p. 136).
- [KB21] Filip Kostelka and André Blais. *The Generational and Institutional Sources of the Global Decline in Voter Turnout*. In: *World Politics* 73.4 (2021), pp. 629–667. [🔗](#) (cit. on p. 160).
- [Kha17] Parag Khanna. *Technocracy in America: Rise of the Info-State*. 2017 (cit. on p. 161).
- [Kin93] J. F. C Kingman. *Poisson Processes*. Clarendon Press : Oxford University Press, 1993 (cit. on pp. 117, 118).
- [KKH+15] Christoph Kling, Jérôme Kunegis, Heinrich Hartmann, Markus Strohmaier, and Steffen Staab. *Voting Behaviour and Power in Online Democracy: A Study of LiquidFeedback in Germany's Pirate Party*. In: *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)* 9.1 (2015), pp. 208–217 (cit. on pp. 6, 65, 67, 75–77).
- [KM12] D. Marc Kilgour and Erica Marshall. *Approval Balloting for Fixed-Size Committees*. In: *Electoral Systems*. Ed. by Dan S. Felsenthal and Moshé Machover. Studies in Choice and Welfare. Springer, 2012, pp. 305–326. [🔗](#) (cit. on pp. 93, 95, 98).
- [KMP18] Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia. *Liquid Democracy: An Algorithmic Perspective*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2018 (cit. on pp. 65, 67, 77, 91).
- [KMPS09] VS Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. *A Unified Approach to Scheduling on Unrelated Parallel Machines*. In: *Journal of the ACM (JACM)* 56.5 (2009), pp. 1–31. [🔗](#) (cit. on p. 114).

- [KN79] Mamoru Kaneko and Kenjiro Nakamura. *The Nash Social Welfare Function*. In: *Econometrica* 47.2 (1979), p. 423.  (cit. on p. 101).
- [KNT10] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. *Structure and Evolution of Online Social Networks*. In: *Link Mining: Models, Algorithms, and Applications*. Ed. by Philip S. Yu, Jiawei Han, and Christos Faloutsos. Springer, 2010, pp. 337–357.  (cit. on p. 76).
- [KPS18] David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. *Leximin Allocations in the Real World*. In: *ACM Transactions on Economics and Computation (TEAC)* 6.3-4 (2018), pp. 1–24.  (cit. on pp. 18, 131).
- [KPV13] Michael Kapralov, Ian Post, and Jan Vondrák. *Online Submodular Welfare Maximization: Greedy Is Optimal*. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2013, pp. 1216–1225.  (cit. on p. 57).
- [KRTV13] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. *An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions*. In: *European Symposium on Algorithms*. 2013, pp. 589–600 (cit. on p. 134).
- [KSG08] Andreas Krause, Ajit Singh, and Carlos Guestrin. *Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies*. In: *Journal of Machine Learning Research* 9.2 (2008).  (cit. on p. 53).
- [KVV90] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. *An Optimal Algorithm for On-Line Bipartite Matching*. In: *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*. 1990, pp. 352–358 (cit. on p. 134).
- [Lan20] Hélène Landemore. *Open Democracy: Reinventing Popular Rule for the Twenty-First Century*. Princeton University Press, 2020 (cit. on pp. 1, 2).
- [LB11] Hui Lin and Jeff Bilmes. *A Class of Submodular Functions for Document Summarization*. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 510–520 (cit. on p. 53).
- [Leio4] Ethan J. Leib. *Deliberative Democracy in America: A Proposal for a Popular Branch of Government*. Penn State Press, 2004 (cit. on p. 5).
- [Leo83] Herman B. Leonard. *Elicitation of Honest Preferences for the Assignment of Individuals to Positions*. In: *Journal of Political Economy* 91.3 (1983), pp. 461–479.  (cit. on p. 140).
- [Ley19] Dominique Leydet. *Which Conception of Political Equality Do Deliberative Mini-Publics Promote?* In: *European Journal of Political Theory* 18.3 (2019), pp. 349–370.  (cit. on p. 187).
- [LI96] Tony Lancaster and Guido Imbens. *Case-Control Studies with Contaminated Controls*. In: *Journal of Econometrics* 71.1-2 (1996), pp. 145–160.  (cit. on pp. 4, 24, 29).

- [LMCS15] Frédéric Lardeux, Eric Monfroy, Broderick Crawford, and Ricardo Soto. *Set Constraint Model and Automated Encoding into SAT: Application to the Social Golfer Problem*. In: *Annals of Operations Research* 235.1 (2015), pp. 423–452.  (cit. on p. 54).
- [LN14] Jan E. Leighley and Jonathan Nagler. *Who Votes Now? Demographics, Issues, Inequality and Turnout in the United States*. Princeton University Press, 2014 (cit. on p. 160).
- [LP10] Julien Lesca and Patrice Perny. *LP Solvable Models for Multiagent Fair Allocation Problems*. In: *ECAI*. Vol. 2010. 2010, pp. 393–398 (cit. on p. 18).
- [LP12] Jeffrey R. Lax and Justin H. Phillips. *The Democratic Deficit in the States*. In: *American Journal of Political Science* 56.1 (2012), pp. 148–166.  (cit. on p. 1).
- [LS10] Jean-François Laslier and M. Remzi Sanver, eds. *Handbook on Approval Voting*. Studies in Choice and Welfare. Springer, 2010 (cit. on p. 93).
- [LS22] Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. Springer, 2022 (cit. on pp. 93, 95, 98, 101, 102, 107).
- [Mal15] Anthoula Malkopoulou. *The Paradox of Democratic Selection: Is Sortition Better than Voting?* In: *Parliamentarism and Democratic Theory: Historical and Contemporary Perspectives*. Verlag Barbara Budrich, 2015, p. 229.  (cit. on p. 5).
- [Man10] Jane Mansbridge. *Deliberative Polling as the Gold Standard*. In: *The good society* 19.1 (2010), pp. 55–62.  (cit. on p. 2).
- [Man19] Robert A. Manduca. *Mechanism Design for Social Good*. 2019.  (cit. on p. 8).
- [Man99] Jane Mansbridge. *Should Blacks Represent Blacks and Women Represent Women? A Contingent “Yes”*. In: *The Journal of politics* 61.3 (1999), pp. 628–657.  (cit. on p. 185).
- [MAS17] MASS LBP. *How to Run a Civic Lottery: Designing Fair Selection Mechanisms for Deliberative Public Processes*. 2017 (cit. on p. 14).
- [MGS12] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. *Online Stochastic Matching: Online Actions Based on Offline Statistics*. In: *Mathematics of Operations Research* 37.4 (2012), pp. 559–573.  (cit. on p. 134).
- [MH11] Sofie Marien and Marc Hooghe. *Does Political Trust Matter? An Empirical Investigation into the Relation between Political Trust and Support for Law Compliance: Does Political Trust Matter?* In: *European Journal of Political Research* 50.2 (2011), pp. 267–291.  (cit. on p. 1).
- [MHH19] Linna Martén, Jens Hainmueller, and Dominik Hangartner. *Ethnic Networks Can Foster the Economic Integration of Refugees*. In: *Proceedings of the National Academy of Sciences* 116.33 (2019), pp. 16280–16285.  (cit. on p. 133).

- [Mil69] James C. Miller. *A Program for Direct and Proxy Voting in the Legislative Process*. In: *Public Choice* 7–7.1 (1969), pp. 107–113.  (cit. on p. 65).
- [MM18] Spencer McKay and Peter MacLeod. *MASS LBP and Long-Form Deliberation in Canada*. In: *Democratic Theory* 5.2 (2018), pp. 108–113.  (cit. on p. 13).
- [MO15] X. Mora and M. Oliver. *Eleccions Mitjançant El Vot d'aprovació. El Mètode de Phragmén i Algunes Variants*. In: *Butlletí de la Societat Catalana de Matemàtiques* 30.1 (2015), pp. 57–101 (cit. on p. 95).
- [Mou04] Hervé Moulin. *Fair Division and Collective Welfare*. MIT press, 2004 (cit. on pp. 14, 18).
- [MP15] Yury Malyskin and Elliot Paquette. *The Power of Choice over Preferential Attachment*. In: *Latin American Journal of Probability and Mathematical Statistics* 12.2 (2015), pp. 903–915 (cit. on pp. 77, 81).
- [MST21] Reshef Meir, Fedor Sandomirskiy, and Moshe Tennenholtz. *Representative Committees of Peers*. In: *Journal of Artificial Intelligence Research* 71 (2021), pp. 401–429.  (cit. on p. 15).
- [MSVV07] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. *Adwords and Generalized Online Matching*. In: *Journal of the ACM (JACM)* 54.5 (2007), 22–es.  (cit. on p. 135).
- [MTW72] Dennis C. Mueller, Robert D. Tollison, and Thomas D. Willett. *Representative Democracy via Random Selection*. In: *Public Choice* 12.1 (1972), pp. 57–68.  (cit. on p. 35).
- [Nas50] John F. Nash Jr. *The Bargaining Problem*. In: *Econometrica: Journal of the Econometric Society* (1950), pp. 155–162.  (cit. on p. 101).
- [NEK+10] Michael A. Neblo, Kevin M. Esterling, Ryan P. Kennedy, David M. J. Lazer, and Anand E. Sokhey. *Who Wants to Deliberate—and Why?* In: *American Political Science Review* (2010), pp. 566–583.  (cit. on p. 13).
- [Neu22] Thirty-sixth Conference on Neural Information Processing Systems. *NeurIPS Paper Checklist Guidelines*. 2022. <https://neurips.cc/public/guides/PaperChecklist>  (cit. on p. 161).
- [New01] M. E. J. Newman. *Clustering and Preferential Attachment in Growing Networks*. In: *Physical Review E* 64.2 (2001), p. 025102.  (cit. on p. 76).
- [NR01] Noam Nisan and Amir Ronen. *Algorithmic Mechanism Design*. In: *Games and Economic Behavior* 35.1–2 (2001), pp. 166–196.  (cit. on p. 112).
- [NR19] Christoph Niessen and Min Reuchamps. *Designing a Permanent Deliberative Citizens' Assembly*. In: *Centre for Deliberative Democracy and Global Governance Working Paper Series* 6 (2019) (cit. on pp. 2, 9, 13).
- [NSD16] Norwegian Centre for Research Data. *European Social Survey Round 8 Data*. Data file edition 2.1. 2016 (cit. on p. 29).

- [nU18] newDemocracy Foundation and United Nations Democracy Fund. *Enabling National Initiatives to Take Democracy Beyond Elections*. 2018 (cit. on p. 63).
- [NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. *An Analysis of Approximations for Maximizing Submodular Set Functions—I*. In: *Mathematical programming* 14.1 (1978), pp. 265–294.  (cit. on pp. 53, 57, 58, 61).
- [OEC20] Organisation for Economic Co-operation and Development. *Innovative Citizen Participation and New Democratic Institutions: Catching the Deliberative Wave*. OECD, 2020 (cit. on pp. 1, 2, 13, 21).
- [OEC21] Organisation for Economic Co-operation and Development. *Eight Ways to Institutionalise Deliberative Democracy*. OECD Public Governance Policy Papers 12. 2021.  (cit. on pp. 2, 13).
- [OEC22] Organisation for Economic Co-operation and Development. *Building Trust to Reinforce Democracy: Main Findings from the 2021 OECD Survey on Drivers of Trust in Public Institutions*. Building Trust in Public Institutions. OECD, 2022 (cit. on pp. 1, 160).
- [Ofb20] Of By For \*. *Democratic lottery — the citizens’ panel on COVID-19*. 2020. <https://joinofbyfor.org/panel/> video in section “selected without parties or campaigns” (cit. on pp. 10, 13).
- [Par11] Joel Matthew Parker. *Randomness and Legitimacy in Selecting Democratic Representatives*. University of Texas at Austin, 2011 (cit. on pp. 14, 184, 185, 187).
- [Par21] Various Authors. *Participedia*. 2021. [https://participedia.net/search?selectedCategory=case&recruitment\\_method=random,stratified](https://participedia.net/search?selectedCategory=case&recruitment_method=random,stratified)  (cit. on p. 13).
- [Pau20] Alois Paulin. *Ten Years of Liquid Democracy Research: An Overview*. In: *Central and Eastern European eDem and eGov Days* 338 (2020), pp. 455–466.  (cit. on p. 2).
- [Pek19] Simon Pek. *Rekindling Union Democracy Through the Use of Sortition*. In: *Journal of Business Ethics* 155.4 (2019), pp. 1033–1051.  (cit. on p. 2).
- [Pew14] Pew Research Center. *Beyond Red vs. Blue: The Political Typology*. 2014 (cit. on p. 35).
- [Pit67] Hanna F. Pitkin. *The Concept of Representation*. Vol. 75. Univ of California Press, 1967 (cit. on p. 93).
- [PKC20] Simon Pek, Jeffrey Kennedy, and Adam Cronkright. *Democracy Transformed: Perceived Legitimacy of the Institutional Shift from Election to Random Selection of Representatives*. In: *Journal of Deliberative Democracy* 14.1 (2020).  (cit. on p. 2).
- [PP15] Juan Perote-Peña and Ashley Piggins. *A Model of Deliberative and Aggregative Democracy*. In: *Economics & Philosophy* 31.1 (2015), pp. 93–121.  (cit. on pp. 53, 159).

- [PPS21] Dominik Peters, Grzegorz Pierczyński, and Piotr Skowron. *Proportional Participatory Budgeting with Additive Utilities*. In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 12726–12737 (cit. on p. 160).
- [PS20] Dominik Peters and Piotr Skowron. *Proportionality and the Limits of Welfarism*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2020, pp. 793–794 (cit. on pp. 99, 101).
- [PS97] Alessandro Panconesi and Aravind Srinivasan. *Randomized Distributed Edge Coloring via an Extension of the Chernoff–Hoeffding Bounds*. In: *SIAM Journal on Computing* 26.2 (1997), pp. 350–368.  (cit. on pp. 114, 121, 130).
- [PST20] Sebastian Pokutta, Mohit Singh, and Alfredo Torricco. *On the Unreasonable Effectiveness of the Greedy Algorithm: Greedy Adapts to Sharpness*. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2020, pp. 7772–7782 (cit. on p. 58).
- [Puk14] Friedrich Pukelsheim. *Proportional Representation: Apportionment Methods and Their Applications*. Springer, 2014 (cit. on pp. 93, 95, 102).
- [Rae89] Douglas W. Rae. *Equalities*. Harvard University Press, 1989 (cit. on p. 185).
- [RBPW09] Gilian Raab, Kathy Buckner, Susan Purdon, and Iona Waterston. *Adjusting for Non-Response by Weighting*. In: *Practical Exemplars for Survey Analysis*. 2009 (cit. on p. 24).
- [RO68] William H. Riker and Peter C. Ordeshook. *A Theory of the Calculus of Voting*. In: *American Political Science Review* 62.1 (1968), pp. 25–42.  (cit. on p. 160).
- [RSÜ05] Alvin E. Roth, Tayfun Sönmez, and M. U. Ünver. *Pairwise Kidney Exchange*. In: *Journal of Economic Theory* 125.2 (2005), pp. 151–188.  (cit. on p. 18).
- [RU10] E. Arthur Robinson and Daniel Ullman. *A Mathematical Look at Politics*. CRC Press, 2010 (cit. on pp. 115, 116).
- [SBJB20] Daniel Steel, Naseeb Bolduc, Kristina Jenei, and Michael Burgess. *Rethinking Representation and Diversity in Deliberative Minipublics*. In: *Journal of Deliberative Democracy* 16.1 (2020), pp. 46–57.  (cit. on p. 187).
- [Sco21] Scotland’s Climate Assembly. *Recommendations for Action*. 2021. [https://webarchive.nrsotland.gov.uk/web/20220321142709/https://www.climateassembly.scot/sites/default/files/2021-09/620640\\_SCT0521502140-001\\_Scotland%E2%80%99s%20Climate%20Assembly\\_Final%20Report%20Goals\\_WEB%20ONLY%20VERSION.pdf](https://webarchive.nrsotland.gov.uk/web/20220321142709/https://www.climateassembly.scot/sites/default/files/2021-09/620640_SCT0521502140-001_Scotland%E2%80%99s%20Climate%20Assembly_Final%20Report%20Goals_WEB%20ONLY%20VERSION.pdf) (cit. on p. 9).
- [SDE17] Sebastian Schneckenburger, Britta Dorn, and Ulle Endriss. *The Atkinson Inequality Index in Multiagent Resource Allocation*. In: *AAMAS*. 2017, pp. 272–280 (cit. on p. 18).
- [SDFH18] Tom W. Smith, Michael Davern, Jeremy Freese, and Michael Hout. *General Social Surveys, 2014 and 2016*. 2018 (cit. on p. 46).

- [SEL+17] Luis Sánchez-Fernández, Edith Elkind, Martin Lackner, Norberto Fernández, Jesús A. Fisteus, Pablo Basanta Val, and Piotr Skowron. *Proportional Justified Representation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2017 (cit. on p. 108).
- [SF19] Luis Sánchez-Fernández and Jesús A. Fisteus. *Monotonicity Axioms in Approval-Based Multi-Winner Voting Rules*. In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2019, pp. 485–493 (cit. on p. 109).
- [SFCS22] Sean R. Sinclair, Felipe Frujeri, Ching-An Cheng, and Adith Swaminathan. *Hindsight Learning in MDPs with Exogenous Inputs*. 2022 (cit. on pp. 134, 135).
- [SFL16] Piotr Skowron, Piotr Faliszewski, and Jerome Lang. *Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 241 (2016), pp. 191–216 (cit. on p. 109).
- [SG19] Pietro Speroni di Fenizio and Daniele A. Gewurz. *The Space of All Proportional Voting Systems and the Most Majoritarian among Them*. In: *Social Choice and Welfare* 52.4 (2019), pp. 663–683.  (cit. on pp. 95, 103).
- [SLB+17] Piotr Skowron, Martin Lackner, Markus Brill, Dominik Peters, and Edith Elkind. *Proportional Rankings*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2017.  (cit. on p. 97).
- [Smio9] Graham Smith. *Democratic Innovations: Designing Institutions for Citizen Participation*. Theories of Institutional Design. Cambridge University Press, 2009 (cit. on pp. 1, 18, 184, 187).
- [SS18a] Barna Saha and Aravind Srinivasan. *A New Approximation Technique for Resource-Allocation Problems*. In: *Random Structures & Algorithms* 52.4 (2018), pp. 680–715.  (cit. on p. 114).
- [SS18b] Graham Smith and Maija Setälä. *Mini-Publics and Deliberative Democracy*. In: *The Oxford Handbook of Deliberative Democracy*. Ed. by Andre Bächtiger, John S. Dryzek, Jane Mansbridge, and Mark Warren. Oxford University Press, 2018, pp. 299–314.  (cit. on p. 2).
- [SSV22] Daniel Schmand, Marc Schröder, and Laura Vargas Koch. *A Greedy Algorithm for the Social Golfer and the Oberwolfach Problem*. In: *European Journal of Operational Research* 300.1 (2022), pp. 310–319.  (cit. on p. 54).
- [Sti79] Jonathan W. Still. *A Class of New Methods for Congressional Apportionment*. In: *SIAM Journal on Applied Mathematics* 37.2 (1979), pp. 401–418.  (cit. on pp. 114, 120, 123).
- [Stoo8] Peter Stone. *Voting, Lotteries, and Justice*. In: *Polity* 40.2 (2008), pp. 246–253.  (cit. on p. 5).
- [Sto11] Peter Stone. *The Luck of the Draw: The Role of Lotteries in Decision Making*. Oxford University Press, 2011 (cit. on pp. 14, 34, 111, 184–187).

- [Sto16] Peter Stone. *Sortition, Voting, and Democratic Equality*. In: *Critical Review of International Social and Political Philosophy* 19.3 (2016), pp. 339–356.  (cit. on pp. 14, 184, 185).
- [Sun14] Albert Sun. *Divide your rent fairly*. 2014. <https://www.nytimes.com/interactive/2014/science/rent-division-calculator.html> (cit. on p. 15).
- [SW97] Cliff Stein and Joel Wein. *On the Existence of Schedules That Are Near-Optimal for Both Makespan and Total Weighted Completion Time*. In: *Operations Research Letters* 21.3 (1997), pp. 115–122.  (cit. on p. 53).
- [SY93] George Steiner and Scott Yeomans. *Level Schedules for Mixed-Model, Just-in-Time Processes*. In: *Management Science* 39.6 (1993), pp. 728–735.  (cit. on p. 114).
- [Szp10] George Szpiro. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present*. Princeton University Press, 2010 (cit. on p. 110).
- [Thi95] T. N. Thiele. *Om flerfoldtsvalg*. In: *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger* (1895), pp. 415–441 (cit. on p. 98).
- [TM12] Markus Triska and Nysret Musliu. *An Effective Greedy Heuristic for the Social Golfer Problem*. In: *Annals of Operations Research* 194.1 (2012), pp. 413–425.  (cit. on p. 54).
- [TVV04] Kalyan T. Talluri, Garrett Van Ryzin, and Garrett Van Ryzin. *The Theory and Practice of Revenue Management*. Vol. 1. Springer, 2004 (cit. on p. 135).
- [UKH22] UK Health Security Agency. *Test and Trace Public Advisory Group Report*. 2022. [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/1073467/Test\\_and\\_Trace\\_Public\\_Advisory\\_Group\\_Report.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1073467/Test_and_Trace_Public_Advisory_Group_Report.pdf)  (cit. on p. 9).
- [UNH20] United Nations High Commissioner for Refugees. *Global Resettlement Needs 2021*. 2020 (cit. on p. 133).
- [UNH21] United Nations High Commissioner for Refugees. *Projected Global Resettlement Needs 2022*. 2021. <https://www.unhcr.org/en-us/protction/resettlement/60d320a64/projected-global-resettlement-needs-2022-pdf.html>  (cit. on p. 8).
- [UNH22] United Nations High Commissioner for Refugees. *Global Trends: Forced Displacement in 2021*. 2022. <https://www.unhcr.org/62a9d1494/global-trends-report-2021> (cit. on pp. 8, 133).
- [UNHnd] United Nations High Commissioner for Refugees. *Refugee Data Finder*. n.d. <https://www.unhcr.org/refugee-statistics/download/?url=qc4U7O>  (cit. on pp. 8, 133).
- [VB21] Alberto Vera and Siddhartha Banerjee. *The Bayesian Prophet: A Low-Regret Framework for Online Decision Making*. In: *Management Science* 67.3 (2021), pp. 1368–1391.  (cit. on p. 134).
- [VBG21] Alberto Vera, Siddhartha Banerjee, and Itai Gurvich. *Online Allocation and Pricing: Constant Regret via Bellman Inequalities*. In: *Operations Research* 69.3 (2021), pp. 821–840.  (cit. on p. 134).

- [Ver22] Philipp Verpoort. *GroupSelect*. 2022. <https://github.com/sortitionfoundation/groupselect-app> (cit. on pp. 5, 10, 52, 60).
- [vNeu53] John von Neumann. *A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem*. In: *Contributions to the Theory of Games*. Ed. by W. Kuhn and A. W. Tucker. Vol. 2. Princeton University Press, 1953, pp. 5–12 (cit. on p. 114).
- [VVS10] Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. *Optimal Online Assignment with Forecasts*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2010, p. 109.  (cit. on p. 135).
- [Wam07] Brian Wampler. *Participatory Budgeting in Brazil: Contestation, Cooperation, and Accountability*. Pennsylvania State University Press, 2007 (cit. on p. 1).
- [Was21] Washington Climate Assembly. *Washingtonians Finding Solutions Together: Washington Climate Assembly Final Report*. 2021. [https://www.waclimateassembly.org/\\_files/ugd/09fdff\\_7123b8b71d304311b075bd5d8460eb06.pdf](https://www.waclimateassembly.org/_files/ugd/09fdff_7123b8b71d304311b075bd5d8460eb06.pdf)  (cit. on p. 10).
- [Wei86] Clarice Ring Weinberg. *Applicability of the Simple Independent Action Model to Epidemiologic Studies Involving Two Factors and a Dichotomous Outcome*. In: *American Journal of Epidemiology* 123.1 (1986), pp. 162–173.  (cit. on p. 28).
- [WHB+09] Gill Ward, Trevor Hastie, Simon Barry, Jane Elith, and John R. Leathwick. *Presence-Only Data and the EM Algorithm*. In: *Biometrics* 65.2 (2009), pp. 554–563.  (cit. on pp. 24, 29).
- [Wor22] World Bank Group. *GINI index (World Bank estimate)*. 2022. <https://data.worldbank.org/indicator/SI.POV.GINI>  (cit. on p. 20).
- [WSC19] Richard Wike, Laura Silver, and Alexandra Castillo. *Many Across the Globe Are Dissatisfied With How Democracy Is Working*. Pew Research Center. 2019. <https://www.pewresearch.org/global/2019/04/29/many-across-the-globe-are-dissatisfied-with-how-democracy-is-working/>  (cit. on pp. 1, 160).
- [ZLT21] Gil Ben Zvi, Eyal Leizerovich, and Nimrod Talmon. *Iterative Deliberation via Metric Aggregation*. In: *Proceedings of the International Conference on Algorithmic Decision Theory (ADT)*. 2021, pp. 162–176.  (cit. on pp. 53, 159).

# APPENDIX

# Appendix for Chapter 2: Fair Algorithms for Selecting Citizens' Assemblies



## A.1 Desiderata for Sortition in the Political Science Literature

In this paper, we approach the problem of panel selection from a pragmatic angle. We ask: taking as given the overall panel selection process (sending out invitations uniformly at random, and then using quotas to enforce representativeness), what is the best selection algorithm for practitioners to use?

To identify desirable properties of a selection algorithm, it is natural to take inspiration from political theory, where advantages and disadvantages of sortition have been discussed in detail [Cou19; Eng89; Fis18; Smio9; Sto16]. However, one should not expect the political theory literature to give concrete instructions for a practical selection algorithm, since the literature focuses on an idealized sortition process that ignores the complications of the real-world settings in which panels must be selected. In particular, the literature assumes that panels can be selected by sampling directly from the population, whereby each member of the population is selected with equal probability and will agree to participate if invited [CM99; Par11; Sto11]. We refer to this procedure as *idealized sortition*. Usually, in practice, a large majority of people decline to participate when invited [Smio9].

Though this literature does not immediately prescribe a practical selection algorithm, it informs our approach by identifying the values that should be pursued when designing selection algorithms. In this section, we outline several prominently advocated properties of idealized sortition, discuss how they are or are not conducive to algorithmic implementation, and describe how these properties complement or contradict one another. Ultimately, our approach of making selection probabilities as equal as possible strives for *promotion of equality*, while guaranteeing the achievement of *representativeness* as implemented by practitioners via quotas.

### A.1.1 Properties of Idealized Sortition

Following a model developed by Engelstad [Eng89] and elaborated upon by others [CM99; Sto11], sortition should simultaneously (1) *promote equality*, (2) *ensure representativeness*, (3) *maximize efficiency*, and (4) *protect against conflict and domination*.

#### Equality

According to Engelstad, “The strongest normative argument in favour of sortition is linked to the idea of social equality and individual welfare”, which stems from the fact that every constituent has an equal selection probability [Eng89]. Subsequent work in political theory has reaffirmed the importance of equal

[Cou19] Courant (2019): *Sortition and Democratic Principles: A Comparative Analysis*.

[Eng89] Engelstad (1989): *The Assignment of Political Office by Lot*.

[Fis18] Fishkin (2018): *Democracy When the People Are Thinking*.

[Smio9] Smith (2009): *Democratic Innovations*.

[Sto16] Stone (2016): *Sortition, Voting, and Democratic Equality*.

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Par11] Parker (2011): *Randomness and Legitimacy in Selecting Democratic Representatives*.

[Sto11] Stone (2011): *The Luck of the Draw*.

selection probabilities, even if different authors deduce this importance from slightly different ideals: Some [Fis09; Fis18; Par11; Sto16] see the equal selection probabilities of idealized sortition as an embodiment of *democratic equality*, the ideal that a democratic decision-making process should give equal consideration to all of its constituents' preferences. Other authors [CM99; Par11] stress equal probabilities as the hallmark of (prospect-regarding [Rae89]) *equality of opportunity*. A related argument is made by Stone [Sto11; Sto16]. Rather than seeing equality as the goal in its own right, he views random allocation with equal probability as the only way to satisfy *allocative justice* in the distribution of public offices among constituents who all have equal claims to authority.

As we discuss in the introduction, perfect equality of selection probabilities is not attainable within the constraints of practical sortition. In this paper, we handle this impossibility by proposing a more gradual version of this goal: Subject to achieving descriptive representation, one should make selection probabilities as equal as possible. The view of political office as a good, and of sortition as a means to allocative justice [Sto16], is a natural foundation for the approach of treating panel selection as a problem of fair division (see [supplementary information 9](#) of the full version).

### Representativeness

Another important benefit of ideal sortition is that, with high probability, the composition of the panel will resemble the population along all dimensions of interest [Sto11]. Descriptive representation is a crucial assumption in Fishkin's argument that the result of a deliberative minipublic can reveal the likely outcome of the whole population deliberating [Fis09; Fis18]. In addition to its contribution to the quality of deliberation, descriptive representation is particularly valuable in contexts of mistrust and marginalization [Man99].

As stated above, the statistical properties of idealized sortition imply that *any possible* division of the population is likely to be represented close to proportionally on the panel, provided that the panel size is sufficiently large. By contrast, no such guarantee can be provided in the realistic setting where constituents decline to participate, which forces practitioners to select specific features for which they want to enforce descriptive representation using quotas. Whereas our approach focuses on making selection probabilities close to equal, we do not sacrifice descriptive representation for this goal. Rather, organizing bodies can still set quotas to ensure a desired level of descriptive representation, and our methods only use the remaining freedom within these constraints to promote equality. In this way, our method allows an assembly organizer to trade off representation and equality by tightening or loosening the quotas.

### Efficiency

In comparison to selecting representatives by election, some authors argue that sortition is more efficient because it requires fewer resources [CM99; Eng89]. For instance, campaigning and organizing elections are not necessary. Arguably, this argument is more specific to the benchmark of elections than to sortition, and subsequent works have put little emphasis on this point [Sto11].

[Fis09] Fishkin (2009): *When the People Speak*.

[Fis18] Fishkin (2018): *Democracy When the People Are Thinking*.

[Par11] Parker (2011): *Randomness and Legitimacy in Selecting Democratic Representatives*.

[Sto16] Stone (2016): *Sortition, Voting, and Democratic Equality*.

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Rae89] Rae (1989): *Equalities*.

[Sto11] Stone (2011): *The Luck of the Draw*.

[Man99] Mansbridge (1999): *Should Blacks Represent Blacks and Women Represent Women?*

[Eng89] Engelstad (1989): *The Assignment of Political Office by Lot*.

When considering the design of the selection algorithm, the only major resource one might seek to use efficiently is time — namely, the time the algorithm takes to run. Given that the selection of the panel from the pool is only a minor task in organizing and convening a citizens' assembly, as organizers spend much more time recruiting the pool and organizing the deliberation. For this reason, reducing the running time of the algorithm seems a frivolous efficiency. As we show in Table 1, our algorithm LEXIMIN runs in seconds for most instances and an hour at most. This is significantly longer than the running time of the benchmark algorithm LEGACY, but much faster than the process of executing other selection algorithms using dice and spreadsheets, as practiced by some organizations. We take this as an indicator that hours versus minutes of running time is not a significant consideration in terms of efficiency.

Existing algorithms often confront practitioners with a hard trade-off between representation and computational efficiency, since more numerous and tighter quotas may drastically increase the running time of these algorithms. While such a concern cannot be theoretically ruled out for any known algorithm (supplementary information 6 of the full version), our algorithms delegate the task of finding panels to a state-of-the-art ILP solver, a mature technology routinely used to solve much harder tasks [GHG+21] than all panel-selection subtasks we have encountered. Therefore, we expect our algorithm to allow for much more complex quotas without substantial increases in running time; the fundamental trade-offs between representativeness and equality, of course, persist. Our algorithms also have an advantage in the (undesirable) situation where no panel formed from the pool can satisfy the quotas. Whereas existing algorithms enter an infinite loop in this situation until the user gives up, our algorithms' first call to the ILP solver will immediately reveal that the quotas are infeasible; in these situations, our implementation solves a second ILP to suggest a minimal relaxation of the quotas that can be satisfied.

[GHG+21] Gleixner et al. (2021): *MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library*.

### Protection against Conflict and Domination

A final family of arguments stresses that, if the members of a panel are chosen via idealized sortition, this procedure prevents interested parties from swaying the selection for their benefit [CM99; Dow09; Eng89]. Stone summarizes these arguments as follows:

“First, [sortition] can prevent wrongful action on the part of the agent who must select officials. [...] Second, it can prevent wrongful action on the part of the officials selected. If the method of selection is in any way predictable, outside interests might bribe or threaten officials into conformity with their wishes. If the method is unpredictable, then such wishes cannot be expressed at least until the results of the lottery become known. [...] Finally, competing elites unable to stack the political process in their favor have less to fight about.” [Sto11]

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Dow09] Dowlen (2009): *Sorting Out Sortition*.

[Eng89] Engelstad (1989): *The Assignment of Political Office by Lot*.

[Sto11] Stone (2011): *The Luck of the Draw*.

In the practical setting of sortition, the additional stages of the selection process (as compared to idealized sortition) inherently create opportunities for dishonest agents to influence the composition and the decisions of the panel in ways that cannot be remedied by a change of selection algorithm. First, with respect to

concerns about wrongful action on the part of the officials, the panel organizers wield a lot of influence in sending out the invitations, setting the quotas, and handling the process of selecting the panel from the pool.

More fundamentally, when any selection algorithm enforcing descriptive representation is used, a dishonest pool member can significantly increase their chances of selection by misrepresenting their features. For example, this pool member might pretend to have a different political orientation because they know that people with this orientation are unlikely to participate, and thus are likely to be underrepresented in the pool. Since, on average, the selection algorithm must choose pool members from this group with higher probability, reporting this feature will likely increase the agent's probability of being selected for the panel. So long as practitioners seek to enforce descriptive representation in the presence of unequal rates of participation across subgroups, this type of manipulation seems unavoidable.

If, despite these challenges, one wanted to design a selection algorithm to discourage manipulation, one would have to target a specific kind of manipulation. For instance, for reducing the effect of bribing or intimidating pool members before they are selected, the algorithm within our framework minimizing the largest selection probabilities might be appropriate. Such an algorithm would increase the cost to the manipulator since any bribed pool member would have a substantial chance of not being selected to the panel, rendering the bribe futile. For other threat models, it would be natural for the selection algorithm to maximize not only the uncertainty of each agent being selected for the panel individually but the uncertainty about the composition of the whole panel. A selection algorithm maximizing this objective of *maximum entropy* could, in principle, be implemented by uniformly drawing sets of  $k$  pool members, repeating this process until one set satisfies all quotas. Whether this selection algorithm can be sped up to the degree of being practically relevant is an interesting question for future work.

### A.1.2 Beyond Idealized Sortition, and the Objective of Maximal Fairness

As we have described, a large body of political theory literature characterizes the desiderata and benefits of *idealized* sortition. However, there is also research that engages, as we do in this work, with sortition beyond the idealized assumption that everyone is willing to participate. Such work often mentions *stratified sampling* [CM99; Ley19; Par11; SBJB20; Smio9] as a sampling method that can be used to reestablish descriptive representation despite differing response rates across subpopulations. For details on stratified sampling and how it relates to our work, see [supplementary information 3](#) of the full version. In the political theory literature touching on stratified sampling, several authors point out that the benefits of idealized sortition do not perfectly extend to stratified sampling [Dow08; Par11; Smio9; Sto11]. To our knowledge, however, the literature stops short of proposing more gradual ideals, such as the maximal fairness objective we propose to approximate equality.

[CM99] Carson and Martin (1999): *Random Selection in Politics*.

[Ley19] Leydet (2019): *Which Conception of Political Equality Do Deliberative Minipublics Promote?*

[Par11] Parker (2011): *Randomness and Legitimacy in Selecting Democratic Representatives*.

[SBJB20] Steel et al. (2020): *Rethinking Representation and Diversity in Deliberative Minipublics*.

[Smio9] Smith (2009): *Democratic Innovations*.

[Dow08] Dowlen (2008): *The Political Potential of Sortition*.

[Sto11] Stone (2011): *The Luck of the Draw*.

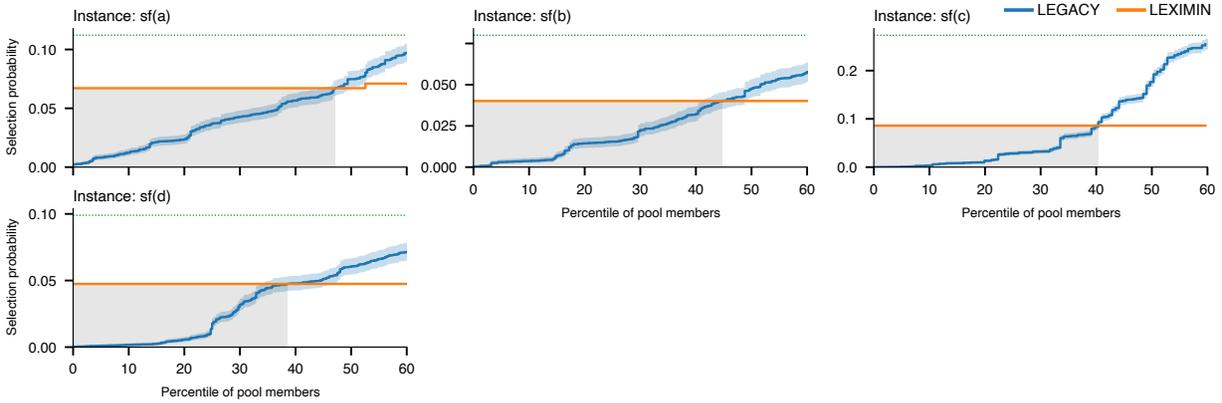
## A.2 Additional Figures and Tables

**Table A.1:** Gini coefficient and geometric mean of probability allocations of both algorithms, for each instance. On every instance, LEGACY has a lower Gini coefficient and a larger geometric mean. For computing the geometric mean, we slightly correct upward empirical selection probabilities of LEGACY that are close to zero (as described in methods section “Statistics” of the full version).

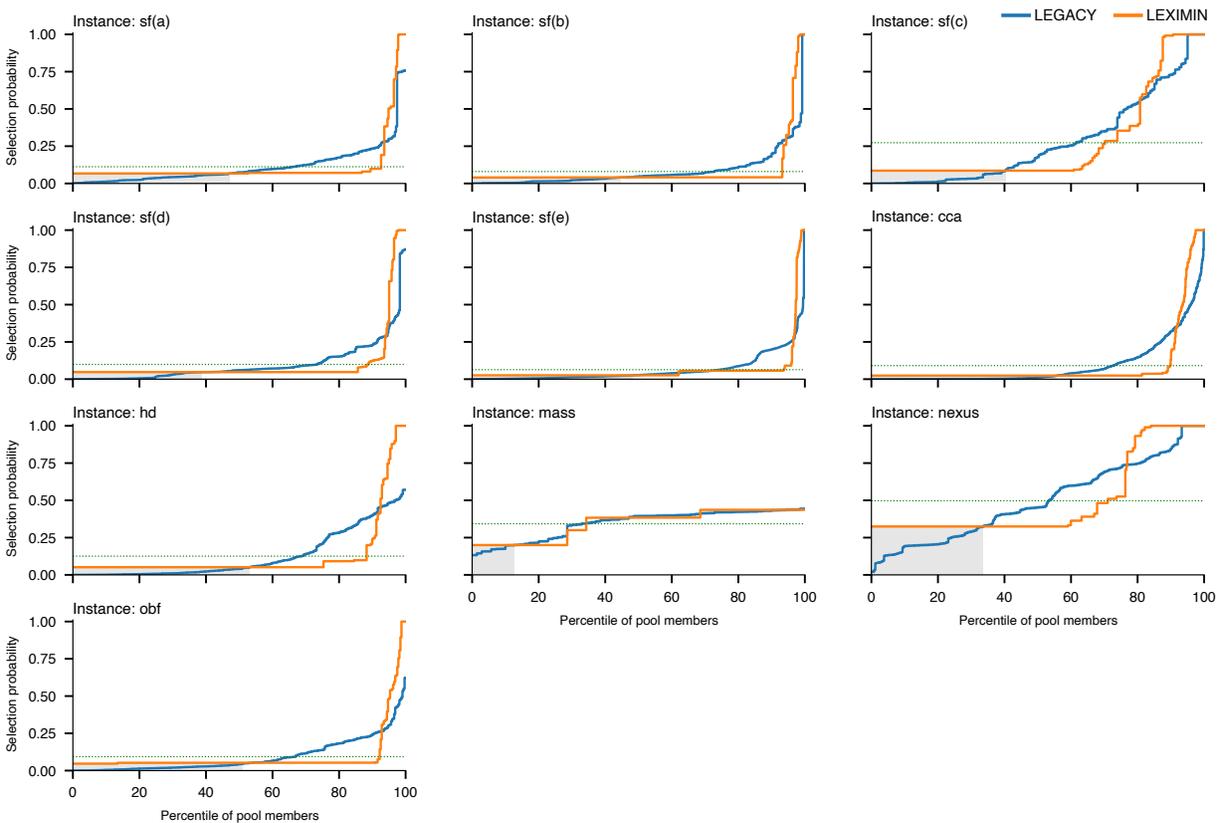
Instance	Gini coefficient of LEGACY (lower is fairer)	Gini coefficient of LEXIMIN (lower is fairer)	Geometric mean of LEGACY (higher is fairer)	Geometric mean of LEXIMIN (higher is fairer)
sf (a)	51.2%	37.3%	6.5%	8.1%
sf (b)	59.6%	47.4%	3.5%	4.8%
sf (c)	57.0%	52.5%	8.3%	16.3%
sf (d)	59.3%	48.7%	3.5%	6.0%
sf (e)	64.4%	51.2%	2.2%	3.9%
cca	75.3%	67.8%	0.7%	3.5%
hd	64.5%	52.9%	3.1%	7.3%
mass	14.9%	14.8%	32.6%	32.7%
nexus	30.8%	25.4%	40.9%	44.2%
obf	58.9%	42.7%	3.7%	6.2%

Instance	Share selected by LEGACY with probability below LEXIMIN minimum selection probability
sf (a)	47.1%
sf (b)	44.8%
sf (c)	40.4%
sf (d)	38.6%
sf (e)	48.4%
cca	55.8%
hd	53.1%
mass	12.9%
nexus	33.6%
obf	51.1%

**Table A.2:** For each instance, the share of pool members selected with lower probability by LEGACY than the minimum selection probability of LEXIMIN is shown. This corresponds to the width of the shaded boxes in Figures 2.2, A.1, and A.2.



**Figure A.1:** Selection probabilities given by LEGACY and LEXIMIN to the bottom 60% of pool members on the 4 instances that are not shown in Figure 2.2. Pool members are ordered across the x axis in order of increasing probability given by the respective algorithms. Shaded boxes denote the range of pool members with a selection probability given by LEGACY that is lower than the minimum probability given by LEXIMIN. LEGACY probabilities are estimated over 10 000 random panels and are indicated with 99% confidence intervals (as described in methods section “Statistics” of the full version). Green dotted lines show the equalized probability ( $k/n$ ).



**Figure A.2:** Selection probabilities given by LEGACY and LEXIMIN on all ten instances. Pool members are ordered across the x axis in order of increasing probability given by the respective algorithms. In contrast to Figures 2.2 and A.1, this graph shows the full range of selection probabilities (up to the 100th percentile). Shaded boxes denote the range of pool members with a selection probability given by LEGACY that is lower than the minimum probability given by LEXIMIN. LEGACY probabilities are estimated over 10 000 random panels and are indicated with 99% confidence intervals (as described in methods section “Statistics” of the full version). Green dotted lines show the equalized probability ( $k/n$ ).