Thesis:
Algorithms for Fair Division

David Kurokawa

CMU-CS-17-122
July 26, 2017

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Ariel D. Procaccia (Chair)
Manuel Blum
Zico Kolter
Tuomas Sandholm
Ioannis Caragiannis (University of Patras)

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

This research was sponsored by the National Science Foundation under grant numbers CCF-1215883, IIS-1350598, and CCF-1525932, Microsoft, and the Natural Sciences and Engineering Research Council of Canada Fellowship.

Keywords: algorithmic game theory, fair division, mechanism design, cake cutting, indivisible goods, maximin share.

Abstract

This thesis covers various aspects of *fair division*: allocating goods/items among interested parties while maintaining axiomatic or quantitative properties. The difficulty arises from the heterogeneous valuations of the agents. That is, agents do not necessarily agree on the value of a set of goods. We consider four settings in depth.

First, we dissect the problem of allocating k equal rewards to the best k of n agents when our only information comes from the agents evaluating each other. We give an an algorithm to accurately do so while disincentivizing agents from strategically lying to benefit themselves (a property called impartiality). We further show our accuracy is best possible under our metric.

Second, we expand the previous setting to when we wish to rank the agents instead of merely producing the top k of them. Here too, we give algorithms to accurately perform this task while maintaining impartiality. We expand on the connection to the first setting by extrapolating the generalization further and demonstrating several impossibility results.

Third, we consider the setting of cake cutting: allocating a single divisible good. We examine the classic problem of envy-free cake cutting when the agents have restricted valuations — specifically, they have piecewise uniform/constant/linear densities. We show that the restriction is no restriction at all, but when parametrizing the complexity of the densities, yields a significant reduction in difficulty. We further examine the cake cutting setting when agents are strategic and demonstrate the existence of standard equilibrium concepts in the space (despite infinite action spaces).

Finally, we expand upon the concept of the maximin share guarantee (a property seen in the study of indivisible goods). We give several results on the existence of the property and approximations to it in various settings.

Contents

Chapter 1

Introduction

Division of goods among interested parties is no new task. Splitting a scrumptious dessert among friends or dividing up a family estate among heirs are just two such examples. At its heart, the core problem is to take some set of goods and to allocate them among a set of agents in a way that is desirable or *fair*. Unfortunately, due to the competing and heterogeneous nature of said agents, it is difficult to define a metric of desirability or fairness. Therefore, often ad hoc and fairly subjective approaches such as negotiation are employed to reach a consensus. These methods fall short in two critical ways: they cannot be automated and are heavily qualitative in their success. Owing to the need for simplicity and automation, drafting is a common approach when the goods are discrete. That is, agents take turns choosing goods they will keep until all goods are taken. Unfortunately, as we will see in Chapter 6, this is often far from ideal by various metrics.

A key reason we are concerned with automation and quantification of fairness is we are involved in an effort to bring these ideas to the real world by building a fair division website called *Spliddit* [50], available at `www.spliddit.org`. Quoting from the website:

> *"Spliddit is a not-for-profit academic endeavor. Its mission is twofold:*
>
> - *To provide easy access to carefully designed fair division methods, thereby making the world a bit fairer.*
> - *To communicate to the public the beauty and value of theoretical research in computer science, mathematics, and economics, from an unusual perspective."*

Since its launch in November 2014, Spliddit has attracted more than 100,000 users (as of July 9, 2017), and has received significant press coverage.

The strive for fairness itself is self-explanatory from the perspective of an external party (i.e. one not eligible for receiving any goods) but is further motivated by the fact that in reality, agents do not act entirely out of self-interest but may act charitably in the spirit of fairness — see for instance, [47] or arguably, humanity in general. Such behavior could of course be described by more advanced agent utility functions, but accurately modelling such functions is neither easy nor even well-defined due to the fluid, vacillating, and contradictory nature of agents in reality. Therefore, often the field abstracts this generosity component of utilities into the external party performing the allocation of the goods.

As with many areas, research in the area can be broken down by the setting or the flavor of the results. Settings in the area include concrete applications such as taxi fare division (i.e. how to divide taxi/Uber/Lyft fare among individuals going to different destinations) and rent splitting (i.e. how to divide rent among roommates in asymmetric situations) to more theoretical and fundamental settings such as the indivisible good setting (which we describe in more depth below). In this thesis we will largely focus on the following four fair division settings.

1. (Chapter 2) n agents rank each other to determine who is in the top $k < n$.
   This work largely stemmed from a real-world motivating example of the problem: the NSF began a test program of having grant applicants review each others' applications to determine who were given grants.

   The main question here is how to allocate the top k positions in a way that is both accurate and *impartial*: no agent has incentive to misreport his ranking of others. We give an algorithm that accomplishes this and demonstrate that there exists no impartial algorithm that is more accurate under our metric.

   This work is based off our results in [60].

2. (Chapter 3) n agents rank each other to determine a complete ranking of the agents.
   This is a generalization of the previous setting and it too comes from a real-world motivating example. When hiring freelancers for a project it is often difficult or infeasible for the employer to comb through the candidates and assess their competence for the work. A suggested approach was therefore to have the agents rank themselves so that the employer can best focus their efforts.

   Similar to the previous setting we wish to produce a complete ranking of the agents that is accurate while maintaining impartiality. In this case, we measure our accuracy by its proximity to replicating various social ranking functions (such as Borda and Kemeny). We give multiple incomparable algorithms that are indeed accurate and impartial and establish impossibility results for stronger notions of impartiality.

   This work is based off our results in [57].

3. (Chapters 4 and 5) n agents divide up a divisible good (one that can be cut, sliced, and diced) — known as cake cutting.
   This is arguably the oldest of fair division problems. Here we are interested in a single good[1] that can be split apart arbitrarily — and therefore is often represented/imagined as a cake. The difficulty arises due to the assumption that the agents have heterogeneous valuations (i.e. they do not necessarily agree on the value of subsets of the cake).

   In Chapter 4 we restrict our attention to when all agents have valuation functions for the cake under special classes (specifically, piecewise uniform/constant/linear value densities) and present two slightly paradoxical results. The first which says this restriction is no restriction at all in terms of a well-studied problem in the field (the complexity of producing envy-free allocations), while the second says this restriction, when coupled with a more flexible complexity model, greatly simplifies the problem. Chapter 5 explores the aspect of cake cutting from a game theoretic viewpoint where agents will lie/game a system to maximize their utility. We show that the standard notion of equilibrium in such settings

[1] Multiple such goods can be considered as one all-encompassing good.

(the subgame perfect Nash equilibrum) exist and give analysis of their properties.

This work is based off our results in [59] and [23].

4. (Chapter 6) n agents divide up m discrete/indivisible goods (ones that cannot be cut, sliced, and diced).
   This is perhaps the most fundamental of fair division problems and has wide applications ranging from divorce settlements to acquiring fresh talent in professional sport leagues. It is also a generalization of the previous setting as that is a limiting case of this.

   In Chapter 6 we dissect the fairness concept known as the maximin share guarantee. We prove that it may not exist for general valuations, but conversely demonstrate that a 2/3 approximation is guaranteed to exist and can be found in polynomial time. We further show that under a natural randomized model we can expect to have such guarantees with high probability.

   This work is based off our results in [63] and [62].

Regarding flavors of results, applications of fair division ideas to real world settings is a burgeoning area with developments in settings such as course scheduling in universities among students (see [27], [80]), and the aforementioned rent splitting among roommates (see [48], [5]). Contributions in Chapters 2 and 3, and our work in [61] are of this flavor. Purely theoretical advancements examining existence or complexity issues of fairness concepts is another direction often seen in the literature. Such results include those revolving around the production of envy-free allocations in bounded time in the cake-cutting setting (see [9], [20], [92]) as well as the complexity of producing an approximation to a well-known equilibrium concept known as competitive equilibrium (see [79]). Chapters 4, 5, 6, and our work in [30] make such contributions to the area. An arguably underrepresented flavor of work is that of examining strategic agents in fair division settings — largely due to it being an ocean of simple impossibility results. Our work in Chapters 2, 3, and 5 however break this trend and give positive results for their settings.

Chapter 2

Impartial Peer Review

2.1 Introduction

The Sensors and Sensing Systems (SSS) program of the National Science Foundation (NSF) recently experimented with a drastically different peer review method. Traditionally, grant proposals submitted to a specific program are evaluated by a panel of reviewers. Potential conflicts of interest play a crucial role in composing the panel; most importantly, principal investigators (PIs) whose proposals are being evaluated by the panel cannot serve on the panel. In stark contrast, the new peer review method — originally designed by Merrifield and Saari [71] for the review of proposals for telescope time — requires the PIs themselves to review each other's proposals! A "dear colleague letter" [53] explains the potential merits of the new process:

> *"This pilot is an attempt to find an alternative proposal review process that can preserve the ability of investigators to submit multiple proposals at more than one opportunity per year while encouraging high quality and collaborative research, placing the burden of proposal review onto the reviewer community in proportion to the burden each individual imposes on the system, simplifying the internal NSF review process, ameliorating concerns of conflict-of-interest, maintaining high quality in the review process, and substantially reducing proposal review costs."*

Under the Saari-Merrifield mechanism, each PI must review m proposals submitted by other PIs; in the NSF pilot, $m = 7$. The PI then ranks the m proposals according to their quality. These reviews are aggregated using the Borda count voting rule, so each PI awards $m - i$ points to the proposal she ranks in position i. A proposal's overall rating is the average over the points awarded by the m PIs who reviewed it. Additionally, a PI's own proposal receives a small bonus based on the similarity between the PI's submitted ranking and the aggregate ranking of the proposals she reviewed; this is meant to encourage PIs to make an effort to produce accurate reviews.

The NSF pilot sparked a lively debate amongst mechanism design and social choice researchers in the blogosphere [87, 98, 72]. While most researchers seem to agree that the NSF should be commended for trying out an ambitious peer review method, serious concerns have been raised regarding the pilot mechanism itself. Perhaps most strikingly, while the NSF announcement [53] states that the "theoretical basis for the proposed review process lies in an area of mathematics referred to as mechanism design", the pilot mechanism provides no theoretical guarantees. In particular, the mechanism is susceptible to strategic manipulation: PIs will often be able to advance their own proposals by giving low scores to competitive proposals (even though they may forfeit some of the small bonus for similarity to others' reviews). Furthermore, while most researchers who sit on NSF panels are well-respected, the pilot mechanism cannot control the quality (or morality) of PIs who submit proposals (and review proposals)— leaving open the very real possibility of game-theoretic mayhem.

In this chapter, we alleviate these concerns by proposing a peer review mechanism which is not susceptible to such manipulations. Each PI who submits a proposal or paper will review some other PIs' proposals or papers. Our mechanism is *impartial*: reviewers will not be able to affect the chances of their own proposals being selected. Our research challenge is therefore to *design provably impartial peer review mechanisms that provide formal quality guarantees.*

We believe that solutions to this problem truly matter. The NSF plays a huge role in enabling scientific research in the United States, and its consideration of alternative peer review methods may transform how scientific funding is allocated in the US. The need to build sound foundations for these methods therefore provides a unique opportunity for computational game theory research, and AI research more broadly.

2.1.1 Our Approach

In our setting there are n PIs, each associated with a proposal. Each PI i has a hypothetical (honest) evaluation of the quality of the proposal j, which is the rating i would give j if she were asked to review that proposal (and could not affect her own chances of selection). The (honest) *score* of a proposal is the average (honest) rating given to it by other PIs. As NSF program directors, if our budget is sufficient to fund k proposals, we would ideally want to select a set of k proposals with maximum honest score.[1] There are two obstacles we must overcome: we cannot possibly ask each PI to review all other proposals, and the reviews may be dishonest.

To address the first problem, we consider only mechanisms which request m reviews per PI (much like the NSF pilot). We define an (m,k)-*selection mechanism* as follows. First, the mechanism asks each PI to review m proposals, in a way that each proposal is reviewed by exactly m PIs; for every such pair (i,j), PI i's evaluation for proposal j is revealed. Based on these elicited reviews, the mechanism selects k vertices. The most natural (m,k)-selection mechanism is an abstract version of the NSF pilot mechanism, which we fondly refer to as the VANILLA mechanism; it chooses m reviews per PI uniformly at random (subject to the constraint that each proposal is reviewed by m PIs), and then selects the k vertices with highest average rating, based only on the sampled reviews.

Returning to the second problem — dishonest reviewing — we will consider only mechanisms where reviewers cannot affect their chances of being selected by misreporting their reviews. A selection mechanism is *impartial* if the probability of proposal i being selected is independent of the ratings given by PI i. The motivation for our work stems from the observation that the VANILLA mechanism is not impartial: we seek mechanisms that are.

[1] We distill the strategic aspects of the NSF reviewing setting and abstract away some other practical aspects, such as the fact that PIs may submit multiple proposals to the same program. However, our model and results easily extend.

How should we evaluate the impartial mechanisms we design? Without any assumptions, competing with an omniscient mechanism that maximizes underlying scores is clearly impossible.[2] We therefore use the VANILLA mechanism as our performance benchmark. Competing with VANILLA is nontrivial, because we give it the "unfair" advantage of assuming that reviews are honest, even though it is not impartial. Specifically, we say that an impartial mechanism *α-approximates VANILLA* if, in the worst case over reviews, the ratio between the expected score (based on the largely unseen set of all possible reviews) of the set of proposals selected by the impartial mechanism, and the expected score of the set of proposals selected by VANILLA, is at least α.

The choice of VANILLA as a benchmark has two main advantages. First, since the VANILLA Mechanism is an abstraction of the NSF pilot mechanism, our choice of benchmark allows us to quantify how much the NSF must sacrifice to achieve impartiality — and our results show that this sacrifice is negligible. Moreover, innovations that are closest to the current accepted practice are the most likely to be adopted.

Second, modulo its lack of impartiality, VANILLA is intuitively the "right" mechanism: it selects those nodes with the highest sampled scores. Furthermore, in an average-case model where each proposal has an intrinsic quality, and reviews are drawn from a well-behaved distribution whose expectation is the true quality of a proposal, VANILLA will pinpoint the best proposals given a sufficiently large m. Even when we assume reviews are worst-case, we can obtain an excellent approximation of VANILLA via an impartial mechanism, and that guarantee immediately extends to the average case model.

2.1.2 Our Results

In Section 2.3 we present an impartial (m, k)-selection mechanism, CREDIBLE SUBSET, which (usually) selects k proposals at random from a slightly larger pool (of size $k+m$) of eligible proposals. We prove that CREDIBLE SUBSET gives an approximation ratio of $\frac{k}{k+m}$ to VANILLA. We think of m, the number of reviews per PI, as being a small constant, and we would like to think of k, the number of proposals to be selected, as significantly larger. In particular, when $m = o(k)$, the approximation ratio goes to 1 as k goes to infinity (in an ideal world, where growth in funding outpaces growth in the quantity of work for reviewers, see Section 2.5).

In Section 2.4, we show that CREDIBLE SUBSET is the *optimal* impartial mechanism, in the sense that its approximation ratio of $\frac{k}{k+m}$ is asymptotically tight (when $k = m^2$ is a constant and the number of PIs n grows).

2.1.3 Related Work

Our paper is closely related to the work of Alon et al. [2]. In parallel with Holzman and Moulin [56], Alon et al. introduced the notion of impartial selection mechanisms (using the term "strategyproofness" for impartiality). Their model can be interpreted as a special case of our model, where $m = n - 1$ (i.e., each PI reviews all other proposals) and all the ratings are in $\{0, 1\}$. The main result of Alon et al. is the design of an impartial mechanism that approximates

[2] Indeed, even VANILLA with truthful reviews will be unable to do so!

the score of the optimal subset of k vertices to a factor that goes to 1 as k grows. When $m = n-1$ and all ratings are in $\{0,1\}$, this is equivalent to approximating Vanilla: Vanilla can see all ratings and will select the optimal subset. But when $m \ll n-1$ we cannot reason about scores directly, as Alon et al. do. In fact, in this regime, which is typical for a peer review setting, our results are incomparable to theirs: our mechanisms use far less information, but the performance of these mechanisms is (necessarily) measured against a weaker benchmark.

Other papers on impartial mechanisms include the ones by de Clippel et al. [33], Holzman and Moulin [56], Fischer and Klimm [45], Berga and Gjorgjiev [13], Tamura and Ohseto [96], and Mackenzie [66].

Merrifield and Saari [71] are not the first researchers to suggest improvements to the peer review process, although most other papers focus on conference reviewing [78, 51, 39, 89]. For example, in a AAAI'11 paper, Roos et al. [89] propose a method for calibrating the ratings of potentially biased reviewers via a maximum likelihood estimation (MLE) approach.

2.2 The Model

Let $N = \{1,2,\dots,n\}$ be the set of proposals and also the set of strategizing reviewers. Each reviewer i has an estimate of the quality of every other proposal $j \neq i$ — the score i would give j if i honestly reviewed j. We represent this setting as a weighted, complete, directed graph $G = (N, E, w_G)$ where $E = \{(i,j) \mid i,j \in N, i \neq j\}$, and $w_G(i,j) \in \mathbb{R}^+$ is the quality of j according to i's evaluation. We call G the *underlying graph*.

Let m be the number of proposals that each PI can review, which must equal to the number of reviews each proposal receives (we assume each PI submits one proposal). In our model, m is the number of outgoing edges from each vertex and the number of incoming edges to each vertex. Slightly abusing terminology, we say that a directed graph is *m-regular* if it satisfies these properties.

A peer review process is governed by an (m,k)-*selection mechanism*, which works in two stages:

1. The mechanism selects (possibly randomly) a directed m-regular graph $G^m = (N, E(G^m))$, called the *sampled graph*. We assume this graph is drawn prior to the next step: that the sampling is done all at once independent of the edge weights.
2. Given the underlying graph G, the weight $w_G(i,j)$ is revealed for each edge $(i,j) \in E(G^m)$. The mechanism then maps these elicited ratings to a subset of selected vertices of size at most k.

Step 1 corresponds to the mechanism assigning m proposals to each PI. Based on the reviews $w_G(i,j)$ for $(i,j) \in E(G^m)$, in Step 2, the mechanism selects a subset of at most k proposals that will receive funding.

Let us reinterpret the NSF pilot mechanism [53] in this framework, abstracting away details such as the use of Borda count and the bonus component for accurate reviews. To this end, let $\mathcal{G}^m$ denote the uniform distribution over m-regular graphs. Given a weighted m-regular graph G^m, let

$$\mathrm{top}_k(G^m) \in \operatorname*{arg\,max}_{Y \subseteq N:\ |Y|=k} \sum_{i \in Y} \sum_{j:(j,i) \in E(G^m)} w_G(j,i),$$

breaking ties lexicographically (i.e., the k nodes with the largest sum of incoming edge weights in the graph). Now, the *Vanilla* mechanism, denoted $\mathcal{M}^v$, is defined as follows:

Vanilla (G, m, k)
Draw $G^m \sim \mathcal{G}^m$.
Return $\text{top}_k(G^m)$.

Algorithm 1: Vanilla

Intuitively, the mechanism assigns proposals to PIs for review based on the graph G^m, and then returns the k highest-rated reviews based on the sampled reviews (for convenience we look at the sum of ratings, which is equivalent to the average).

For a mechanism $\mathcal{M}$ and an underlying graph G, let $\mathcal{M}(G)$ be a random variable, which takes the value $X \subseteq N$ with the same probability that $\mathcal{M}$ outputs X when the underlying graph is G. Then we can use $\mathbb{P}[i \in \mathcal{M}(G)]$ to denote the probability that $\mathcal{M}$ selects $i \in N$ when the underlying graph is G. We say that $\mathcal{M}$ is *impartial* if for any $i \in N$ and any two underlying graphs G and G' that differ only in the weights on the outgoing edges of i, $\mathbb{P}[i \in \mathcal{M}(G)] = \mathbb{P}[i \in \mathcal{M}(G')]$.

Unfortunately, Vanilla is clearly not impartial. To see this, let $k = 1$, $m = 1$, and define the weights of G and G' as follows:

$$w_G(i,j) = \begin{cases} n+1 & i = 1 \\ 1 & j = 1, i \neq 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$w_{G'}(i,j) = \begin{cases} 0 & i = 1 \\ 1 & j = 1, i \neq 1 \\ 0 & \text{otherwise} \end{cases}.$$

Then $\mathbb{P}[1 \in \mathcal{M}^v(G)] = 0$, whereas $\mathbb{P}[1 \in \mathcal{M}^v(G')] = 1$ (using lexicographic tie-breaking, 1 would be selected even if only 0-weight edges are sampled).

One of our main aims for this chapter is to design (m,k)-selection mechanisms that are simultaneously impartial (unlike Vanilla), yet similarly practical in terms of the number of reviews per proposal and similar in the quality of the output. We measure the quality of a mechanism by the expected score of the vertices it selects. Formally, let $\text{sc}(i,G) = \sum_{(j,i)\in E} w_G(j,i)$ be the *score of vertex i in G*, and let $\text{sc}(X,G) = \sum_{i\in X} \text{sc}(i,G)$ be the score of a set of vertices $X \subseteq N$ in G. We can now define

$$\text{sc}(\mathcal{M}, G) = \mathbb{E}_{X\sim\mathcal{M}(G)}[\text{sc}(X,G)].$$

This is our optimization objective.

Note that for some underlying graphs G, Vanilla itself may do poorly in terms of $\text{sc}(\mathcal{M}, G)$. As an extreme example, let $k = 1$, $m = 1$, and define the weights of the underlying graph G as follows:

$$w_G(i,j) = \begin{cases} 1000 & i = 1 \text{ and } j = 2 \\ 1/n & j = 1 \\ 0 & \text{otherwise.} \end{cases}$$

It is very likely that the edge $(1,2)$ will not be sampled by Vanilla, and therefore the mechanism will likely select vertex 1, the only one with non-zero score. However, $\mathrm{sc}(1,G) = \frac{n-1}{n} < 1$, whereas $\mathrm{sc}(2,G) = 1000$. This is not a shortcoming of Vanilla specifically — it is clear that such examples can be constructed for any (m,k)-selection mechanism when m is much smaller than n.

Nevertheless, we can use Vanilla as a benchmark. We wish to design impartial mechanisms whose quality guarantee is quite close to that of Vanilla pointwise (assuming all reviews given to Vanilla were truthful). We say that an (m,k)-selection mechanism $\mathcal{M}$ α-approximates Vanilla, for $\alpha = \alpha(m,n,k) \leq 1$, if for *every* underlying graph G,

$$\frac{\mathrm{sc}(\mathcal{M},G)}{\mathrm{sc}(\mathcal{M}^v,G)} \geq \alpha.$$

2.3 The Credible Subset Mechanism

In this section we present and analyze an (m,k)-selection mechanism, the *Credible Subset* mechanism. The mechanism relies on two ideas:

1. Every vertex that has the potential to be among the top k by changing its outgoing edges must have a chance to be selected. Such vertices are called *credible*. There are not too many of them, and they include the actual top k.
2. A credible vertex can potentially affect the number of credible vertices (by giving a low score to another credible vertex), and therefore the probability of selecting a credible vertex must be independent of the number of credible vertices.

The Credible Subset mechanism, denoted $\mathcal{M}^{cs}$, formally works as follows.

Credible Subset (G,m,k)
Draw $G^m \sim \mathcal{G}^m$.
$P \leftarrow \{i \notin \mathrm{top}_k(G^m) \mid \text{if } i \text{ reported } \forall j : w(i,j) = 0, i \text{ would be in } \mathrm{top}_k(G^m)\}$
$S \leftarrow \mathrm{top}_k(G^m) \cup P$.
With probability $\frac{|S|}{k+m}$ return a random k-subset of S, and with probability $1 - \frac{|S|}{k+m}$ return $\emptyset$.

Algorithm 2: Credible Subset

Let us verify that Credible Subset is well-defined, in the sense that $\frac{|S|}{k+m} \leq 1$. Recall that for the purpose of computing $\mathrm{top}_k(G^m)$, ties are broken lexicographically. This implies that, for a given $i \notin \mathrm{top}_k(G^m)$, the only way for i to enter P would be to reduce weights on outgoing edges to some of the top k vertices. It can reduce its outgoing weights to at most m vertices; thus, any vertex that makes it into the top k after reducing weights must have been in the top $k + m$ to begin with, where $k + m$ is defined with respect to the tie-breaking order. We conclude that there cannot be more than m vertices that can enter $\text{top}_k(G^m)$ by reducing their outgoing weights. That is, $|P| \leq m$, and hence

$$|S| = |\text{top}_k(G^m)| + |P| \leq k + m.$$

Theorem 2.3.1. *Credible Subset is an impartial (m, k)-selection mechanism which approximates Vanilla to a factor of $\frac{k}{k+m}$.*

Proof. We first establish impartiality. The mechanism is clearly impartial with respect to vertices $i \in N \setminus S$: for any G and G' that differ only in the weights of outgoing edges from i,

$$\mathbb{P}\left[i \in \mathcal{M}^{cs}(G) \mid i \notin S\right] = 0 = \mathbb{P}\left[i \in \mathcal{M}^{cs}(G') \mid i \notin S\right].$$

The mechanism is also impartial for $i \in S$. Indeed, some k-subset of S is selected with probability $\frac{|S|}{k+m}$. Given that some k-subset of S is selected, the probability that $i \in S$ is selected is $\frac{k}{|S|}$. Thus,

$$\mathbb{P}\left[i \in \mathcal{M}^{cs}(G)\right] = \frac{\mid S \mid}{k+m} \cdot \frac{k}{\mid S \mid} = \frac{k}{k+m}. \tag{2.1}$$

In other words, for two graphs G and G' as above,

$$\mathbb{P}\left[i \in \mathcal{M}^{cs}(G) \mid i \in S\right] = \frac{k}{k+m} = \mathbb{P}\left[i \in \mathcal{M}^{cs}(G') \mid i \in S\right],$$

and we conclude that for all $i \in N$,

$$\mathbb{P}\left[i \in \mathcal{M}^{cs}(G)\right] = \mathbb{P}\left[i \in \mathcal{M}^{cs}(G')\right].$$

Next we establish the approximation guarantees of Credible Subset. Notice that Credible Subset samples from $\mathcal{G}^m$, just as Vanilla does. In addition, for a fixed sampled graph $G^m \sim \mathcal{G}^m$, Vanilla outputs $\text{top}_k(G^m)$. Thus, for every underlying graph G, the approximation ratio given by Credible Subset is

$$\begin{aligned}
&\frac{\text{sc}(\mathcal{M}^{cs}, G)}{\text{sc}(\mathcal{M}^{v}, G)} \\
&\quad = \frac{\sum_{G^m} \mathbb{P}\left[G^m\right] \cdot \sum_{i \in N} \mathbb{P}\left[i \in \mathcal{M}^{cs}(G) \mid G^m\right] \cdot \text{sc}(i, G)}{\sum_{G^m} \mathbb{P}\left[G^m\right] \cdot \sum_{i \in N} \mathbb{P}\left[i \in \mathcal{M}^{v}(G) \mid G^m\right] \cdot \text{sc}(i, G)} \\
&\quad \geq \frac{\sum_{G^m} \mathbb{P}\left[G^m\right] \cdot \sum_{i \in N} \mathbb{I}[i \in \text{top}_k(G^m)] \cdot \frac{k}{k+m} \cdot \text{sc}(i, G)}{\sum_{G^m} \mathbb{P}\left[G^m\right] \cdot \sum_{i \in N} \mathbb{I}[i \in \text{top}_k(G^m)] \cdot \text{sc}(i, G)} \\
&\quad = \frac{k}{k+m},
\end{aligned}$$

where the second transition follows from Equation (2.1), and $\mathbb{I}[E]$ is an indicator variable that takes that value 1 if the event E is true and 0 if E is false. ■

We remark that the mechanism may return subsets of size smaller than k — empty subsets, in fact! Choosing empty subsets is not necessary: the same approximation guarantee can be achieved by defining a finer distribution over subsets preserving that each vertex in S is selected with probability $\frac{k}{k+m}$ (this is the insight that drives the proof of Theorem 2.3.1). We focus on the simpler formulation of the mechanism for ease of exposition, and further discuss this point in Section 2.5.

2.4 Impossibility Results

In Section 2.3 we proved that Credible Subset approximates Vanilla to a factor of $\frac{k}{k+m}$. When $m = o(k)$, this is $1 - o(1)$. But when both k and m are constants, this ratio is bounded away from 1 even when $n \to \infty$. It is natural to wonder, though, if an impartial (m, k)-selection mechanism can approximate Vanilla to a factor of $1 - o(1)$ when k and m are constants and n grows. After all, in this regime the performance of Vanilla will be very poor in the worst case (as G^m gives an extremely incomplete picture of G), so Vanilla becomes easier to approximate. We answer this question in the negative: we show below that the $\frac{k}{k+m}$ ratio is essentially the best possible for impartial (m, k)-selection mechanisms.

Let us start with an informal discussion of a simple upper bound of $\frac{k}{k+1}$ that only assumes that $k \leq m$ (that is, it gives a constant upper bound for $k = O(1)$ even if m grows). Let G be an underlying graph such that

$$w_G(i, j) = \begin{cases} \epsilon & j = 1 \\ 0 & \text{otherwise} \end{cases}$$

Vanilla will certainly select vertex 1. Consider an impartial (m, k)-selection mechanism $\mathcal{M}$, and let $\mathbb{P}[1 \in \mathcal{M}(G)] = p$. Since 1 is the only vertex with nonzero score, the approximation ratio of $\mathcal{M}$ on G is p.

Next, consider the underlying graph G' with weights:

$$w_{G'}(i, j) = \begin{cases} \epsilon & j = 1 \\ 1 & i = 1 \\ 0 & \text{otherwise} \end{cases}$$

For $\epsilon \ll \frac{1}{n-1}$, Vanilla will certainly select k vertices with score 1, so $\text{sc}(\mathcal{M}^v, G') = k$. By impartiality, $\mathbb{P}[1 \in \mathcal{M}(G')] = p$, hence

$$\text{sc}(\mathcal{M}, G') \leq (1 - p)k + p(k - 1 + (n - 1)\epsilon).$$

Since ϵ is arbitrarily small, the approximation ratio is upper-bounded in the limit by

$$\alpha = \min\left\{p, (1 - p) + \frac{p(k - 1)}{k}\right\}.$$

Maximizing α over all $p \in [0, 1]$ gives $p = \frac{k}{k+1}$ as an upper bound on the approximation ratio.

Let us now turn to our more intricate upper bound.

Theorem 2.4.1. *Let $c \in (0, 1/4)$, $k = m^2$, and $m \leq n^c$. Then any impartial mechanism at best $\left(\frac{k}{k+m} + \epsilon(n)\right)$-approximates Vanilla, for $\epsilon(n) = o(1)$.*

We require the following straightforward probabilistic lemma.

Lemma 2.4.2. *Let $c \in (0, 1/4)$. Suppose n^c distinct elements are drawn from a universe of size n uniformly at random and independently. Suppose this experiment is repeated n^c times, and let the selected set in round t be denoted N_t. Then, with high probability, $N_t \cap N_{t'} = \emptyset$, for all $t \neq t'$.*

Proof.

$$\begin{aligned}\mathbb{P}\left[\exists i,j \in \{1,\ldots,n^c\}, i \neq j,\ N_i \cap N_j \neq \emptyset\right] &\leq \binom{n^c}{2} \cdot \mathbb{P}\left[N_1 \cap N_2 \neq \emptyset\right] \\ &\leq \binom{n^c}{2} \cdot \frac{n^{2c}}{n - n^c} \\ &\leq \frac{2n^{4c}}{n} \\ &\xrightarrow{n\to\infty} 0\end{aligned}$$

■

Proof of Theorem 2.4.1. Let $\mathcal{M}$ be an impartial mechanism. Consider a set $X \subset N$ of size m. We will build up a matching μ between X and $N \setminus X$, such that the probability $\mathcal{M}$ samples the edge $(\mu(i), i)$ is small (roughly m/n) for all i. This will imply that $\mathcal{M}$ will have to select i with similar probability on two graphs which differ only in the weight of the edge $(\mu(i), i)$.

We will now select vertices and relabel them, adding them to X as we progress. Select an arbitrary vertex and label it 1. Let $\mu(1) = \operatorname{argmin}_j \mathbb{P}\left[\mathcal{M} \text{ samples } (j,1)\right]$ (the vertex with the smallest probability of $(j,1)$ being sampled by $\mathcal{M}$). Let $q_1 = \mathbb{P}\left[\mathcal{M} \text{ samples } (\mu(1),1)\right]$; note that $q_1 \leq \frac{m}{n-1}$ by a simple averaging argument. Then, for each $i \in \{2,\ldots,m\}$, select another arbitrary vertex and label it i such that $i \notin \{1,\ldots,i-1\} \cup \{\mu(1),\ldots,\mu(i-1)\}$, and let

$$\mu(i) = \operatorname{argmin}_{j \notin \{1,\ldots,i\} \cup \{\mu(1),\ldots,\mu(i-1)\}} \mathbb{P}\left[\mathcal{M} \text{ samples } (j,i)\right],$$

be the vertex such that $(\mu(i), i)$ has the smallest probability of being sampled by $\mathcal{M}$ which is not already part of the matching, and

$$q_i = \mathbb{P}\left[\mathcal{M} \text{ samples } (\mu(i), i)\right]$$

be that probability. Note that $q_i \leq \frac{m}{n-2(i-1)-1}$, else the expected number of edges incident to i would be larger than m.

Now, we construct an underlying graph G that is defined using the following weights:

$$w_G(i,j) = \begin{cases} 1 & i \in X, j \notin X \\ \epsilon \ll \frac{1}{m} & i \notin X, j \in X \\ 0 & \text{otherwise} \end{cases}$$

For each $i \in X$, let the graph G'_i on n vertices be as follows:

$$w_{G'_i}(j,j') = \begin{cases} M \gg 1 & j = \mu(i), j' = i \\ 1 & j \in X, j \neq i, j' \notin X \\ \epsilon \ll \frac{1}{m} & j \notin X, j' \in X, (j,j') \neq (\mu(i), i) \\ 0 & \text{otherwise} \end{cases}$$

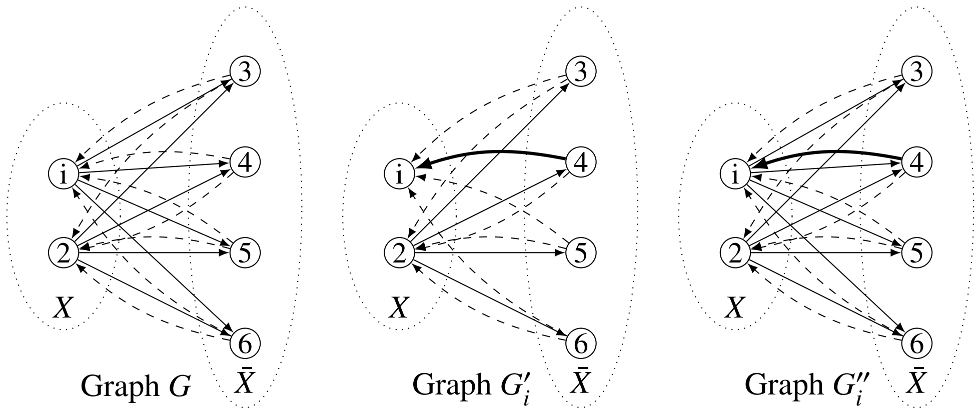

Figure 2.1: Example of the graphs G, G'_i, G''_i where $i = 1$, $\mu(i) = 4$, $X = \{i, 2\}$. Solid lines represent edges of weight 1, dashed lines edges of weight ϵ, and thick lines edges of weight M. Edges not present have weight 0. G and G''_i differ only on the weight of edge $(4, i)$; G''_i and G'_i differ only on the weight of outgoing edges from i.

Notice that G'_i differs from G in two ways: it has one high-weight edge to i, and the outgoing edges from i have weight 0 rather than weight 1. For an illustration, see Figure 2.1.

We begin by showing that

$$\mathrm{sc}(\mathcal{M}^v, G) \geq |X| k(1 - o(1)). \tag{2.2}$$

To prove (2.2), denote the set of vertices adjacent to a set Y in the sampled graph G^m by $\mathcal{N}_{G^m}(Y)$. Notice that the vertices $j \in \mathcal{N}_{G^m}(X)$ have strictly higher sampled ratings than all other vertices in G^m. Moreover, $|\mathcal{N}_{G^m}(X)| \leq k$, so Vanilla will select all $j \in \mathcal{N}_{G^m}(X)$. Thus,

$$\begin{aligned}
\mathrm{sc}(\mathcal{M}^v, G) &= \sum_j \mathbb{P}\left[j \in \mathrm{top}_k(G^m)\right] \mathrm{sc}(j, G) \\
&\geq \sum_{j \notin X} \mathbb{P}\left[j \in \mathrm{top}_k(G^m)\right] \mathrm{sc}(j, G) \\
&\geq \sum_{j \notin X} \mathbb{P}\left[j \in \mathcal{N}_{G^m}(X)\right] \mathrm{sc}(j, G) \\
&\geq |X| \sum_{j \notin X} \mathbb{P}\left[j \in \mathcal{N}_{G^m}(X)\right] = |X| \cdot \mathbb{E}\left[|\mathcal{N}_{G^m}(X)|\right] \\
&\geq |X|(k(1 - o(1))),
\end{aligned}$$

where the final transition follows from Lemma 2.4.2 and the assumption that $c \in (0, 1/4)$ and $m \leq n^c$.

Next, we claim that

$$\mathrm{sc}(\mathcal{M}^v, G'_i) \geq M. \tag{2.3}$$

Let G^m denote the sampled graph. Then, notice that there is a trivial upper bound on the size of $|\mathcal{N}_{G^m}(X \setminus \{i\})|$:

$$|\mathcal{N}_{G^m}(X \setminus \{i\})| \leq m(|X| - 1) = k - m. \tag{2.4}$$

Therefore,

$$\begin{aligned}
\mathrm{sc}(\mathcal{M}^v, G_i') &= \sum_j \mathbb{P}\left[j \in \mathrm{top}_k(G^m)\right] \mathrm{sc}(j, G_i') \\
&\geq M \cdot \mathbb{P}\left[i \in \mathrm{top}_k(G^m)\right] \geq M \cdot \mathbb{P}\left[X \subset \mathrm{top}_k(G^m)\right] \\
&= M \cdot \mathbb{P}\left[|\mathcal{N}_{G^m}(X \setminus \{i\})| \leq k - m\right] = M.
\end{aligned}$$

The fourth transition follows from the observation that the only vertices with nonzero sampled ratings are in $X \cup \mathcal{N}_{G^m}(X \setminus \{i\})$ (which implies Vanilla will select all of them, if there are not more than k), and the final equality comes from from (2.4).

Now, we revisit the impartial mechanism $\mathcal{M}$. We show the probability i is selected by $\mathcal{M}$ in G cannot be too different from the probability i is selected by $\mathcal{M}$ in G_i'. Let $p_i = \mathbb{P}[i \in \mathcal{M}(G)]$. Consider the "intermediate" graph G_i'' such that

$$w_{G_i''}(j, j') = \begin{cases} M \gg 1 & j = \mu(i), j' = i \\ 1 & j \in X, j' \notin X \\ \epsilon \ll \frac{1}{m} & j \notin X, j' \in X, (j, j') \neq (\mu(i), i) \\ 0 & \text{otherwise} \end{cases}$$

That is, G_i'' is the graph G with the added heavy-weight edge to i, or the graph G_i' with the outgoing edges from i set to 1.

Let G^m be the graph sampled by $\mathcal{M}$. If $(\mu(i), i) \notin E(G^m)$, $\mathcal{M}$ cannot distinguish between G and G_i'', and thus must select i with the same probability in those cases. Then, by impartiality, $\mathcal{M}$ must select i with equal (unconditional) probability in G_i', G_i'', since they differ only in the outgoing edges from i.

In more detail, let us denote $p_i = \mathbb{P}[i \in \mathcal{M}(G)]$. We have

$$\begin{aligned}
p_i &= \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \in E(G^m)\right] \mathbb{P}\left[(\mu(i), i) \in E(G^m)\right] + \\
&\quad \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \notin E(G^m)\right] \mathbb{P}\left[(\mu(i), i) \notin E(G^m)\right] \\
&= \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \in E(G^m)\right] \mathbb{P}\left[(\mu(i), i) \in E(G^m)\right] + \\
&\quad \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \notin E(G^m)\right] \left(1 - \mathbb{P}\left[(\mu(i), i) \in E(G^m)\right]\right).
\end{aligned}$$

Then, we explicitly write p_i in terms of q_i:

$$\begin{aligned}
p_i =& \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \in E(G^m)\right] q_i \\
&+ \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \notin E(G^m)\right] (1 - q_i).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left[i \in \mathcal{M}(G_i'') \mid (\mu(i), i) \notin E(G^m)\right] \\
&= \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \notin E(G^m)\right] \\
&= \frac{p_i - q_i \mathbb{P}\left[i \in \mathcal{M}(G) \mid (\mu(i), i) \in E(G^m)\right]}{(1 - q_i)} \leq \frac{p_i}{(1 - q_i)}.
\end{aligned}$$

We can use this inequality to derive an upper bound on the probability that $i \in \mathcal{M}(G_i'')$:

$$\begin{aligned}\mathbb{P}\left[i \in \mathcal{M}(G_i'')\right] &= (1-q_i)\mathbb{P}\left[i \in \mathcal{M}(G_i'')|(\mu(i),i) \notin E(G^m)\right] \\ &\quad + q_i\mathbb{P}\left[i \in \mathcal{M}(G_i'')|(\mu(i),i) \in E(G^m)\right] \\ &\leq (1-q_i)\frac{p_i}{1-q_i} + q_i = p_i + q_i.\end{aligned}$$

Then, by impartiality, $\mathbb{P}\left[i \in \mathcal{M}(G_i')\right] = \mathbb{P}\left[i \in \mathcal{M}(G_i'')\right] \leq p_i + q_i$. It follows that

$$\begin{aligned}&\frac{\mathrm{sc}(\mathcal{M}, G_i')}{\mathrm{sc}(\mathcal{M}^v, G_i')} \\ &\leq \frac{(p_i+q_i)(M+(k-1)(|X|-1)) + (1-p_i-q_i)k(|X|-1)}{M} \\ &= p_i + q_i + \frac{((p_i+q_i)(k-1) + (1-p_i-q_i)k)(|X|-1)}{M} \\ &\leq p_i + q_i + \frac{((p_i+q_i)k + (1-p_i-q_i)k)(|X|-1)}{M} \\ &= p_i + q_i + \frac{k(|X|-1)}{M}\end{aligned} \tag{2.5}$$

where the first inequality comes from a simple calculation of scores, Equation (2.3), and the bound $p_i + q_i \geq \mathbb{P}\left[i \in \mathcal{M}(G_i')\right]$.

On the other hand, let $p = \frac{\sum_{i \in X} p_i}{m}$. Then

$$\begin{aligned}\frac{\mathrm{sc}(\mathcal{M}, G)}{\mathrm{sc}(\mathcal{M}^v, G)} &\leq \frac{(k - \sum_{i \in X} p_i)|X| + \epsilon(n-|X|)\sum_{i \in X} p_i}{(1-o(1))|X|k} \\ &= \frac{(k - \sum_{i \in X} p_i)m + \epsilon(n-m)\sum_{i \in X} p_i}{(1-o(1))mk} \\ &= \frac{(k-pm)m + \epsilon\,(n-m)\,pm}{(1-o(1))mk} \\ &= \frac{\left(1-\frac{pm}{k}\right) + \epsilon\,(n-m)\,\frac{p}{k}}{(1-o(1))} \leq \frac{\left(1-\frac{pm}{k}\right) + \epsilon n\frac{p}{k}}{(1-o(1))}.\end{aligned} \tag{2.6}$$

Now, some $p_i \leq p$, by a simple averaging argument; consider that i. In the construction of μ above, we showed the upper bound $q_i \leq \frac{m}{n-2(i-1)-1}$ on the probability that $(\mu(i), i)$ is sampled by $\mathcal{M}$. Notice that the approximation ratio for $\mathcal{M}$ is at most

$$\begin{aligned}\alpha &\leq \min\left\{p_i + q_i + \frac{k(|X|-1)}{M}, \frac{\left(1-\frac{pm}{k}\right)+\epsilon n\frac{p}{k}}{(1-o(1))}\right\} \\ &\leq \min\left\{p + q_i + \frac{k(|X|-1)}{M}, \frac{\left(1-\frac{pm}{k}\right)+\epsilon n\frac{p}{k}}{(1-o(1))}\right\},\end{aligned}$$

by (2.5) and (2.6). Since ϵ is arbitrarily small, M is arbitrarily large, and $q_i = o(1)$, $\alpha \leq \min\left\{p, \left(1-\frac{pm}{k}\right)\right\} + o(1)$. We derive an upper bound on the minimum by equalizing the two expressions and solving for p, which yields $p = \frac{k}{k+m}$. It follows that $\alpha \leq \frac{k}{k+m} + o(1)$. ■

We remark that Alon et al. [2] prove an upper bound of $\frac{k^2+k-1}{k^2+k}$ for their setting, which is the special case of ours in the regime $m = n - 1$. They do this by creating a graph where all edges have weight 0 except for a cycle of length $k + 1$ of edges of weight 1. One of the vertices in this cycle — call it i — is selected with probability at most $k/(k + 1)$. The upper bound is obtained by reducing the weight on i's outgoing edge to 0. In this new graph, i is still selected with probability at most $\frac{k}{k+1}$ by impartiality, so the mechanism's score is at most $\frac{k}{k+1}k + \frac{1}{k+1}(k - 1)$, whereas the optimal solution (which is equivalent to Vanilla in this regime) achieves score k. It is interesting to note that this argument does not extend to the case of $m \ll n$, because Vanilla is unlikely to see the cycle of valuable edges.

2.5 Discussion

From a practical point of view, with NSF reviewing in mind, Theorem 2.3.1, and Credible Subset itself, are quite compelling. To implement the insights behind Theorem 2.3.1, one should slightly expand the set of eligible winners to include all "credible" proposals (associated with PIs who can manipulate their way into the top k), and randomly choose k among them. This seems justifiable, because it is difficult to distinguish between proposals at the very top.

Our formulation of Credible Subset selects empty subsets with small probability to achieve impartiality. As noted above, we can replace this with a distribution over nonempty subsets. Moreover, in practice, this aspect of the mechanism can perhaps be ignored: PIs would be able to ever-so-slightly increase the probability of their own proposals being accepted by decreasing the number of credible vertices, but the incentives for manipulation under this almost impartial version of Credible Subset would be weak compared to Vanilla.

One of the ways in which the mechanism of Merrifield and Saari [71] differs from our setting is that reviewers are restricted to ranking the proposals. Since Borda count is used to aggregate the rankings, this is equivalent to limiting the reviewers to handing out the ratings $m - 1, m - 2, \ldots, 0$ (exactly one of each) — even though their true ratings may be different. Our ideas readily extend to this setting.

Finally, while we have focused on NSF reviewing in the introduction (and, indeed, this is the real-world setting that motivated us), our results can certainly be applied to conference reviewing. For example, in large conferences such as AAAI and IJCAI, the PC includes hundreds of people — a large fraction of the researchers who actually submit papers to the conference. These conferences are a great fit with our model and results, because: (i) Vanilla is, essentially, the mechanism that is typically used (modulo choosing the m-regular graph in a way that matches reviewers with suitable papers), and (ii) k (the number of papers selected for presentation and publication) is much larger than m (the number of reviews per PC member) — in IJCAI'13, the values were $k = 413$ and $m < 10$, making the Credible Subset Mechanism (or a variation thereof) eminently practical.

Chapter 3

Impartial Peer Ranking

3.1 Introduction

While online outsourcing continues to grow rapidly — with individuals and businesses looking for 24-hour productivity and access to specialized skills — employers are faced with two core challenges during the hiring process. First, online labor markets (OLMs) today require clients to evaluate applications from expert crowdworkers to make hiring decisions. However, assessing workers' applications accurately requires domain expertise; this prevents clients from hiring workers in areas where they lack expertise. Second, even if clients do possess some domain expertise that they can leverage to hire workers, they can still struggle with significant search friction. On online expert outsourcing platforms like Upwork, it takes employers three days to screen, interview, and hire candidates. We therefore focus on this early roadblock in OLMs: the ability to hire crowdworkers.

Our approach is to crowdsource aspects of the hiring process itself. That is, we suggest to have the job candidates evaluate each other and produce a ranking of said candidates that the employer can then review in a more efficient manner. As they should have the necessary domain expertise to make informed decisions and the work would be split among many agents, our aforementioned two core challenges should be largely assuaged. Of course, a main concern is that candidates then have incentive to strategize their responses so that they themselves appear higher in the computed ranking. Thus, as in Chapter 2, we will require impartiality of any amalgamation approach of the peer rankings.

The need for computing a consensus ranking of agents from (incomplete) rankings given by the agents themselves is certainly not limited to our motivating example of outsourcing labor markets. For instance, a similar setting is readily seen in massive open online courses (MOOCs). The sheer size of the student body can often make any non-automated grading intractable. Peer grading is therefore a common route to alleviate such onerous work on the instructors and has been extensively studied (e.g. see [83], [95]). Indeed, any setting where a myriad of agents must be ranked can benefit from our approaches — for example, essay competitions and job promotions. At a smaller scale, even authorship ordering on publications follow the paradigm — though with such few agents our results are not practical in this domain.

3.1.1 Model and Results

Suppose we have a social ranking function f that takes in several (possibly incomplete) rankings of n agents and produces a consensus output ranking. In our setting we wish to have the input rankings be given by the n agents themselves and produce an output ranking that is in some sense, an approximation to what f would produce on the full rankings. Importantly, we wish to do so while maintaining that no agent has any incentive to misreport his ranking. More precisely, no agent should have the ability to affect the probability distribution of his own rank.

Throughout this chapter we will focus on the case $f \in C_2$ where C_2 (as in the style of [46]) is the set of social ranking functions that need only the pairwise comparison matrix. That is, they need only know the fraction of input rankings that rate i before j for all i and j. We will often call such f, *pairwise* ranking functions. This class includes common social ranking functions such as Borda and Kemeny. Given an $f \in C_2$, we then give algorithms that are both impartial and accurate.

In Section 3.3, we introduce the k-PARTITE algorithm, which, in a nutshell, randomly partitions the agents into subsets, builds a probability distribution over the positions of members of one subset based on the opinions of members of other subsets, and then generates a distribution over rankings that is consistent with these distributions over positions. We prove that k-PARTITE is impartial, and, when used in conjunction with any pairwise rule, it provides small *backward error* with respect to that rule: With high probability, k-PARTITE places each agent in the same position that the given pairwise rule $f \in C_2$ would have placed him had the input rankings been slightly perturbed.

In Section 3.4, we present that COMMITTEE algorithm. It randomly chooses a subset of agents, who serve as the eponymous committee. Each committee member is positioned based on the opinions of other committee members, and then all other agents are ordered by the committee. The key idea is that, to avoid conflicts and achieve impartiality, each committee member has slots that are reserved for him, and he is inserted into the reserved slot that most closely matches the aggregate opinion of other committee members. We prove that COMMITTEE provides *mixed error* guarantees with respect to any given pairwise rule. That is, with high probability, COMMITTEE places each agent in a position that is *close* to where the given pairwise rule would have placed him had the input rankings been slightly perturbed. Taking on some *forward error* — a mismatch between the positions — allows for improved backward error compared to k-PARTITE.

In Section 3.5, we apply the COMMITTEE algorithm (after some slight modifications) to the setting that is the special case explored in Chapter 2 — where we wish to find only the top k agents instead of producing a complete ranking. We explore its efficacy on an experimental framework given by [8] and find that this approach outperforms all algorithms in [8] under some parameter choices of k and m (the number of agents each agent reviews).

3.1.2 Related Work

Amalgamating rankings of several agents into one is well studied in voting theory (see [97] for a survey). We are not concerned with the primary facet of this setting in that we are not interested in what input rankings should produce which consensus ranking. Instead, we take such a social ranking function for granted and instead focus on how to approximate its behavior while maintaining that the agents have no incentive to misreport their true rankings — i.e. maintaining impartiality.

As briefly alluded to before, our setting can be readily seen as a generalization of the setting in Chapter 2 where we wished to instead produce only the "top k" of the n agents (where $k < n$). That is, from a ranking of the n individuals, we can certainly truncate the top k of them.

The problem is also heavily intertwined with credit division — dividing up a divisible reward (such as money) among agents who rank each others' contributions. There, the continuous nature due to the divisibility of the reward allows for simple, deterministic approaches (see [33]). Yet, due to the rigidity and discreteness of rankings, we are not admitted such elegance in our setting. Indeed, several papers such as [66] and [13] have demonstrated the difficulty of our problem by demonstrating several impossibility results. We partially circumvent this by smoothing the discreteness via randomization.

3.2 Notation and Definitions

Let us begin with some notation.

1. n: the number of agents (assumed to be ≥ 2).
2. $[k]$ for any $k \in \mathbb{Z}_{>0}$: the set $\{1, \ldots, k\}$ (hence $[n]$ denotes the set of all agents).
3. Σ: the set of all permutations of $[n]$.
4. Σ^n: the set of all agent preference profiles.
5. $\sigma(i)$ for $\sigma \in \Sigma$: the i^{th} agent in the ranking σ.
6. $\sigma^{-1}(i)$ for $\sigma \in \Sigma$: the rank of agent i in the ranking σ.
7. Ω: the set of all pairwise comparison matrices.
8. C_2: the set of deterministic[1] ranking functions that need only the pairwise comparison matrix. We refer to such functions as *pairwise* ranking functions.
9. $\|.\|_\infty$: the L_∞ Frobenius style norm on a matrix. That is: $\|X\|_\infty := \max_{i,j} |X_{i,j}|$.

As a slight abuse of notation, when we refer to the function A we mean either of the following.

- $A : \Sigma^n \to \Omega$: the function that takes in a preference profile and returns the pairwise comparison matrix.
- $A : \Sigma^n \times 2^{[n]} \to \Omega$: the function that takes in a preference profile and set of agents and returns the pairwise comparison matrix when considering only the input rankings of the given set of agents. This is equivalent to the previous representation when the set of agents is $[n]$.

Similarly, when we refer to a deterministic ranking function f we mean any of the following.

- $f : \Sigma^n \to \Sigma$: a function that takes in a preference profile and returns a ranking.
- $f : \Sigma^n \times 2^{[n]} \to \Sigma$: a function that takes in a preference profile and set of agents and returns a ranking using only the input rankings of the given set of agents. This is equivalent to the

[1]Strictly speaking we do not require this determinism, but we assume it as it greatly reduces the opaqueness of the proofs.

previous representation when the set of agents is $[n]$.

- (If $f \in C_2$) $f : \Omega \to \Sigma$: a function that takes in a pairwise comparison matrix and returns a ranking.

Randomized ranking functions are defined similarly (with their range as distributions over Σ).

Our main focus of interest will be ranking functions that achieve impartiality:

Definition 3.2.1. *A (possibly randomized) ranking function f is* impartial *to agent i if he cannot affect the distribution of his own ranking. That is, if for all preference profiles $(\sigma_1, \ldots, \sigma_n) \in \Sigma^n$ there exists no $\tilde{\sigma}_i$ such that $x \neq \tilde{x}$ where*

$$x \in [0,1]^n \text{ and } x_j \text{ is the probability } i \text{ is ranked } j \text{ in } f(\sigma_1, \ldots, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \ldots, \sigma_n), \text{ and}$$
$$\tilde{x} \in [0,1]^n \text{ and } \tilde{x}_j \text{ is the probability } i \text{ is ranked } j \text{ in } f(\sigma_1, \ldots, \sigma_{i-1}, \tilde{\sigma}_i, \sigma_{i+1}, \ldots, \sigma_n).$$

f is said to be impartial if it is impartial for all agents.

Our definition of impartiality may appear to be rather stringent as one may find the following notion more appropriate.

Definition 3.2.2. *A (possibly randomized) ranking function f is* score-impartial *to agent i if he cannot affect the expectation of his value. That is, if for all preference profiles $(\sigma_1, \ldots, \sigma_n) \in \Sigma^n$ and $v \in \mathbb{R}^n$, there exists no $\tilde{\sigma}^i$ such that $\mathbb{E}[x \cdot v] \neq \mathbb{E}[\tilde{x} \cdot v]$ (where x and $\tilde{x}$ are defined as in Definition 3.2.1).*

However, as the next observation demonstrates, this seemingly weaker notion of impartiality is equivalent to ours.

Observation 3.2.3. *If f is a score-impartial ranking function for agent $i \in [n]$, then it is also impartial for agent i.*

Proof. Assume for purposes of contradiction that f is not impartial in that we have $x \neq \tilde{x}$ where x and $\tilde{x}$ are defined as in Definition 3.2.1. Now let $j = \arg\min_k \{k \mid x_k \neq \tilde{x}_k\}$. Then if

$$v_k = \begin{cases} 1 & \text{if } k \leq j \\ 0 & \text{otherwise} \end{cases}$$

we do not have the assumed score-impartiality — a contradiction. ■

Intuitively, this tells us that if we are unaware of an agent's value for each rank, then to ensure score-impartiality for an agent we must enforce regular impartiality as per Definition 3.2.1. In fact a slight alteration to the proof can show the stronger result that even if we knew an agent's value for each rank up to an arbitrarily small (but nonzero) amount of noise, we would still require regular impartiality to achieve score-impartiality.

3.3 k-partite, Forward (i.e. Standard) Error, and Backward Error

Given that our goal is to approximate ranking functions in C_2, our measure of error is critical to the statement of the formal problem. A perhaps standard definition of forward (i.e. standard) error follows.

Definition 3.3.1. *Let $f \in C_2$. A ranking function g (not necessarily $\in C_2$) is said to have* $(\Delta_{prob}, \Delta_{forward})$ forward error *w.r.t. f if for every preference profile $\boldsymbol{\sigma} \in \Sigma^n$ and $i \in [n]$ we have*

$$\frac{\left|f(\boldsymbol{\sigma})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right|}{n} < \Delta_{forward}$$

with probability $\geq 1 - \Delta_{prob}$.

Intuitively, a low amount of forward error implies that every agent i is placed near his correct rank (as determined by f) with high probability. Unfortunately, as the next theorem states, impartial ranking functions cannot approximate this class of ranking functions well under this error measure (let alone the popular Borda rule).

Theorem 3.3.2. *For all $n \geq 2$, there exists no impartial ranking function g that gives a $(1/2, 1/3)$ forward error to the Borda function f (which is in C_2).*

Proof. For $n = 2$ a direct analysis (which we omit) gives the result. Let us therefore consider only the case $n \geq 3$ and assume such a g exists.

Suppose we have the preference profile $\boldsymbol{\sigma}$ where $i \neq 2$ gives the ranking $i-1, \ldots, n, 1, \ldots, i-2$. Note that if agent 2 continued this trend and gave the ranking $1, \ldots, n$ then all agents have the same Borda score.

Now let us consider agent 2 in more depth and the probability vector $p \in [0,1]^n$ where p_i denotes the probability agent 2 will be in position i when g determines the ranking. By impartiality we know that p does not depend on 2's ranking. As p is a probability vector, we must have one of the following.

1. The first $\lfloor n/2 \rfloor$ entries of p sum to $\leq 1/2$.
   In this case, if agent 2 has the ranking $2, 1, 3, 4, 5, \ldots, n$ in $\boldsymbol{\sigma}$, then it is not too difficult to see that $f(\boldsymbol{\sigma})^{-1}(2) = 1$.
2. The last $\lfloor n/2 \rfloor$ entries of p sum to $\leq 1/2$.
   In this case, if agent 2 has the ranking $1, 3, 2, 4, 5, \ldots, n$ in $\boldsymbol{\sigma}$, then it is not too difficult to see that $f(\boldsymbol{\sigma})^{-1}(2) = n$.

In either case, we find that with probability $\geq 1/2$, g will place 2 in a position at least $\lfloor n/2 \rfloor$ places from f's placement. That is, with probability $\geq 1/2$ we have $\left|f(\boldsymbol{\sigma})^{-1}(2) - g(\boldsymbol{\sigma})^{-1}(2)\right| \geq \lfloor n/2 \rfloor \geq n/3$ — giving at best a forward error of $(1/2, 1/3)$. ■

With this impossibility in hand, we set our sights on an alternate error measure in the style of a standard backwards error (à la numerical stability analysis).

Definition 3.3.3. *Let $f \in C_2$. A ranking function g (not necessarily $\in C_2$) is said to have* $(\Delta_{prob}, \Delta_{backward})$ backward error *w.r.t. f if for every preference profile $\boldsymbol{\sigma} \in \Sigma^n$ and $i \in [n]$ there exists a matrix $\tilde{A} \in \Omega$ s.t.*

1. $\left\|A(\boldsymbol{\sigma}) - \tilde{A}\right\|_\infty < \Delta_{backward}$

2. $f(\tilde{A})^{-1}(i) = g(\boldsymbol{\sigma})^{-1}(i)$

with probability $\geq 1 - \Delta_{prob}$.

Intuitively, a low amount of backward error implies that every agent i is placed in a rank that had the agents altered their preference profile slightly, i would be in the correct rank with high probability.

Now consider Algorithm 3: k-PARTITE. As it appears somewhat opaque, it is best to understand its ideas when we assume that all the X_i are the same size (i.e. k divides n and $|X_i| = n/k$) and so also, that the $\gamma_i = k$. Slight but convoluted adjustments are made when this is not the case which for purposes of intuition can be safely ignored.

The crux of the algorithm is the construction of the doubly stochastic Z matrix which in turn is the sum of $Z^{(i)}$ matrices (a weighted sum when k does not divide n). Intuitively, the (a, b) entry of these matrices indicate the probability that a should be placed in position b overall. First, agents are randomly split into k groups of nearly equal size $X_1, \ldots, X_k$ and then each such group separately ranks all n agents producing rankings τ_i. The $Z^{(i)}$ matrix represents X_i's contribution to Z. Specifically, each agent not in X_i is placed in his exact position dictated by τ_i with probability $1/k$ and in all positions that the agents in X_i themselves were assigned to in τ_i with probability $1/(n(k-1))$. This info is encoded as the only non-zero entries in $Z^{(i)}$ — each column then sums to $1/k$ and each row representing an agent in X_i is zero, and all other rows sum to $1/(k-1)$. Z is then computed as the sum of these $Z^{(i)}$ matrices. Owing to its doubly stochastic nature (see Observations 3.7.1 and 3.7.2 for a proof of this property) allows us to use the Birkhoff-von Neumann theorem to sample from this distribution and remain faithful to the probabilities.

```
Let f ∈ C_2 and σ ∈ Σ^n be given as input
Randomly split all n agents into k groups X_1, ..., X_k where |X_i| ∈ {⌊n/k⌋, ⌈n/k⌉}
for i = 1, ..., k do
    τ_i ← f(σ, X_i)
    γ_i ← n/|X_i|
    Let Z^(i) ∈ R^{n×n} where
```

$$Z^{(i)}_{a,b} \leftarrow \begin{cases} \frac{1}{\gamma_i} & \text{if a} \notin X_i \text{ and } \tau_i(b) = a \\ \frac{1}{n(\gamma_i - 1)} & \text{if a} \notin X_i \text{ and } \tau_i(b) \in X_i \\ 0 & \text{otherwise} \end{cases}$$

```
end for
Z ← Σ_{i∈[k]} ((n/|X_i| − 1)/(k − 1)) Z^(i)  (a doubly stochastic matrix — see Observations 3.7.1 and 3.7.2)
Use Birkhoff-von Neumann to sample a ranking σ s.t. a is ranked b with probability Z_{a,b}
return σ
```

Algorithm 3: k-PARTITE

We will see that this algorithm is not only impartial, but also admits limited amounts of backward error. To see this, we will first need the following lemmas.

Lemma 3.3.4. *If k agents $X = \{x_1, \ldots, x_k\}$ are sampled without replacement from the $[n]$ with preference profile $\boldsymbol{\sigma} \in \Sigma^n$, then with probability $< n^2 \exp\left(-\frac{k\epsilon^2}{2}\right)$ we have $\|A(\boldsymbol{\sigma}) - A(\boldsymbol{\sigma}, X)\|_\infty \geq$*

ϵ. That is

$$\mathbb{P}\left[\|A(\boldsymbol{\sigma}) - A(\boldsymbol{\sigma}, X)\|_\infty \geq \epsilon\right] < n^2 \exp\left(-\frac{k\epsilon^2}{2}\right).$$

Proof.

$$\begin{aligned}
\mathbb{P}\left[\|A(\boldsymbol{\sigma}) - A(\boldsymbol{\sigma}, X)\|_\infty \geq \epsilon\right] &\leq \sum_{i<j} \mathbb{P}\left[\left|A(\boldsymbol{\sigma})_{i,j} - A(\boldsymbol{\sigma}, X)_{i,j}\right| \geq \epsilon\right] \\
&\leq \sum_{i<j} 2\exp\left(-\frac{k\epsilon^2}{2}\right) \text{ (by Hoeffding's inequality)} \\
&= 2\binom{n}{2} \exp\left(-\frac{k\epsilon^2}{2}\right) \\
&< n^2 \exp\left(-\frac{k\epsilon^2}{2}\right).
\end{aligned}$$

∎

Lemma 3.3.5. *For every $f \in C_2$, $\boldsymbol{\sigma} \in \Sigma^n$, and $\epsilon > 0$, k-PARTITE gives at most*

$$\left(1 - \left(\frac{k-2}{k-1}\right)\left(1 - n^2 k \exp\left(-\frac{\lfloor n/k \rfloor \epsilon^2}{2}\right)\right), \epsilon\right)$$

backward error to f.

Proof. Observe that

$$\begin{aligned}
\mathbb{P}\left[\exists i \in [k] \text{ s.t. } \|A(\boldsymbol{\sigma}) - A(\boldsymbol{\sigma}, X_i)\|_\infty \geq \epsilon\right] &\leq \sum_{i=1}^{k} \mathbb{P}\left[A(\boldsymbol{\sigma}) - A(\boldsymbol{\sigma}, X_i) \geq \epsilon\right] \\
&\leq \sum_{i=1}^{k} n^2 \exp\left(-\frac{|X_i|\,\epsilon^2}{2}\right) \text{ (by Lemma 3.3.4)} \\
&\leq \sum_{i=1}^{k} n^2 \exp\left(-\frac{\lfloor n/k \rfloor\,\epsilon^2}{2}\right) \\
&= n^2 k \exp\left(-\frac{\lfloor n/k \rfloor\,\epsilon^2}{2}\right).
\end{aligned}$$

Further observe that for any agent a he is placed directly where one of the X_i places him with probability

$$\begin{aligned}
\sum_{i=1 \text{ s.t. } a \notin X_i}^{k} \frac{n/|X_i| - 1}{k-1} \frac{1}{\gamma_i} &= \sum_{i=1 \text{ s.t. } a \notin X_i}^{k} \frac{n/|X_i| - 1}{k-1} \frac{1}{n/|X_i|} \\
&= 1 - \frac{1}{n(k-1)} \sum_{i=1 \text{ s.t. } a \notin X_i}^{k} |X_i| \\
&\geq 1 - \frac{1}{n(k-1)} \sum_{i=1}^{k} |X_i| \\
&= 1 - \frac{1}{n(k-1)} n \\
&= \frac{k-2}{k-1}.
\end{aligned}$$

Therefore we can deduce that k-PARTITE gives at most $\left(1 - \left(\frac{k-2}{k-1}\right)\left(1 - n^2 k \exp\left(-\frac{\lfloor n/k \rfloor \epsilon^2}{2}\right)\right), \epsilon\right)$ backward error as stated. ■

We are now ready for the main result of k-PARTITE.

Theorem 3.3.6. *For every $f \in C_2$ and $\boldsymbol{\sigma} \in \Sigma^n$, k-PARTITE is impartial and if $k = \left\lfloor \left(\frac{n}{\ln n}\right)^{1/3} \right\rfloor$ then gives at most*

$$(4/k, 4/k) \in \left(O\left(\left(\frac{\ln n}{n}\right)^{1/3}\right), O\left(\left(\frac{\ln n}{n}\right)^{1/3}\right)\right)$$

backward error to f.

Proof. That the algorithm is impartial is clear from the inability of any agent i affecting the i^{th} row of the Z matrix. Let us therefore turn our attention to the error bound.

From Lemma 3.3.5 it suffices to show that if we have $\epsilon = 4/k$ we get that

$$1 - \left(\frac{k-2}{k-1}\right)\left(1 - n^2 k \exp\left(-\frac{\lfloor n/k \rfloor \epsilon^2}{2}\right)\right) \leq \frac{4}{k}.$$

Observe that

$$\begin{aligned}
n^2 k \exp\left(-\frac{\lfloor n/k \rfloor \epsilon^2}{2}\right) &\leq n^2 k \exp\left(-\frac{(n/k - 1)\epsilon^2}{2}\right) \\
&= n^2 k \exp\left(-\frac{(n/k - 1)(4/k)^2}{2}\right) \\
&= n^2 k \exp\left(\frac{8}{k^2}\right) \exp\left(-\frac{8n}{k^3}\right) \\
&\leq n^2 \left(n^{1/3}\right) \exp\left(\frac{8}{2^2}\right) \exp\left(-\frac{8n}{\frac{n}{\ln n}}\right) \\
&= e^2 n^{-17/3} \\
&\leq n^{-2}.
\end{aligned}$$

Thus we see that

$$\begin{aligned}
&1-\left(\frac{k-2}{k-1}\right)\left(1-n^2k\exp\left(-\frac{\lfloor n/k\rfloor\,\epsilon^2}{2}\right)\right)\\
&=\frac{1}{k-1}+\left(\frac{k-2}{k-1}\right)\left(n^2k\exp\left(-\frac{\lfloor n/k\rfloor\,\epsilon^2}{2}\right)\right)\\
&\le\frac{1}{k-1}+(1)\left(n^{-2}\right)\\
&\le 2/k+2/k\\
&=4/k.
\end{aligned}$$

■

A natural question is why we insist on what appears to be such a convoluted algorithm instead of a more natural approach such as in Algorithm 4: Naive-bipartite. We will revisit this question in further detail in Section 3.4 after introducing the pertinent concepts.

1: Let $f \in C_2$ and $\boldsymbol{\sigma} \in \Sigma^n$ be given as input
2: Randomly split the n agents into two sets X and Y where $|X| = \left\lceil \frac{n}{2} \right\rceil$ and $|Y| = \left\lfloor \frac{n}{2} \right\rfloor$
3: $\tau_1 \leftarrow f(\boldsymbol{\sigma}, X)$ restricted to the agents only in Y
4: $\tau_2 \leftarrow f(\boldsymbol{\sigma}, Y)$ restricted to the agents only in X
5: σ interlaces τ_1 and τ_2. That is:

$$\sigma(i) \leftarrow \begin{cases} \tau_1\left((i+1)/2\right) & \text{if } i \text{ is odd} \\ \tau_2\left(i/2\right) & \text{if } i \text{ is even} \end{cases}$$

6: **return** σ

Algorithm 4: Naive-bipartite

3.4 Committee and Mixed Error

k-partite demonstrated that there exist impartial mechanisms that accurately imitate any $f \in C_2$. Specifically, we saw that k-partite admits a backward error of $\left(O\left(\left(\frac{\ln n}{n}\right)^{1/3}\right), O\left(\left(\frac{\ln n}{n}\right)^{1/3}\right)\right)$. Observe however that the algorithm/error is somewhat hamstrung by the fact that an agent must be (with high probability) ranked in *exactly* a location that a small perturbation of the input rankings would give. Consequently, in this section we introduce an orthogonal axis of error and allow for an agent to be *close*, instead of exactly at such a location. Consider the following error measure.

Definition 3.4.1. *Let $f \in C_2$. A ranking function g (not necessarily $\in C_2$) is said to have $(\Delta_{prob}, \Delta_{backward}, \Delta_{forward})$* mixed error *w.r.t. f if with probability at least $1 - \Delta_{prob}$, we have that for every preference profile $\boldsymbol{\sigma} \in \Sigma^n$ and $i \in [n]$ there exists a matrix $\tilde{A} \in \Omega$ s.t.*

1. $\left\|A(\boldsymbol{\sigma}) - \tilde{A}\right\|_\infty < \Delta_{backward}$
2. $\frac{\left|f(\tilde{A})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right|}{n} < \Delta_{forward}$

This is in some sense a union of the forward and backward error concepts seen in Section 3.3 and is a standard error concept often seen in scientific computing and numerical stability analysis.

Now consider Algorithm 5: Committee. Intuitively, this algorithm is given a committee of agents $X = \{x_1, \dots, x_k\}$ who determine the entire ranking. First, for each committee member x_i, we determine their rank using only the rankings given by the remaining $k-1$ members. However, as directly placing each committee member in this fashion may cause collisions (i.e. multiple members may be assigned the same rank) we restrict placement of x_i to only the positions $i, i+k, i+2k, \dots$. Specifically, we assign x_i to the closest such position to the rank given to x_i by the other committee members. There are then k of the n ranks assigned. Second, the committee ranks all of the n agents, and the non-committee members are placed in the order ranked by the committee in the remaining $n-k$ slots.

Let $f \in C_2$, $\boldsymbol{\sigma} \in \Sigma^n$, and $X = \{x_1, \dots, x_k\} \subseteq [n]$ be given as input
for $i = 1, \dots, k$ **do**
    $c \leftarrow \arg\min_{j \in \{i, i+k, \dots\}} |j - f(\boldsymbol{\sigma}, X \setminus \{x_i\})|$ (break ties arbitrarily)
    $\sigma(c) \leftarrow x_i$
end for
$\tau \leftarrow f(\boldsymbol{\sigma}, X)$
$j \leftarrow 1$
for $i = 1, \dots, n$ **do**
    if $\tau(i) \notin X$ **then**
        while $\sigma(j)$ is occupied **do**
            $j \leftarrow j + 1$
        end while
        $\sigma(j) \leftarrow \tau(i)$
    end if
end for
return σ

Algorithm 5: Committee

Any statements on the accuracy of the algorithm clearly rely heavily on the make-up of the committee. However, when the committee X is decided completely at random, the algorithm then satisfies the following guarantee.

Lemma 3.4.2. *For every $f \in C_2$, $\boldsymbol{\sigma} \in \Sigma^n$, and $\epsilon > 0$, Committee with a randomly chosen committee $X = \{x_1, \dots, x_k\}$ gives at most $\left(n^2\left(|X|+1\right)\exp\left(-\frac{(|X|-1)\epsilon^2}{2}\right), \epsilon, \frac{|X|+1}{n}\right)$ mixed error to f.*

Proof. First observe that,

$$
\begin{aligned}
\mathbb{P}\left[\exists i \in [k] \text{ s.t. } \|A(\boldsymbol{\sigma}) - A\left(\boldsymbol{\sigma}, X \setminus \{x_i\}\right)\|_\infty \geq \epsilon\right] &\leq \sum_{i=1}^{k} \mathbb{P}\left[\|A(\boldsymbol{\sigma}) - A\left(\boldsymbol{\sigma}, X \setminus \{x_i\}\right)\|_\infty \geq \epsilon\right] \\
&\leq \sum_{i=1}^{k} n^2 \exp\left(-\frac{(k-1)\,\epsilon^2}{2}\right) \\
&= n^2 k \exp\left(-\frac{(k-1)\,\epsilon^2}{2}\right)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbb{P}\left[\|A(\boldsymbol{\sigma}) - A\left(\boldsymbol{\sigma}, X\right)\|_\infty \geq \epsilon\right] &\leq n^2 \exp\left(-\frac{k\epsilon^2}{2}\right) \\
&\leq n^2 \exp\left(-\frac{(k-1)\,\epsilon^2}{2}\right).
\end{aligned}
$$

Thus

$$
\begin{aligned}
&\mathbb{P}\left[\|A(\boldsymbol{\sigma}) - A\left(\boldsymbol{\sigma}, X\right)\|_\infty \geq \epsilon \text{ or } \exists i \in [k] \text{ s.t. } \|A(\boldsymbol{\sigma}) - A\left(\boldsymbol{\sigma}, X \setminus \{x_i\}\right)\|_\infty \geq \epsilon\right] \\
&\quad \leq n^2 (k+1) \exp\left(-\frac{(k-1)\,\epsilon^2}{2}\right).
\end{aligned}
$$

We can therefore conclude that with probability at least $1 - n^2(k+1)\exp\left(-\frac{(k-1)\epsilon^2}{2}\right)$, that the instantiations of $A(\boldsymbol{\sigma}, X), A\left(\boldsymbol{\sigma}, X \setminus \{x_1\}\right), \ldots, A\left(\boldsymbol{\sigma}, X \setminus \{x_k\}\right)$, and $g(\boldsymbol{\sigma})$ where g is Committee, leads us to the following.

- For each agent $i \in X$, if $\tilde{A} = A\left(\boldsymbol{\sigma}, X \setminus \{x_i\}\right)$ then $\left\|A(\boldsymbol{\sigma}) - \tilde{A}\right\|_\infty < \epsilon$. Moreover, as i is placed within $k-1$ positions of where $g(\boldsymbol{\sigma})$ locates him, we have

$$
\left|f(\tilde{A})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right| < k \Rightarrow \frac{\left|f(\tilde{A})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right|}{n} < \frac{k}{n} < \frac{k+1}{n}.
$$

- For each agent $i \notin X$, if $\tilde{A} = A(\boldsymbol{\sigma}, X)$ then $\left\|A(\boldsymbol{\sigma}) - \tilde{A}\right\|_\infty < \epsilon$. Moreover, as i is placed within k positions of where $g(\boldsymbol{\sigma})$ locates him, we have

$$
\left|f(\tilde{A})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right| < k+1 \Rightarrow \frac{\left|f(\tilde{A})^{-1}(i) - g(\boldsymbol{\sigma})^{-1}(i)\right|}{n} < \frac{k+1}{n}.
$$

  Together, these two cases complete the proof (substituting in $|X| = k$).

■

From here, we can see the following.

Theorem 3.4.3. *For every $f \in \mathcal{C}_2$, $\boldsymbol{\sigma} \in \Sigma^n$, and $\epsilon > 0$,* Committee *with a randomly chosen committee X of size $|X| = 1 + \frac{2}{\epsilon^2} \ln\left(\frac{n^3}{\epsilon}\right)$ is impartial and gives at most $(\epsilon, \epsilon, (|X| + 1)/n)$ mixed error to f.*

Proof. The algorithm's impartiality is clear and Lemma 3.4.2 tells us that Committee with a randomly chosen X gives us at most $\left(n^2 (|X| + 1) \exp\left(-\frac{(|X|-1)\epsilon^2}{2}\right), \epsilon, \frac{|X|+1}{n}\right)$ mixed error to f. In particular, as we can safely assume $|X| < n$ (as otherwise the theorem is vacuously true due to the forward error being ≥ 1) we have that we get a mixed error of at most $\left(n^3 \exp\left(-\frac{(|X|-1)\epsilon^2}{2}\right), \epsilon, \frac{|X|+1}{n}\right)$. Setting $\epsilon = n^3 \exp\left(-\frac{(|X|-1)\epsilon^2}{2}\right)$ and solving for $|X|$ then gives the result. ■

In particular, this theorem allows for an incomparable error to Theorem 3.3.6. That is, we can reduce the backwards error so long as we are willing to take on some forward error. For example, setting $|X|$ appropriately gives at most $\left(n^{-2/5}, n^{-2/5}, 2/n + (34/5)n^{-1/5} \ln n\right)$ mixed error.

In addition to Committee's mixed error guarantees, the algorithm is especially useful in the important case of when we know of a small group of agents who together have an accurate/representative view of the rankings. Such a scenario often arises for instance when there is a committee of agents whose opinion is of relatively large importance such as in, for example, conference review processes. Any theoretical guarantees however, would have to incorporate the knowledge of the committee's accuracy.

With our definition of mixed error in hand, let us briefly revisit the question of the seemingly extraneous complexity of k-partite. That is, why we do not consider an algorithm such as Algorithm 4: Naive-bipartite. We demonstrate that this algorithm does not admit tolerable mixed error in general.

Consider $f \in \mathcal{C}_2$ that is defined as:

- Let $X = \{i > 1 | \text{at least one person ranks } i \text{ before } 1\}$
- Return the ranking starting with the agents of X ordered lexicographically, followed by the agents of $[n] \setminus (\{1\} \cup X)$ ordered lexicographically, and agent 1 inserted into the $\lfloor n/3 \rfloor$ position overall (shifting appropriately).

Now consider the preference profile where i reports the ranking $i, 1, 2, \ldots, i-1, i+1, \ldots, n$ (and 1 reports $1, \ldots, n$). Then Naive-bipartite will always return a ranking where agent 1 is placed first or second — as he will always top his set. Clearly this does not give any tolerable mixed error (it does not even admit a mixed error of $(1/2, 1, 1/4)$).

3.5 Committee in the "Top k" Setting

In this section we hark back to the setting of Chapter 2: instead of impartially computing an entire ranking of the agents, we wish to only select the top k of them where each agent only evaluates m other agents. Since the publication of our work, [8] has introduced both a new impartial algorithm which they call ExactDollarPartition (EDP), and an alternative (empirical) metric of success. Specifically, though our positive results of Chapter 2 are the best possible under our metric, [8] rightfully argues the approach is lacking in some respects: problems arise when m is large and our guarantees are in the worst case — leaving much to be desired for more practical scenarios. They therefore give many reasonable randomly produced settings to highlight this problem and demonstrate the EDP algorithm can quite appealingly handle these cases.

Here we add to this by showing that a modification to our Committee algorithm for this "top k" problem, Algorithm 6: Top-k-committee, is quite promising under some of [8]'s experimental settings. Intuitively, the algorithm first randomly chooses a committee $X = \{x_1, \ldots, x_k\}$ and x_i is deemed to be in the top k if $X \setminus \{x_i\}$ believes this to be true. We then go through the remaining agents one at a time and deem whether they should be in the top k by the opinion of agents in X. In either case once this decision is made, they are added to X so that they can assist in categorizing the remaining agents — consequently, the committee grows throughout the algorithm. To further maximize accuracy, the agent on the "chopping block" is chosen to be one whom we are fairly confident on in terms of their membership to the top k. Specifically, we choose the agent ranked highest or lowest by those in X.

Let $f \in C_2$ and $\boldsymbol{\sigma} \in \Sigma^n$ be given as input
Randomly select k agents from $[n]$ — call $X = \{x_1, \ldots, x_k\}$
$A \leftarrow \emptyset$ (the set of agents in the top k)
$B \leftarrow \emptyset$ (the set of agents not in the top k)
for $i = 1, \ldots, k$ **do**
    if $f(\boldsymbol{\sigma}, X \setminus \{x_i\})^{-1}(x_i) \leq k$ **then**
        $A \leftarrow A \cup \{x_i\}$
    else
        $B \leftarrow B \cup \{x_i\}$
    end if
end for
while $|A| < k$ **do**
    if $k - |A| > n - k - |B|$ **then**
        $i \leftarrow \arg\min_{j \in [n] \setminus X} f(\boldsymbol{\sigma}, X)^{-1}(j)$
        $A \leftarrow A \cup \{i\}$
    else
        $i \leftarrow \arg\max_{j \in [n] \setminus X} f(\boldsymbol{\sigma}, X)^{-1}(j)$
        $B \leftarrow B \cup \{i\}$
    end if
    $X \leftarrow X \cup \{i\}$
end while
return A

Algorithm 6: Top-k-committee

Our experimental results are encapsulated in Figure 3.1. We ran the experiments given in [8] for $n = 120$ and various values of m, k, and ϕ. That is, for each setting of these three variables, we ran 100 trials where agents' true rankings were determined by a Mallows model with dispersion parameter ϕ and each agent randomly reviewed m other agents giving Borda scores to each. The true top k agents are those occupying the first k ranks when we amalgamate all agents' complete rankings via the Borda rule. The accuracy of an algorithm is then measured by the percentage of the true top k it was able to select. See [8] for a thorough breakdown of the experimental setup.

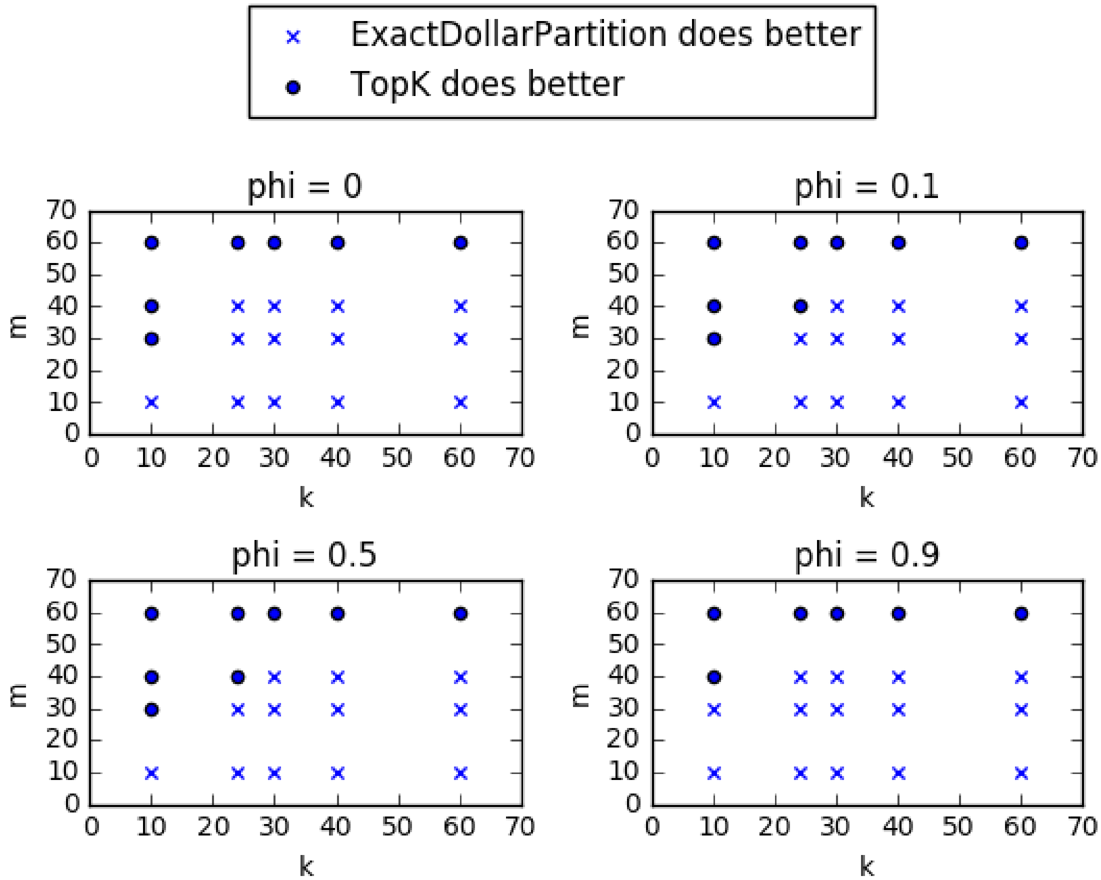

Figure 3.1: m (the number of agents each reviews) versus k (the number of agents to be selected) for various values of ϕ (the dispersion parameter for Mallows model to construct the examples).

As we can see in the figures, Top-k-committee does better when m is large and when k is small. In both cases this is due to the fact that the algorithm has essentially a completely different source of error than EDP. That is, Top-k-committee's struggles are due to it not using the rankings of the agents not in the committee, while in contrast, EDP's is mainly due to a rounding error of a randomization (which is especially prominent when k is small). Use of one over the other therefore is not categorical and requires an analysis of the exact setting.

3.6 Discussion

Our setting of producing a consensus ranking from rankings produced by the ranked themselves is certainly applicable to real-world situations such as previously described in crowd-sourcing websites. We therefore are (at the time of writing) in the process of applying our theoretical ideas in experiments where job applicants on one such crowd-sourcing website do indeed evaluate each other. For the sake of practicality, an important concern of our approaches is the tolerability of our error in these experiments — especially when n is not too large. This error is somewhat abated by a critical implication of impartiality: we can assume that the rankings procured from each agent will be an accurate representation of their beliefs. This is not the case if we were to run some non-impartial algorithm such as simply computing the Borda ranking. Therefore, if we were to run Borda we would almost certainly accrue error due to agents strategizing and reporting untruthful rankings. The use of our impartial algorithms thus induces a trade-off between the approximation error and removing the error caused by strategic agents. This trade-off is one of our primary investigations in our experiments.

From a theoretical standpoint, previous impossibility results, as well as our own on the forward error, necessitate the introduction of different metrics of approximation. We therefore view our (randomized) backward and mixed error definitions as a key contribution of this work. However, a glaring open question is the tightness of our error results. That is, can we do much better than k-PARTITE and COMMITTEE. Furthermore, extending these error ideas to ranking functions outside of the C_2 class and providing accurate algorithms is a logical next step to investigate.

3.7 Proof k-PARTITE (Algorithm 3) is Well-defined

In this section, which we view as an appendix, we prove that the Z matrix given in k-PARTITE is indeed doubly stochastic (and therefore the algorithm is well-defined) via the following two observations.

Observation 3.7.1. *In the algorithm k-PARTITE, the rows of the Z matrix sum to* 1.

Proof. Let us first consider the row sums of the $Z^{(i)}$.

Clearly the a^{th} row sums to 0 if $a \in X_i$. Otherwise for $a \notin X_i$, we find that it sums to

$$\frac{1}{\gamma_i} + |X_i| \frac{1}{n(\gamma_i - 1)} = \frac{1}{\gamma_i} + \frac{1}{\gamma_i(\gamma_i - 1)} = \frac{1}{\gamma_i - 1}.$$

We therefore find that the a^{th} row of Z sums to

$$\begin{aligned}
\sum_{i \in [k]} \left(\frac{n/\left|X_i\right| - 1}{k-1} \right) \left(\text{sum of } a^{th} \text{ of } Z^{(i)} \right) &= \sum_{i \in [k], a \notin X_i} \frac{n/\left|X_i\right| - 1}{k-1} \frac{1}{\gamma_i - 1} \\
&= \sum_{i \in [k], a \notin X_i} \frac{\gamma_i - 1}{k-1} \frac{1}{\gamma_i - 1} \\
&= \sum_{i \in [k], a \notin X_i} \frac{1}{k-1} \\
&= 1.
\end{aligned}$$

■

Observation 3.7.2. *In the algorithm k-PARTITE, the columns of the Z matrix sum to* 1.

Proof. Let us first consider the column sums of the $Z^{(i)}$.

If $\tau_i(b) \notin X_i$ we have that the column has only one non-zero entry with a value of $1/\gamma_i = |X_i|/n$. Otherwise, if $\tau_i(b) \in X_i$, we find that it sums to

$$(n - |X_i|)\,\frac{1}{n(\gamma_i - 1)} = (\gamma_i\,|X_i| - |X_i|)\frac{1}{\gamma_i\,|X_i|\,(\gamma_i - 1)} = \frac{1}{\gamma_i} = \frac{|X_i|}{n}.$$

Therefore all columns sum to $|X_i|/n$.

We therefore find that the b^{th} column of Z sums to

$$\begin{aligned}
\sum_{i\in[k]} \left(\frac{n/\,|X_i| - 1}{k-1}\right)\left(\text{sum of } b^{th} \text{ column of } Z^{(i)}\right) &= \sum_{i\in[k]} \frac{n/\,|X_i| - 1}{k-1}\frac{|X_i|}{n} \\
&= \frac{1}{k-1}\sum_{i\in[k]}\left(1 - \frac{|X_i|}{n}\right) \\
&= \frac{1}{k-1}\left(k - \sum_{i\in[k]}\frac{|X_i|}{n}\right) \\
&= \frac{1}{k-1}\,(k-1) \\
&= 1.
\end{aligned}$$

■

Chapter 4

Cake Cutting with Piecewise Valuations

4.1 Introduction

More than six decades ago, Steinhaus [91] posed the problem of envy-free (EF) cake cutting: when multiple agents have heterogeneous valuations over a divisible cake, how can we divide the cake between the agents so that each agent (weakly) prefers its piece to every other piece? For two agents, the trivial solution is given by the *cut and choose* protocol: one agent divides the cake into two pieces that it values equally, and the other agent chooses its preferred piece.

In 1960, Selfridge and Conway independently proposed an elegant EF cake cutting algorithm for the case of three agents (see, e.g., [20]). The general case continued to tantalize researchers for decades. In a 1988 episode of his PBS show, Sol Garfunkel, the famous mathematical educator, proclaimed it to be one of the greatest problems of 20th Century mathematics. Finally, in 1995—half a century after the problem was posed—Brams and Taylor [20] published an EF cake cutting algorithm for any number of agents.

Our story would end here (somewhat prematurely), if not for a disturbing property of the Brams-Taylor algorithm: although it is guaranteed to terminate in finite time, the number of operations carried out by the protocol can be arbitrarily large, depending on the preferences of the agents. In other words, for every t there are preferences such that the algorithm performs at least t operations. This is a major flaw, especially from the computer scientist's—or parent's, for that matter— point of view; if you start cutting a cake during your child's third birthday party, you would like to finish before he turns eighty!

The problem of designing a *bounded* EF cake cutting algorithm (where the number of operations depends only on the number of agents) remained an open problem until the 2016 work of Aziz and Mackenzie [9] where they gave such an algorithm (albeit requiring an extraordinary number of operations — on the order of $n^{n^{n^{n^{n^n}}}}$). In this chapter we go over a result preceding that of [9] but is still of some interest due to the unwieldy complexity of [9]. The difficulty in the EF cake cutting problem seems to stem from the complexity of agents' preferences, which are generally represented by arbitrary continuous density functions. We therefore ask the following question:

> *Assuming that agents' preferences are restricted, can we design bounded (or even computationally efficient) EF cake cutting algorithms?*

4.1.1 Model and Results

Agents' preferences are represented by valuation functions, which assign values to given pieces of cake. We consider several classes of structured, concisely representable valuations that were originally proposed by Chen et al. [32], and further studied in several recent papers [31, 34, 12, 18]. An agent with a *piecewise uniform* valuation function is interested in a subset of the cake, and simply wants to receive as much of that subset as possible. As an intuitive example where piecewise uniform valuations may arise, suppose that the cake represents access time to a shared backup server; an agent may be able to use as much time as it can get, but only when its computer is idle. Agents with *piecewise constant* valuations are interested in several contiguous pieces of cake, so that each piece is valued uniformly (one crumb is as good as another) but crumbs from different pieces are valued differently. This class is more general than the class of piecewise uniform valuations; in fact, piecewise constant valuations can approximate general valuations to an arbitrary precision. *Piecewise linear* valuations are even more general, and in a sense are almost fully expressive.

To discuss bounded cake cutting algorithms, we also need to define which operations the algorithm is allowed to perform. Here we draw on the well-studied Robertson-Webb model [88, 28, 43, 100, 85], which allows two types of operations: *cut*, which returns a piece of cake with a given value for a given agent, and *eval*, which queries an agent on its value for a given piece. This model is essentially beyond reproach as it is sufficient to simulate all famous discrete cake cutting algorithms.

A natural starting point for our study is the design of EF cake cutting algorithms for the most restricted of the three classes, piecewise uniform valuations. Strikingly though, our first result is that the a bounded EF algorithm for piecewise uniform valuations doubles as a bounded EF algorithm for general valuations requiring the same number of operations. In other words, EF cake cutting under piecewise uniform valuations is already as hard as the general case!

Nevertheless, the three classes of valuation functions have a distinct advantage over general valuations in that they can be parameterized by the number of "pieces" in the word "piecewise". For example, in our backup server setting, this parameter k would represent the number of time intervals in which the agent's computer is idle. Can we design EF algorithms that are tractably bounded by a function of the number of agents n and the number of pieces k? Our answer, which we view as our main result, is the most positive one could hope for: even for piecewise *linear* valuations, we design an EF cake cutting algorithm whose number of queries (in the Robertson-Webb model) is bounded by a *polynomial* function in n and k. We feel that this strong result alleviates the tension around the apparent nonexistence of polynomial time EF cake cutting algorithms for unrestricted valuations, and paints a compelling picture of what makes the problem difficult.

Encouraged by this result, we next ask whether we can strengthen it even further by designing EF algorithms that satisfy additional desirable properties and run in time that is bounded by a function of n and k. It turns out that the answer is negative when the additional property is *strategyproofness*, in the sense that an agent can never gain from manipulating the algorithm. Moreover, we find that there are no finite cake cutting algorithms that satisfy *Pareto-optimality*—a well-known criterion of economic efficiency—even if one does not ask for EF.

4.1.2 Related work

Several papers support our premise that EF cake cutting is extremely difficult. Stromquist [92] showed that there are no bounded algorithms, albeit under the strong assumption that the algorithm must allocate contiguous pieces of cake; his result was strengthened by Deng et al. [37], but they made the same assumption. Procaccia [85] proved an unconditional but rather weak lower bound of $\Omega(n^2)$ in the Robertson-Webb model.

Tractable cake cutting algorithms do exist when the number of agents is very small: the solutions for the cases of two and three agents have long been known. The previously mentioned path-breaking work of Aziz and Mackenzie [9] have demonstrated that the problem is bounded, but the operational complexity is wildly intractable (on the order of $n^{n^{n^{n^{n^{n}}}}}$).

We obtain a strong positive result by restricting the agents' valuations. Alternatively, one can relax the target property itself, by requiring only *approximate* EF, so that envy is bounded by a given ϵ. This goal is implicit in the work of Su [94], and explicit in a paper of Lipton et al. [65], who design an ϵ-EF algorithm whose number of queries (in the Robertson-Webb model) is polynomial in n and $1/\epsilon$.

4.2 Preliminaries

First, some notation.

- $[0,1]$: The cake is modelled as this real interval.
- $\mathcal{N} = \{1,\ldots,n\}$: The set of agents (of which there are n).
- $\forall k \in \mathbb{Z}_{>0} : [k] = \{1,\ldots,k\}$
- $\mathtt{val}_i(X)$: The value agent i has for a piece of cake $X \subseteq [0,1]$. Often for an interval $[x,y] \subseteq [0,1]$ we will abuse notation slightly by writing $\mathtt{val}_i(x,y)$ instead of $\mathtt{val}_i([x,y])$.
- v_i: The *value density function* for agent i whose derivative is undefined or discontinuous at only a finite number of points. That is, for every interval $[x,y] \subseteq [0,1]$, $\mathtt{val}_i(x,y) = \int_x^y v_i(z)dz$. Note that this implies that agent valuations are additive and non-atomic (i.e. $\mathtt{val}_i(x,x) = 0$).

We assume that agent valuations are normalized so that $\mathtt{val}_i(0,1) = 1$. This assumption is without loss of generality as the properties we consider (envy-freeness, Pareto-optimality, strategyproofness) are invariant to scaling the valuation functions by a constant factor.

Following Chen et al. [32], we consider three restricted classes of valuations. We say that an agent has a *piecewise constant* valuation when its value density function is piecewise constant, that is, $[0,1]$ can be partitioned into a finite number of subintervals such that the function is constant on each interval. We define *piecewise linear* valuations similarly. *Piecewise uniform* valuations are a special case of piecewise constant where on each subinterval the density is either some fixed constant $c > 0$, or zero. Piecewise uniform valuations are less expressive than piecewise constant valuations, which are less expressive than piecewise linear valuations. The reader is encouraged to verify that these formal definitions are consistent with their intuitive interpretations above.

An *allocation* $(X_1, \ldots, X_n)$ assigns a piece of cake X_i (that is itself composed of a finite number of subintervals of the cake $[0,1]$) to each agent i such that no two pieces overlap.[1] An allocation is *envy-free (EF)* if $\texttt{val}_i(X_i) \geq \texttt{val}_i\left(X_j\right)$ for all $i, j \in \mathcal{N}$. That is, each agent weakly prefers his own piece to the piece given to any other agent.

In the rest of the chapter, we assume that we are operating in the standard Robertson-Webb query model. That is, the algorithm can only ask agents two types of queries:

1. *Eval query* of the form $\text{eval}(i, x, y)$: asks agent $i \in \mathcal{N}$ for its value for the interval $[x, y]$ — that is, $\text{eval}(i, x, y) = \texttt{val}_i(x, y)$.
2. *Cut query* of the form $\text{cut}(i, x, w)$: asks agent $i \in \mathcal{N}$ the minimum (leftmost) point $y \in [0,1]$ such that $\texttt{val}_i(x, y) = w$ or claims impossibility if no such y exists.

For example, consider the cut and choose protocol; it can be simulated using two queries in the Roberston-Webb model. First, a $\text{cut}(1, 0, 1/2)$ query gives a point w such that the interval $[0, w]$ is worth $1/2$ to agent 1, and hence the value of the complement $[w, 1]$ is also $1/2$. Next, an $\text{eval}(2, 0, w)$ query gives the value of agent 2 for $[0, w]$. If this value is at least $1/2$, we allocate $[0, w]$ to agent 2 and $[w, 1]$ to agent 1, and if it smaller than $1/2$, we switch the allocated pieces.

4.3 General vs. Piecewise Uniform Valuations

Although confining agent valuations to piecewise uniform valuations may seem overly restrictive as a first step, our first result shows that this is not the case. In fact, EF cake cutting for piecewise uniform valuations is just as hard as EF cake cutting for general valuations, in that an algorithm for the former doubles as an algorithm for the latter.

Theorem 4.3.1. *Let $\mathcal{A}$ be an algorithm that computes an EF allocation for n arbitrary piecewise uniform valuations in less than $f(n)$ queries. Then $\mathcal{A}$ can compute EF allocations in less than $f(n)$ queries for general valuation functions.*

Proof. Let $\texttt{val}_1, \ldots, \texttt{val}_n$ be general valuation functions for the agents. Run $\mathcal{A}$ on these valuations. There are two cases to consider.

Case 1: $\mathcal{A}$ terminates in $f(n)$ queries or less, and outputs an allocation $(X_1, \ldots, X_n)$.
We claim that $(X_1, \ldots, X_n)$ is EF with respect to $\texttt{val}_1, \ldots, \texttt{val}_n$. To prove this, we construct *piecewise uniform* valuations U_i based on the queries and responses when $\mathcal{A}$ runs on the $\texttt{val}_i$. The high-level idea is to construct U_i which are equivalent to the $\texttt{val}_i$ in the sense that $\mathcal{A}$ would treat them identically, and then prove envy-freeness of $(X_1, \ldots, X_n)$ for $\texttt{val}_i$ using the envy-freeness of $(X_1, \ldots, X_n)$ for U_i.

Let W_i be the set of all endpoints for all queries and responses associated with agent i when $\mathcal{A}$ runs on valuations $\texttt{val}_i$. That is, if we were to construct W_i iteratively with each query to agent i, then a query and response $b = \text{cut}(i, a, w)$ or $w = \text{eval}(i, a, b)$ would add both a and b to W_i.

Similarly, denote by Y the set of all endpoints for the contiguous intervals in the allocation produced by $\mathcal{A}$. That is, wherever the interval $[0,1]$ is cut to construct a part of the final allocation, we place the cut point in Y.

[1]Technically we allow overlap at a finite number of points since valuations are non-atomic.

Finally, let $Z_i = W_i \cup Y \cup \{0,1\}$ denote an ordered set (using the natural ordering on the reals) and $z_{i,j}$ denote the j^{th} smallest element of Z_i. We are now ready to define the value density function u_i (which pins down the valuation function U_i):

$$u_i(x) = \begin{cases} M_i & \text{if } \exists j \text{ s.t. } x \in \left[z_{i,j+1} - \frac{\mathtt{val}_i(z_{i,j}, z_{i,j+1})}{M_i}, z_{i,j+1}\right] \\ 0 & \text{otherwise,} \end{cases}$$

where $M_i = \max_j \left(\frac{\mathtt{val}_i(z_{i,j}, z_{i,j+1})}{z_{i,j+1} - z_{i,j}}\right)$.

For a given interval $\left[z_{i,j}, z_{i,j+1}\right]$, U_i satisfies two crucial properties:

1. $U_i(z_{i,j}, z_{i,j+1}) = \mathtt{val}_i\left(z_{i,j}, z_{i,j+1}\right)$.
2. If $U_i(z_{i,j}, z_{i,j+1}) > 0$ then there exists $\epsilon > 0$ such that for all $x \in \left[z_{i,j+1} - \epsilon, z_{i,j+1}\right]$, $U_i(x) = M_i$.

In turn, these two properties imply that:

1. $\mathcal{A}$ will ask the same queries and terminate with the same allocation when run on U_i instead of $\mathtt{val}_i$.
2. $U_i(X_i) = \mathtt{val}_i(X_i)$, where X_i is the piece given to agent i in the allocation returned by $\mathcal{A}$.

To see this, note that the first property ensures all eval query responses are the same for both $\mathtt{val}_i$ and U_i. The two properties together similarly ensure all cut query responses are also unaffected; in particular, the second property guarantees that cutting slightly to the left of $z_{i,j+1}$ would give strictly smaller value, hence the leftmost cut point with the same value is still $z_{i,j+1}$. Finally, since Y is included in Z_i, the first property implies that $U_i(X_i) = \mathtt{val}_i(X_i)$ for the allocation returned by $\mathcal{A}$.

Case 2: $\mathcal{A}$ terminates in $f(n)$ or more queries.
Consider the queries asked and responses given after $\mathcal{A}$ has asked $f(n)-1$ queries. Now consider U_i as defined in case 1, except with $Z_i = W_i \cup \{0,1\}$ (we drop the set of points Y since we do not know the allocation that $\mathcal{A}$ will return). U_i satisfies the property that $\mathcal{A}$ will behave the same with respect to U_i and $\mathtt{val}_i$. However, this means that $\mathcal{A}$ will take at least $f(n)$ queries when operating on U_i, and this contradicts the assumption that $\mathcal{A}$ finds an EF allocation in less than $f(n)$ steps for any piecewise uniform valuations. ■

4.4 Bounded Algorithm for Piecewise Linear Valuations

We have shown that restricting agents' valuations to piecewise uniform valuations does not make the problem of finding EF allocations any easier. However, these results rely crucially on the allowance of any number of discontinuities in the value density functions. In the piecewise uniform case, the discontinuities are the points where the density function jumps to a constant c or drops to 0. For piecewise linear valuations, we refer to the endpoints of the subintervals on which the density is linear (hence these are discontinuities of the derivative of the density function rather than of the density function itself.) We use the term *break points* of the value density function.

In this section, we consider what happens when we bound the total number of break points across agents' value density functions. Even when the agent valuations are piecewise linear, and assuming that there are at most k break points across all agents' valuations, we design a cake cutting algorithm that finds an EF allocation with at most $O(n^6 k \ln k)$ queries in the Robertson-Webb model. Before presenting this algorithm we introduce a few definitions and subroutines.

Definition 4.4.1. *A* separating interval *of* $[a,b]$ *is an interval* $[\alpha,\beta] \subseteq [a,b]$ *such that:*

1. *For all* $i \in \mathcal{N}$ *we have* $\mathtt{val}_i(\alpha,\beta) \leq \frac{1}{n}\mathtt{val}_i(a,b)$.
2. *There exists an agent* p *such that* $\mathtt{val}_p(\alpha,\beta) = \frac{1}{n}\mathtt{val}_p(a,b)$.

We refer to p *as the* champion *of the separating interval.*

Given an interval $[a,b]$, we construct a finite cover of separating intervals. That is, we find a finite set $C = \{[\alpha_{i,j},\beta_{i,j}]\}$ (j indexes the separating intervals with champion i) such that $[\alpha_{i,j},\beta_{i,j}]$ is a separating interval of $[a,b]$ with champion i, and for every $x \in [a,b]$, there exists an i and j such that $x \in [\alpha_{i,j},\beta_{i,j}]$. Algorithm 7 produces exactly this.

Cover(a, b)
    $C \leftarrow \emptyset$
    $\alpha \leftarrow a$
    while true **do**
        Let $\beta \leq b$ be the minimal value such that $[\alpha,\beta]$ is worth exactly $\mathtt{val}_i(a,b)/n$ to some agent i.
        if no such β exists **then**
            break
        end if
        $C \leftarrow C \cup [\alpha,\beta]$
        $\alpha \leftarrow \beta$
    end while
    Let α^* be the largest value such that $[\alpha^*,b]$ is worth exactly $\mathtt{val}_i(a,b)(n-1)/n$ to some agent i.
    return $C \cup [\alpha^*,b]$

Algorithm 7: Cover $[a,b]$ by separating intervals

Note that line 4 can be simulated with cut$(i,\alpha,\mathtt{val}_i(a,b)/n)$ queries, and line 11 can be simulated with cut$(i,0,\mathtt{val}_i(a,b)(n-1)/n)$ queries.[2] $\mathtt{val}_i(a,b)$ can be of course obtained via an eval(i,a,b) query.

In each iteration of the loop, we add a separating interval since we know that $[\alpha,\beta]$ has value exactly $\mathtt{val}_i(a,b)/n$ to some agent i, and we choose the smallest possible β. All other agents j have value at most $\mathtt{val}_j(a,b)/n$. What remains to be shown is that all points are in some separating interval. We move from left to right in the loop without skipping over any points, so

[2]Obtaining the largest α^* may require a cut from right to left, but this can be avoided by tweaking line 11 to a more opaque form.

the only possible missing points would be in the case where no viable β exists. However, in this case, $[\alpha, b]$ has value less than $\texttt{val}_i(a,b)/n$ for *all* agents i. Line 11 ensures that we cover $[\alpha, b]$ since $[\alpha^*, b]$ has value at least $\texttt{val}_i(a,b)/n$ for *some* agent i and therefore $\alpha^* < \alpha$.

Definition 4.4.2. *The* sandwich allocation *of* $[a,b]$ *with respect to separating interval* $[\alpha,\beta]$ *with champion* p*, is the allocation where* p *receives* $[\alpha,\beta]$ *and the remaining agents each receive some* X_j *for* $j \in [n-1]$*, where* X_j *is defined as the union of four intervals:*

- $[a + (j-1)\gamma, a + j\gamma]$
- $[\alpha - j\gamma, \alpha - (j-1)\gamma]$
- $[\beta + (j-1)\delta, \beta + j\delta]$
- $[b - j\delta, b - (j-1)\delta]$

where $\gamma = \frac{\alpha - a}{2(n-1)}$ *and* $\delta = \frac{b-\beta}{2(n-1)}$.

In words, the sandwich allocation divides $[a,\alpha]$ to $2(n-1)$ subintervals of equal length, and adds subintervals j and $n-j+1$ (enumerating from left to right) to X_j. A similar process is done for $[\beta, b]$. See Figure 4.1 for an illustration.

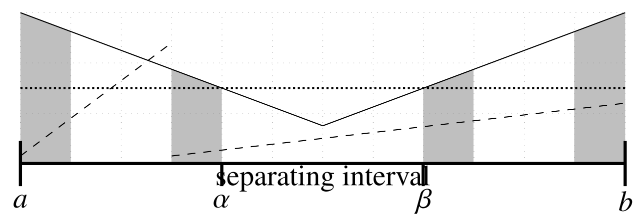

Figure 4.1: A sandwich allocation for agents 1 (the champion), 2, and 3, with dotted, solid, and dashed densities, respectively. The value of agent 1 for the separating interval is $\texttt{val}_1(a,b)/3$. Agent 2 receives the first and fourth quarters of $[a,\alpha]$ and $[\beta,b]$; note that its value for this allocation (the gray area) is $\texttt{val}_2([a,\alpha] \cup [\beta,b])/2$.

We require the following well-known property of piecewise linear valuations [32, 18].

Lemma 4.4.3. *Suppose that an agent has linear value density on interval* $[c,d]$*, and that* $[c,d]$ *is divided into* $2k$ *equal pieces. Let* X_j *for* $j \in [k]$ *denote the piece formed by combining the* j^{th} *piece from the left (moving right) and the* j^{th} *piece from the right (moving left). That is,* X_1 *is the left-most and right-most piece,* X_2 *is the second from the left combined with the second from the right, etc. Then the agent is indifferent between the* X_j*.*

We can now show that if there are no break points outside of the separating interval, then the sandwich allocation is EF (see Figure 4.1).

Lemma 4.4.4. *Let* $[\alpha,\beta]$ *be a separating interval of* $[a,b]$*. Furthermore, suppose that there are no break points in the agents' piecewise linear value density functions on* (a,α) *and* (β,b)*. Then the sandwich allocation of* $[a,b]$ *with separating interval* $[\alpha,\beta]$ *is EF.*

Proof. By assumption there are no break points in (a,α) and (β,b), so each agents' density function is linear on these intervals. Let p denote the champion of the separating interval. Lemma 4.4.3 tells us that the agents are indifferent among the pieces given to agents in $\mathcal{N} \setminus \{p\}$. Agent $i \in \mathcal{N} \setminus \{p\}$ therefore receives value exactly $(\texttt{val}_i(a,b) - \texttt{val}_i(\alpha,\beta))/(n-1) \geq \texttt{val}_i(\alpha,\beta)$ since $\texttt{val}_i(\alpha,\beta) \leq \texttt{val}_i(a,b)/n$ (by the definition of sandwich allocation).

We can now argue that the sandwich allocation is EF. An agent in $\mathcal{N} \setminus \{p\}$ does not envy another agent in the same set since the agent is indifferent among the pieces given to agents in $\mathcal{N} \setminus \{p\}$. These agents also do not envy agent p since they receive value at least $\texttt{val}_i(\alpha, \beta)$ from their pieces. It remains to show that agent p does not envy any other agent. Agent p receives value $\texttt{val}_i(a,b)/n$ from its piece. Since agent p is indifferent among the pieces in $\mathcal{N} \setminus \{p\}$, it receives value $(\texttt{val}_i(a,b) - \texttt{val}_i(a,b)/n)/(n-1) = \texttt{val}_i(a,b)/n$ for these pieces. Agent p is therefore indifferent among all the pieces in the sandwich allocation. ■

We are now ready to give our algorithm that computes an EF allocation for agents with piecewise linear valuations and at most k total break points. At a high-level, our algorithm constructs a cover of separating intervals. For each separating interval in the cover, we attempt to construct an EF allocation. If any of these attempts are successful, we are done. Otherwise, we split $[a,b]$ at every endpoint of an interval in the cover and recurse on these smaller subintervals. Critically, our allocation is chosen so that if we do indeed require a split, then we will separate at least two break points.

EF-Allocate()

1: **return** EF-Allocate(0, 1)

EF-Allocate(a, b)

1: $C \leftarrow$ Cover(a, b)
2: For each $[\alpha, \beta] \in C$, check if the sandwich allocation of $[a,b]$ for separating interval $[\alpha, \beta]$ is EF (for all agents). If it is then return the sandwich allocation.
3: Otherwise let Z be all endpoints of separating intervals in C. Sort Z from smallest to largest, giving points $\{z_1, \ldots, z_m\}$. Recursively call EF-Allocate on intervals formed by consecutive points in Z (i.e., EF-Allocate(z_i, z_{i+1})). Return the allocation formed by joining the allocations returned by each of these recursive calls.

Algorithm 8: EF procedure for piecewise linear valuations

Theorem 4.4.5. *Algorithm 8: EF-Allocate will terminate, produce an EF allocation and require at most $O(n^6 k \ln k)$ queries.*

Proof. As the algorithm can only return by producing an EF allocation or recursing, it will produce an EF allocation if it terminates. Moreover, each iteration of the algorithm will issue a nonzero number of queries (in order to construct a cover and sandwich allocations). Therefore, if we show the number of queries is $O(n^6 k \ln k)$, we will have also shown the algorithm will terminate and produce an EF allocation.

Lemma 4.4.4 tells us that for a separating interval $[\alpha, \beta]$, the sandwich allocation is EF if there are no break points in (a, α) and (β, b), or in other words, all break points are included in $[\alpha, \beta]$[3]. If Algorithm 8 does not find an EF allocation in line 2, then no separating interval in the cover contains all break points. Therefore, recursing on intervals formed by consecutive points in Z (the ordered set of endpoints of separating intervals in C) will separate at least two break

[3]Technically, a break point can appear at a or b but in this case the break point is inconsequential to the valuations on $[a,b]$ and so we ignore it.

points. If there are at most k break points in $[a, b]$, there can be at most $k-1$ break points in any of the intervals recursed on. The base case of this recursion is the case where $k \leq 1$. If $k = 1$, then the sandwich allocation for the separating interval containing the break point will be EF. If $k = 0$, then the sandwich allocation of any separating interval will be EF.

Now let us consider the number of queries our algorithm uses. It is not difficult to see that computing the cover will take at most $n^3+n < 2n^3$ queries and produce a set of at most cardinality $n^2+1 < 2n^2$. Moreover, checking if a sandwich allocation is EF will require at most $4(n-1)n$ queries. This is because the sandwich allocation splits $[a, \alpha]$ and $[\beta, b]$ each into $2(n-1)$ intervals, so there are $4(n-1)$ intervals to ask the agents to evaluate (as there is no need to evaluate $[\alpha, \beta]$). The maximum number of queries $T(n, k)$ can therefore be implicitly given by:

$$\begin{aligned} T(n,k) &\leq 2n^3 + 2n^2(4(n-1)n) + \sum_{j=1}^{2n^2} T(n,k_j) \\ &< 8n^4 + \sum_{j=1}^{2n^2} T(n,k_j), \end{aligned}$$

where due to the property that we split break points, $k_j < k$ for all j, and due to the property that a break point can appear in the interior of only one of the recursively allocated intervals, $\sum_{j=1}^{2n^2} k_j \leq k$[4]. We now show by induction on k that:

$$T(n,k) \leq \begin{cases} 8n^4 & \text{if } k \leq 1 \\ 24n^6 k \ln k & \text{otherwise} \end{cases}$$

As a base case, it is clear the statement holds for $k \leq 1$. We now assume this statement holds true for k, and inductively establish it for $k+1$.

$$\begin{aligned} T(n,k+1) &< 8n^4 + \sum_{j=1}^{2n^2} T(n,k_j) \\ &\leq 8n^4 + \sum_{j\, s.t.\, k_j \leq 1} 8n^4 + \sum_{j\, s.t.\, 1<k_j<k} 24n^6 k_j \ln k_j \\ &\leq 8n^4 + 16n^6 + \sum_{j\, s.t.\, 1<k_j<k} 24n^6 k_j \ln k \\ &< 24n^6 + 24n^6 \ln k \sum_{j\, s.t.\, 1<k_j<k} k_j \\ &\leq 24n^6 + 24n^6 k \ln k \\ &= 24n^6(1 + k\ln k) \\ &\leq 24n^6(k+1)\ln(k+1), \end{aligned}$$

where the last inequality uses the fact that $1 + k\ln k \leq (k+1)\ln(k+1)$ for $k \geq 1$. This is easy to see for $k \geq 2$ since $1, \ln(k) \leq \ln(k+1)$, and we can manually verify the case of $k = 1$. Therefore,

[4]Again, we ignore ignore any inconsequential break points that are at the endpoint of an interval.

the number of queries made by Algorithm 8 is $O(n^6 k \ln k)$. Since the number of queries is bounded, we know that Algorithm 8 terminates (and therefore returns an EF allocation). ■

4.5 Pareto Optimality and Strategyproofness

Theorem 4.4.5 is encouraging, and it seems natural to ask whether one can do better: can we design tractable (in n and the number of break points k) algorithms that achieve allocations that are EF and satisfy additional desirable properties? Unfortunately, for the two prominent properties that we consider in this section, the answer is negative.

The property of *Pareto optimality* is a standard notion of economic efficiency; an allocation $X_1, \ldots, X_n$ is Pareto optimal if there is no other allocation $X'_1, \ldots, X'_n$ such that $\mathtt{val}_i\left(X'_i\right) \geq \mathtt{val}_i\left(X_i\right)$ for all $i \in \mathcal{N}$, and there exists $j \in \mathcal{N}$ such that $\mathtt{val}_i\left(X'_i\right) > \mathtt{val}_i\left(X_i\right)$. It turns out that the Robertson-Webb model does not permit algorithms that produce Pareto optimal allocations — even if other properties such as envy-freeness are not required!

Theorem 4.5.1. *There is no (finite) Pareto optimal cake cutting algorithm for piecewise constant valuations.*

Proof. Suppose $\mathcal{A}$ is a cake cutting algorithm and let all n agents answer queries to $\mathcal{A}$ in a way that is consistent with uniform value density functions (that is, $v_i(x) = 1$ for all $x \in [0, 1]$). Now take any interval $[a, b]$ of non-trivial length that is given to a single agent and does not contain any endpoint of any query. Call the owner of this piece agent p. Change p's value density to be:

$$v_p(x) = \begin{cases} 2 & \text{if } x \in \left[a + \frac{b-a}{4}, \frac{a+b}{2}\right] \\ 0 & \text{if } x \in \left(\frac{a+b}{2}, b - \frac{b-a}{4}\right] \\ 1 & \text{otherwise} \end{cases}$$

Running $\mathcal{A}$ on these new valuations (with p changing to v_p and all other agents unchanged) produces the same allocation as running $\mathcal{A}$ on agents with uniform value density functions as the answers to the eval and cut queries remain unaffected. However, the allocation produced by $\mathcal{A}$ is clearly not Pareto optimal as assigning $\left[\frac{a+b}{2}, b - \frac{b-a}{4}\right]$ to some other agent would raise the receiver's utility without affecting p. ■

Taking a game-theoretic point of view [32], we would like to design cake cutting algorithms that are *strategyproof*, in the sense that agents can never benefit from answering the algorithm's queries untruthfully, regardless of what other agents do. In other words, truthfully answering the algorithm's queries must be a (weakly) *dominant strategy*.

In contrast to Pareto optimality, strategyproofness alone can be achieved easily, e.g., by always allocating the entire cake to a fixed agent. However, if we additionally ask for an algorithm that is EF and bounded (in n and k), we obtain an impossibility result even for piecewise constant valuations.

Theorem 4.5.2. *For any function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and any number of agents $n \geq 2$, there exists no strategyproof and EF cake cutting algorithm on piecewise constant valuations that requires at most $f(n, k)$ queries for every number of break points k.*

Proof. Suppose for sake of contradiction $\mathcal{A}$ is such an algorithm. Now let $\epsilon \in \left(0, 3^{-(n+1)f(n,2n)}\right)$ and define the piecewise uniform valuations U and V as follows. Let $[u_s, u_t]$ be an interval of length $\epsilon/(2n-3)$ and $[v_s, v_t] \subsetneq [u_s, u_t]$ with $v_t - v_s = \epsilon/(2n-2)$.

$$U(x) = \begin{cases} \frac{1}{1-\epsilon/(2n-3)}, & x \notin [u_s, u_t] \\ 0, & x \in [u_s, u_t] \end{cases}$$
$$V(x) = \begin{cases} \frac{1}{1-\epsilon/(2n-2)}, & x \notin [v_s, v_t] \\ 0, & x \in [v_s, v_t] \end{cases}$$

Now we consider $\mathcal{A}$ given the following $n+1$ settings:

- All agents have valuations U
- Agent 1 is uniform, agent 2 has valuation V, all other agents have valuation U
- Agent 2 is uniform, agent 3 has valuation V, all other agents have valuation U

  $\vdots$
- Agent n is uniform, agent 1 has valuation V, all other agents have valuation U

However, as we have not defined these valuation functions precisely, we have some freedom in answering the queries given by $\mathcal{A}$. Specifically, we will answer the queries as to maximize the size of the interval the break points of U (and therefore V) are in. Intuitively, with any single query we can answer such that the interval we know the break points are located in is reduced by at most a factor of 3.

Rigorously, let $I = [I_{left}, I_{right}]$ denote the minimally sized interval we know all break points reside in. Initially, $I = [0, 1]$. Now consider the evaluation query eval(i, a, b) for some i and $a < b$. If $|I \bigcap [a, b]| < \frac{1}{3}|I|$ then assume there are no break points in $[a, b]$ for any agent, and instead they are in the largest contiguous interval of $I \setminus [a, b]$ (which has length at least $\frac{1}{3}|I|$). Otherwise, assume all break points are in $I \bigcap [a, b]$. In this way, no evaluation query can reduce I by more than a factor of 3. Similarly, consider a cut query $b = \text{cut}(i, a, w)$. For sake of simplicity, assume that $a \geq I_{left}$, since otherwise we can assume we are considering the query cut$(i, I_{left}, w - \text{eval}(i, a, I_{left}))$. That is, w is replaced with $w - \text{eval}(i, a, I_{left})$ and a with I_{left}. There are then three cases to consider:

- Case 1: a is on or past the one third mark of I in terms of length (i.e. $a \geq I_{left} + \frac{1}{3}|I|$). Then assume all break points are to the left of a, and so all break points are in $[I_{left}, a]$.
- Case 2: a is left of the one third mark, and $w < \frac{1}{3}\text{eval}(i, I_{left}, I_{right})$.
  Then answer the cut query as if no break points exist in $[I_{left}, a]$, but assume all break points are in fact in $[b, I_{right}]$.
- Case 3: a is left of the one third mark, and $w \geq \frac{1}{3}\text{eval}(i, I_{left}, I_{right})$.
  Then assume all break points are inside $[a, a + \frac{1}{3}|I|]$.

In any case, we are left with again a reduction of I by at most a factor of 3.

As there are at most $(n+1)f(n, 2n)$ queries asked, the condition $\epsilon \in \left(0, 3^{-(n+1)f(n,2n)}\right)$ ensures we can answer all queries in a consistent manner such that after all $n+1$ runs of $\mathcal{A}$, we still have an interval of size at least ϵ in length to place all break points. Call this the ϵ-interval.

We claim that for each of the $n+1$ settings, $\mathcal{A}$ must allocate all of the ϵ-interval to a single agent — ignoring zero-measure subsets given to other agents. Note that this "single agent" is not necessarily the same agent for different settings. Intuitively, this is due to the fact that $\mathcal{A}$ does not identify any properties of any agent valuations in the interval, except for each agent's total value of the interval. Rigorously, outside the ϵ-interval, we may assume U and V are constant, and it is simple to see that by suitably setting U and V we can assign all of the value inside the ϵ-interval to any agent who receives a non-zero-measure subset of said interval. Therefore, two agents cannot both have non-zero-measure subsets of the ϵ-interval as we can suitably set U and V such that one of the concerning agents is envious of another.

Now let p be the agent who is given all of this interval in the setting with all valuations U, and let q be the agent given this interval when agent p is uniform, agent $p+1$ (i.e. agent 1 if $p=n$) is V, and all other agents are U. We claim $p=q$.

There are two cases we must consider. Suppose instead $q \equiv p+1 \pmod{n}$. q has valuation V, and by envy-freeness, must receive a piece of length at least $\left(1-\frac{\epsilon}{2n-2}\right)/n+\frac{\epsilon}{2n-2}$. The first term is $1/n$ of the cake that has non-zero value to q, and the second is the length of $[v_s, v_t]$ — which is worthless to q. Also by envy-freeness, p receives a piece of length $1/n$ and the other $n-2$ agents receives pieces of length at least $\left(1-\frac{\epsilon}{2n-3}\right)/n$. Thus, the total length of cake distributed must be at least:

$$\begin{aligned}\frac{1-\frac{\epsilon}{2n-2}}{n}+\frac{\epsilon}{2n-2}+\frac{1}{n}+(n-2)\frac{1-\frac{\epsilon}{2n-3}}{n} \\ = 1+\frac{\epsilon}{2n(2n-3)} \\ > 1\end{aligned}$$

which is clearly impossible. Similarly, if $q \not\equiv p+1 \pmod{n}$ and $q \neq p$, then the total length of cake distributed also must be > 1. Thus $p=q$ as claimed.

Now consider the following settings given to $\mathcal{A}$:

1. All agents have valuations U
2. p has uniform valuation, all other agents have valuation U
3. p has uniform valuation, $r \equiv p+1 \pmod{n}$ has valuation V, all other agents have valuation U

Since p receives the ϵ-interval in setting 1, p must get a piece of length at least $L=(1-\frac{\epsilon}{2n-3})/n+\frac{\epsilon}{2n-3}$ due to envy-freeness. p must therefore get a piece of length at least L in setting 2, as otherwise p would be strictly better off by misrepresenting his true valuations as U (forming setting 1). This allows us to bound the amount that r can receive in setting 2. We can bound this amount by subtracting the lengths that p must receive in addition to the lengths that agents other than r and p must receive due to envy-freeness:

$$\begin{aligned}&1-\left(\frac{1-\frac{\epsilon}{2n-3}}{n}+\frac{\epsilon}{2n-3}\right)-(n-2)\frac{1-\frac{\epsilon}{2n-3}}{n} \\ =&\frac{1-\frac{\epsilon}{2n-3}}{n}\end{aligned}$$

Therefore, r receives a piece of length at most $\left(1 - \frac{\epsilon}{2n-3}\right)/n$ in setting 2. Finally, consider what occurs if r misrepresents his valuation as V in setting 2 to form setting 3. Since we have shown that p must still get all of the ϵ-interval, r receives a piece of length at least $\left(1 - \frac{\epsilon}{2n-2}\right)/n$. As this piece does not include the ϵ-interval, it is advantageous for r in setting 2 to falsely report a valuation of V — forming setting 3. This contradicts the strategyproofness assumption of $\mathcal{A}$.

■

We can obtain analogs of Theorems 4.5.1 and 4.5.2 for piecewise uniform valuations, at the expense of slightly weakening the algorithm's computational power: for Pareto optimality we require the algorithm to be bounded rather than simply finite, and for strategyproofness and envy-freeness we also require the number of contiguous intervals in the algorithm's allocation to be bounded.

4.6 Discussion

One of the nice features of piecewise uniform, constant, and linear valuations is that they can be concisely represented. For example, a piecewise linear value density function is of the form $f(x) = a_j \cdot x + b_j$ on each subinterval I_j, so we simply need to know a_j and b_j for all $j \leq k+1$, where k is the number of break points (including 0 and 1) of the density function. Given the full, explicit representations it is easy to compute an EF allocation in polynomial time in the size of the representation. Several recent papers [32, 34, 12] leverage this insight by making a powerful assumption: the inputs to the cake cutting algorithm are the agents' full valuation functions.

In contrast, our algorithmic model is based on the Robertson-Webb model. Conceptually, this model captures what we normally think of as cake cutting protocols. The Robertson-Webb model is harder than the full revelation model: any polynomial time algorithm in the former model gives a polynomial time algorithm in the latter model, but the converse is not true. To illustrate this difference, observe that when full piecewise constant valuations are reported, it is straightforward to achieve a Pareto optimal allocation (via a linear program that maximizes social welfare), whereas in the Robertson-Webb model Pareto optimality cannot be achieved (Theorem 4.5.1). In addition, in the full revelation model it is impossible to reason about general valuations—which have an infinite representation—hence in that model there is no analog of our Theorem 4.3.1.

In fact, the main open question of Chen et al. [32] is whether their protocol can be simulated in the Robertson-Webb model. Their main result is a strategyproof and EF algorithm for piecewise uniform valuations that are fully reported to the algorithm. Our results essentially give a negative answer to this question, with one caveat: they also assume that the algorithm may throw away pieces of cake.[5]

The most enigmatic question still remains open: is there a polynomially bounded (in n) EF cake cutting algorithm (i.e., one that can be simulated in the Robertson-Webb model) for general valuations? Our Theorem 4.3.1 may be the key to unlocking this mystery: whether one aims

[5]Counterintuitively, it is known that fair cake cutting algorithms can perform better when allowed to throw away pieces [5].

to prove a possibility or an impossibility result, one can focus on piecewise uniform valuations, which are exactly as hard as the general case.

Chapter 5

Cake Cutting Equilibria

5.1 Introduction

In this chapter we return again to the cake cutting setting: The misleadingly childish metaphor for the challenging and important task of *fairly* dividing a heterogeneous divisible good between multiple agents. In particular, there is a significant amount of AI work on cake cutting [85, 31, 34, 18, 12, 7, 59, 26, 25, 32, 10, 24, 64], which is tightly intertwined with emerging real-world applications of fair division more broadly [50, 61]. Here however, we will expound upon an aspect of the problem that is arguably underappreciated: the game theoretic underpinnings.

Recall (perhaps) the simplest cake cutting protocol of *cut and choose* as described in Chapter 4. The first agent cuts the cake into two pieces that it values equally; the second agent then chooses the piece that it prefers, leaving the first agent with the remaining piece. It is easy to see that this protocol yields a proportional and envy-free allocation (in fact these two notions coincide when there are only two agents). However, taking the game-theoretic point of view, it is immediately apparent that the agents can often do better by disobeying the protocol when they know each other's valuations. For example, in the cut and choose protocol, assume that the first agent only desires a specific small piece of cake, whereas the second agent uniformly values the cake. The first agent can obtain its entire desired piece, instead of just half of it, by carving that piece out.

So how would strategic agents behave when faced with the cut and choose protocol? A standard way of answering this question employs the notion of *Nash equilibrium*: each agent would use a strategy that is a best response to the other agent's strategy. To set up a Nash equilibrium, suppose that the first agent cuts two pieces that the second agent values equally; the second agent selects its more preferred piece, and the one less preferred by the first agent in case of a tie. Clearly, the second agent cannot gain from deviating, as it is selecting a piece that is at least as preferred as the other. As for the first agent, if it makes its preferred piece even bigger, the second agent would choose that piece, making the first agent worse off. Interestingly enough, in this equilibrium the tables are turned; now it is the second agent who is getting exactly half of its value for the whole cake, while the first agent generally gets more. Crucially, the equilibrium outcome is also proportional and envy-free. In other words, even though the agents are strategizing rather than following the protocol, the outcome in equilibrium has the same fairness properties as the "honest" outcome!

With this motivating example in mind, we would like to make general statements regarding the equilibria of cake cutting protocols. We wish to identify a general family of cake cutting protocols — which captures the classic cake cutting protocols — so that each protocol in the family is guaranteed to possess (approximate) equilibria. Moreover, we wish to argue that these equilibrium outcomes are fair. Ultimately, our goal is to be able to reason about the fairness of cake divisions that are obtained as outcomes when agents are presented with a standard cake cutting protocol and behave strategically.

5.1.1 Model and Results

To set the stage for a result that encompasses classic cake cutting protocols, we introduce (in Section 5.2) the class of *generalized cut and choose (GCC)* protocols. A GCC protocol is represented by a tree, where each node is associated with the action of an agent. There are two types of nodes: a *cut node*, which instructs the agent to make a cut between two existing cuts; and a *choose* node, which offers the agent a choice between a collection of pieces that are induced by existing cuts. Moreover, we assume that the progression from a node to one of its children depends only on the relative positions of the cuts (in a sense to be explained formally below). We argue that classic protocols — such as Dubins-Spanier [40], Selfridge-Conway (see [88]), Even-Paz [44], as well as the original cut and choose protocol — are all GCC protocols. We view the definition of the class of GCC protocols as one of our main contributions.

In Section 5.3, we observe that GCC protocols may not have exact Nash equilibria (NE). We then explore two ways of circumventing this issue, which give rise to our two main results.

1. We prove that every GCC protocol has at least one ϵ-NE for every $\epsilon > 0$, in which agents cannot gain more than ϵ by deviating, and ϵ can be chosen to be arbitrarily small. In fact, we establish this result for a stronger equilibrium notion, (approximate) *subgame perfect Nash equilibrium (SPNE)*, which is, intuitively, a strategy profile where the strategies are in NE even if the game starts from an arbitrary point.
2. We slightly augment the class of GCC protocols by giving them the ability to make *informed tie-breaking* decisions that depend on the entire history of play, in cases where multiple cuts are made at the exact same point. While, for some valuation functions of the agents, a GCC protocol may not possess any exact SPNE, we prove that it is always possible to modify the protocol's tie-breaking scheme to obtain SPNE.

In Section 5.4, we observe that for any proportional protocol, the outcome in any ϵ-equilibrium must be an ϵ-proportional division. We conclude that under the classic cake cutting protocols listed above — which are all proportional — strategic behavior preserves the proportionality of the outcome, either approximately, or exactly under informed tie-breaking.

One may wonder, though, whether an analogous result is true with respect to envy-freeness. We give a negative answer, by constructing an envy-inducing SPNE under the Selfridge-Conway protocol, a well-known envy-free protocol for three agents. However, we are able to design a curious GCC protocol in which every NE outcome is a contiguous envy-free allocation and vice versa, that is, the set of NE outcomes coincides with the set of contiguous envy-free allocations. It remains open whether a similar result can be obtained for SPNE instead of NE.

5.1.2 Related Work

The notion of GCC protocols is inspired by the Robertson-Webb [88] model of cake cutting — the concrete complexity model that specifies how a cake cutting protocol may interact with the agents (which formed the basis of our questions in Chapter 4). Their model underpins a significant body of work in theoretical computer science and AI, which focuses on the complexity of achieving different fairness or efficiency notions in cake cutting [42, 43, 100, 36, 7, 85, 59]. In Section 5.2, we briefly review the Roberston-Webb model and explain why it is inappropriate for reasoning about equilibria.

In the context of the strategic aspects of cake cutting, Nicolò and Yu [77] were the first to suggest equilibrium analysis for cake cutting protocols. Focusing exclusively on the case of two agents, they design a specific cake cutting protocol whose unique SPNE outcome is envy-free. And while the original cut and choose protocol also provides this guarantee, it is not "procedural envy free" because the cutter would like to exchange roles with the chooser; the two-agent protocol of Nicoló and Yu aims to solve this difficulty. Brânzei and Miltersen [25] also investigate equilibria in cake cutting, but in contrast to our work they focus on one cake cutting protocol — the Dubins-Spanier protocol — and restrict the space of possible strategies to *threshold strategies*. Under this assumption, they characterize NE outcomes, and in particular they show that in NE the allocation is envy-free. Brânzei and Miltersen also prove the existence of ϵ-equilibria that are ϵ-envy-free; again, this result relies on their strong restriction of the strategy space, and applies to one specific protocol.

Several papers by computer scientists [32, 74, 69] take a mechanism design approach to cake cutting; their goal is to design cake cutting protocols that are *strategyproof*, in the sense that agents can never benefit from manipulating the protocol. This turns out to be an almost impossible task [101, 24]; positive results are obtained by either making extremely strong assumptions (agents' valuations are highly structured), or by employing randomization and significantly weakening the desired properties. In contrast, our main results, given in Section 5.3, deal with strategic outcomes under a large class of cake cutting protocols, and aim to capture well-known protocols; our result of Section 5.4 is a positive result that achieves fairness "only" in equilibrium, but without imposing any restrictions on the agents' valuations.

5.2 The Model

Let us start with some notation and definitions (much of which will be a refresher from Chapter 4 so we do not belabour the points).

- $[0,1]$: The cake is modelled as this real interval.
- $\mathcal{N} = \{1, \ldots, n\}$: The set of agents (of which there are n).
- $\texttt{val}_i(X)$: The value agent i has for a piece of cake $X \subseteq [0,1]$. Often for an interval $[x,y] \subseteq [0,1]$ we will abuse notation slightly by writing $\texttt{val}_i(x,y)$ instead of $\texttt{val}_i([x,y])$. We assume WLOG that $\texttt{val}_i(0,1) = 1$.
- v_i: The *value density function* for agent i whose derivative is undefined or discontinuous at only a finite number of points. That is, for every interval $[x,y] \subseteq [0,1]$, $\texttt{val}_i(x,y) =$

$\int_x^y v_i(z)dz$. Note that this implies that agent valuations are additive and non-atomic (i.e. $\texttt{val}_i(x,x) = 0$).

A *piece of cake* is a finite union of disjoint intervals. We are interested in allocations of disjoint pieces of cake $X_1, \ldots, X_n$, where X_i is the piece that is allocated to agent $i \in \mathcal{N}$. A piece is *contiguous* if it consists of a single interval.

We study two fairness notions. An allocation X is *proportional* if for all $i \in \mathcal{N}$, $\texttt{val}_i(X_i) \geq 1/n$; and *envy-free* if for all $i, j \in \mathcal{N}$, $\texttt{val}_i(X_i) \geq \texttt{val}_i\left(X_j\right)$. Note that envy-freeness implies proportionality.

5.2.1 Generalized Cut and Choose Protocols

Recall from Chapter 4 the standard communication model in cake cutting proposed by Robertson and Webb [88] which restricted interaction between a protocol and the agents to two types of queries:

1. *Eval query* of the form $\text{eval}(i, x, y)$: asks agent $i \in \mathcal{N}$ for its value for the interval $[x, y]$ — that is, $\text{eval}(i, x, y) = \texttt{val}_i(x, y)$.
2. *Cut query* of the form $\text{cut}(i, x, w)$: asks agent $i \in \mathcal{N}$ the minimum (leftmost) point $y \in [0, 1]$ such that $\texttt{val}_i(x, y) = w$ or claims impossibility if no such y exists.

Note however, the communication model does not give much information about the actual implementation of the protocol and what allocations it produces. For example, the protocol could allocate pieces depending on whether a particular cut was made at a specific point (see Algorithm 10).

For this reason, we define a generic class of protocols that are implementable with natural operations, which capture all bounded[1] and discrete cake cutting algorithms, such as cut and choose, Dubins-Spanier, Even-Paz, Successive-Pairs, and Selfridge-Conway (see, e.g., [84]). At a high level, the standard protocols are implemented using a sequence of natural instructions, each of which is either a *Cut* operation, in which some agent is asked to make a cut in a specified region of the cake; or a *Choose* operation, in which some agent is asked to take a piece from a set of already demarcated pieces indicated by the protocol. In addition, every node in the decision tree of the protocol is based exclusively on the execution history and absolute ordering of the cut points, which can be verified with any of the following operators: $<, \leq, =, \geq, >$.

More formally, a *generalized cut and choose (GCC)* protocol is implemented exclusively with the following types of instructions:

- *Cut*: The syntax is "*i Cuts* in *S*", where $S = \{[x_1, y_1], \ldots, [x_m, y_m]\}$ is a set of contiguous pieces (intervals), such that the endpoints of every piece $[x_j, y_j]$ are $0, 1$, or cuts made in the previous steps of the protocol. Agent i can make a cut at any point $z \in [x_j, y_j]$, for some $j \in \{1, \ldots, m\}$.
- *Choose*: The syntax is "*i Chooses* from *S*", where $S = \{[x_1, y_1], \ldots, [x_m, y_m]\}$ is a set of contiguous pieces, such that the endpoints of every piece $[x_j, y_j] \in S$ are 0, 1, or cuts made in the previous steps of the protocol. Agent i can choose any *single* piece $[x_j, y_j]$ from S to keep.

[1] In the sense that the number of operations is upper-bounded by a function that takes the number of agents n as input.

- *If-Else Statements*: The conditions depend on the result of choose queries and the absolute order of all the cut points made in the previous steps.

A GCC protocol uniquely identifies every contiguous piece by the symbolic names of all the cut points contained in it. For example, Algorithm 9 is a GCC protocol. Algorithm 10 is not a GCC protocol, because it verifies that the point where agent 1 made a cut is exactly $1/3$, whereas a GCC protocol can only verify the ordering of the cut points relative to each other and the endpoints of the cake. Note that, unlike in the communication model of Robertson and Web [88], GCC protocols cannot obtain and use information about the valuations of the agents — the allocation is only decided by the agents' *Choose* operations.

agent 1 *Cuts* in $\{[0,1]\}$ // $@x$
agent 1 *Cuts* in $\{[0,1]\}$ // $@y$
agent 1 *Cuts* in $\{[0,1]\}$ // $@z$
if $(x < y < z)$ **then**
  agent 1 *Chooses* from $\{[x,y],[y,z]\}$
end if

Algorithm 9: A GCC protocol. The notation "// $@x$" assigns the symbolic name x to the cut point made by agent 1.

agent 1 *Cuts* in $\{[0,1]\}$ // $@x$
if $\left(x = \frac{1}{3}\right)$ **then**
  agent 1 *Chooses* from $\{[0,x],[x,1]\}$
end if

Algorithm 10: A non-GCC protocol.

As an illustrative example, we now discuss why the discrete version of Dubins-Spanier belongs to the class of GCC protocols — but first we must describe the original protocol. Dubins-Spanier is a proportional (but not envy-free) protocol for n agents, which operates in n rounds. In round 0, each agent makes a mark x_i^1 such that the piece of cake to the left of the mark is worth $1/n$, i.e., $\mathtt{val}_i\left(0, x_i^1\right) = 1/n$. Let i^* be the agent that made the leftmost mark; the protocol allocates the interval $[0, x_{i^*}^1]$ to agent i^*; the allocated interval and satisfied agent are removed. In round t, the same procedure is repeated with the remaining $n-t$ agents and the remaining cake. When there is only one agent left, it receives the remaining cake. To see why the protocol is proportional, first note that in round t the remaining cake is worth at least $1-t/n$ to each remaining agent, due to the additivity of the valuation functions and the fact that the pieces allocated in previous rounds are worth at most $1/n$ to these agents. The agent that made the leftmost mark receives a piece that it values at $1/n$. In round $n-1$, the last agent is left with a piece of cake worth at least $1-(n-1)/n = 1/n$.

The protocol admits a GCC implementation as follows. For the first round, each agent i is required to make a cut in $\{[0,1]\}$ at some point denoted by x_i^1. The agent i^* with the leftmost cut $x_{i^*}^1$ can be determined using *If-Else* statements whose conditions only depend on the ordering of the cut points $x_1^1, \ldots, x_n^1$. Then, agent i^* is asked to choose "any" piece in the singleton set $\{[0, x^1_{i^*}]\}$. The subsequent rounds are similar: at the end of every round the agent that was allocated a piece is removed, and the protocol iterates on the remaining agents and remaining cake. Note that agents are not constrained to follow the protocol, i.e., they can make their marks (in response to cut instructions) wherever they want; nevertheless, an agent can guarantee a piece of value at least $1/n$ by following the Dubins-Spanier protocol, regardless of what other agents do.

While GCC protocols are quite general, a few well-known cake cutting protocols are beyond their reach. For example, the Brams-Taylor [20] protocol is an envy-free protocol for n agents, and although its individual operations are captured by the GCC formalism, the number of operations is not bounded as a function of n (i.e., it may depend on the valuation functions themselves). Its representation as a GCC protocol would therefore be infinitely long. In addition, some cake cutting protocols use *moving knives* (see, e.g., [22]); for example, they can keep track of how an agent's value for a piece changes as the piece smoothly grows larger. These protocols are not discrete, and, in fact, cannot be implemented even in the Robertson-Webb model.

5.2.2 The Game

We study GCC protocols when the agents behave strategically. Specifically, we consider a GCC protocol, coupled with the valuation functions of the agents, as an *extensive-form game of perfect information* (see, e.g., [90]). In such a game, agents execute the *Cut* and *Choose* instructions strategically. Each agent is fully aware of the valuation functions of the other agents and aims to optimize its overall utility for the chosen pieces, given the strategies of other agents.

While the perfect information model may seem restrictive, we note that the same assumption is also made in previous work on equilibria in cake cutting [77, 25]. More importantly, it underpins foundational papers in a variety of areas of microeconomic theory, such as the seminal analysis of the Generalized Second Price (GSP) auction by Edelman et al. [41]. A common justification for the complete information setting, which is becoming increasingly compelling as access to big data becomes pervasive, is that agents can obtain a significant amount of information about each other from historical data.

In more detail, the game can be represented by a tree (called a *game tree*) with Cut and Choose nodes:

- In a *Cut* node defined by "*i cuts in S*", where $S = \{[x_1, y_1], \ldots, [x_m, y_m]\}$, the strategy space of agent i is the set S of points where agent i can make a cut at this step — i.e. $\cup_{z \in S} z$.
- In a *Choose* node defined by "*i chooses from S*", where $S = \{[x_1, y_1], \ldots, [x_m, y_m]\}$, the strategy space is the set $\{1, \ldots, m\}$, i.e., the indices of the pieces that can be chosen by the agent from the set S.

The strategy of an agent defines an action for *each* node of the game tree where it executes a *Cut* or a *Choose* operation. If an agent deviates, the game can follow a completely different branch of the tree, but the outcome will still be well-defined.

The strategies of the agents are in *Nash equilibrium (NE)* if no agent can improve its utility by unilaterally deviating from its current strategy, i.e., by cutting at a different set of points and/or by choosing different pieces. A *subgame perfect Nash equilibrium (SPNE)* is a stronger equilibrium notion, which means that the strategies are in NE in every subtree of the game tree. In other words, even if the game started from an arbitrary node of the game tree, the strategies would still be in NE. An *ϵ-NE* (resp., ϵ-SPNE) is a relaxed solution concept where an agent cannot gain more than ϵ by deviating (resp., by deviating in any subtree).

5.3 Existence of Equilibria

It is well-known that finite extensive-form games of perfect information can be solved using *backward induction*: starting from the leaves and progressing towards the root, at each node the relevant agent chooses an action that maximizes its utility, given the actions that were computed for the node's children. The induced strategies form an SPNE. Unfortunately, although we consider finite GCC protocols, we also need to deal with *Cut* nodes where the action space is infinite, hence naïve backward induction does not apply.

In fact, it turns out that not every GCC protocol admits an exact NE — not to mention SPNE. For example, consider Algorithm 9, and assume that the value density function of agent 1 is strictly positive. Assume there exists a NE where agent 1 cuts at x^*, y^*, z^*, respectively, and chooses the piece $[x^*, y^*]$. If $x^* > 0$, then the agent can improve its utility by making the first cut at $x' = 0$ and choosing the piece $[x', y^*]$, since $\mathtt{val}_1(x', y^*) > \mathtt{val}_1(x^*, y^*)$. Thus, $x^* = 0$. Moreover, it cannot be the case that $y^* = z^*$, since the agent only receives an allocation if $y^* < z^*$. Then, by making the second cut at any $y' \in (y^*, z^*)$, agent 1 can obtain the value $\mathtt{val}_1(0, y') > \mathtt{val}_1(0, y^*)$. It follows that there is no exact NE where the agent chooses the first piece. Similarly, it can be shown that there is no exact NE where the agent chooses the second piece, $[y^*, z^*]$. This illustrates why backward induction does not apply: the maximum value at some *Cut* nodes may not be well defined.

5.3.1 Approximate SPNE

One possible way to circumvent the foregoing example is by saying that agent 1 should be happy to make the cut y very close to z. For instance, if the agent's value is uniformly distributed over the case, cutting at $x = 0, y = 1 - \epsilon, z = 1$ would allow the agent to choose the piece $[x, y]$ with value $1 - \epsilon$; and this is true for any ϵ.

More generally, we have the following theorem.

Theorem 5.3.1. *For any n-agent GCC protocol $\mathcal{P}$ with a bounded number of steps, any n valuation functions $\mathtt{val}_1, \ldots, \mathtt{val}_n$, and any $\epsilon > 0$, the game induced by $\mathcal{P}$ and $\mathtt{val}_1, \ldots, \mathtt{val}_n$ has an ϵ-SPNE.*

The proof of Theorem 5.3.1 is relegated to Section 5.5. In a nutshell, the high-level idea of our proof relies on discretizing the cake — such that every cell in the resulting grid has a very small value for each agent — and computing the optimal outcome on the discretized cake using backward induction. At every cut step of the protocol, the grid is refined by adding a point between every two consecutive points of the grid from the previous cut step. This ensures that any ordering of the cut points that can be enforced by playing on the continuous cake can also be enforced on the discretized instance. Therefore, for the purpose of computing an approximate SPNE, it is sufficient to work with the discretization. We then show that the backward induction outcome from the discrete game gives an ϵ-SPNE on the continuous cake.

5.3.2 Informed Tie-Breaking

Another approach for circumventing the example given at the beginning of the section is to change the *tie-breaking* rule of Algorithm 9, by letting agent 1 choose even if $y = z$ (in which case agent 1 would cut in $x = 0, y = 1, z = 1$, and get the entire cake). Tie-breaking matters: Even the Dubins-Spanier protocol fails to guarantee SPNE existence due to a curious tie-breaking issue [25].

To accommodate more powerful tie-breaking rules, we slightly augment GCC protocols, by extending their ability to compare cuts in case of a tie. Specifically, we can assume without loss of generality that the *If-Else* statements of a GCC protocol are specified only with weak inequalities (as an equality can be specified with two inequalities and a strong inequality via an equality and weak inequality), which involve only pairs of cuts. We consider *informed GCC protocols*, which are capable of using *If-Else* statements of the form "*if* [$x < y$ or ($x = y$ and history of events $\in \mathcal{H}$)] *then*". That is, when cuts are made in the same location and cause a tie in an *If-Else*, the protocol can invoke the power to check the entire history of events that have occurred so far. We can recover the $x < y$ and $x \leq y$ comparisons of "uninformed" GCC protocols by setting $\mathcal{H}$ to be empty or all possible histories, respectively. Importantly, the history can include where cuts were made exactly, and not simply where in relation to each other.

We say that an informed GCC protocol $\mathcal{P}'$ is *equivalent up to tie-breaking* to a GCC protocol $\mathcal{P}$ if they are identical, except that some inequalities in the *If-Else* statements of $\mathcal{P}$ are replaced with informed inequalities in the corresponding *If-Else* statements of $\mathcal{P}'$. That is, the two protocols are possibly different only in cases where two cuts are made at the exact same point.

For example, in Algorithm 9, the statement "*if* $x < y < z$ *then*" can be specified as "*if* $x < y$ *then if* $y < z$ then". We can obtain an informed GCC protocol that is equivalent up to tie-breaking by replacing this statement with "*if* $x < y$ *then if* $y \leq z$ then" (here we are not actually using augmented tie-breaking). In this case, the modified protocol may feel significantly different from the original — but this is an artifact of the extreme simplicity of Algorithm 9. Common cake cutting protocols are more complex, and changing the tie-breaking rule preserves the essence of the protocol.

We are now ready to present our second main result.

Theorem 5.3.2. *For any n-agent GCC protocol $\mathcal{P}$ with a bounded number of steps and any n valuation functions $\mathtt{val}_1, \ldots, \mathtt{val}_n$, there exists an informed GCC protocol $\mathcal{P}'$ that is equivalent to $\mathcal{P}$ up to tie-breaking, such that the game induced by $\mathcal{P}'$ and $\mathtt{val}_1, \ldots, \mathtt{val}_n$ has an SPNE.*

Intuitively, we can view $\mathcal{P}'$ as being "undecided" whenever two cuts are made at the same point, that is, $x = y$: it can adopt either the $x < y$ branch or the $x > y$ branch — there *exists* an appropriate decision. The theorem tells us that for any given valuation functions, we can set these tie-breaking points in a way that guarantees the existence of an SPNE. In this sense, the tie-breaking of the protocol is *informed* by the given valuation functions. Indeed, this interpretation is plausible as we are dealing with a game of perfect information.

The proof of Theorem 5.3.2 is somewhat long, and has been relegated to Section 5.6. This proof is completely different from the proof of Theorem 5.3.1; in particular, it relies on real analysis instead of backward induction on a discretized space. The crux of the proof is the development of an auxiliary notion of *mediated games* (not to be confused with Monderer and

Tennenholtz's *mediated equilibrium* [73]) that may be of independent interest. We show that mediated games always have an SPNE. The actions of the mediator in this SPNE are then reinterpreted as a tie-breaking rule under an informed GCC protocol. In the context of the proof it is worth noting that some papers prove the existence of SPNE in games with infinite action spaces (see, e.g., [52, 54]), but our game does not satisfy the assumptions required therein.

5.4 Fair Equilibria

The existence of equilibria (Theorems 5.3.1 and 5.3.2) gives us a tool for predicting the strategic outcomes of cake cutting protocols. In particular, classic protocols provide fairness guarantees when agents act honestly; but do they provide any fairness guarantees in equilibrium?

We first make a simple yet crucial observation. In a proportional protocol, every agent is guaranteed a value of at least $1/n$ regardless of what the others are doing. Therefore, in every NE (if any) of the protocol, the agent still receives a piece worth at least $1/n$; otherwise it can deviate to the strategy that guarantees it a utility of $1/n$ and do better. Similarly, an ϵ-NE must be ϵ-proportional, i.e., each agent must receive a piece worth at least $1/n - \epsilon$. Hence, classic protocols such as Dubins-Spanier, Even-Paz, and Selfridge-Conway guarantee (approximately) proportional outcomes in any (approximate) NE (and of course this observation carries over to the stronger notion of SPNE).

One may wonder, though, whether the analogous statement for envy-freeness holds; the answer is negative. We demonstrate this via the Selfridge-Conway protocol — a 3-agent envy-free protocol, which is given in its truthful, non-GCC form as Algorithm 11. To see why the protocol is envy free, note that the division of three pieces in steps 4, 5, and 6 is trivially envy free. For the division of the trimmings in step 9, agent i is not envious because it chooses first, and agent j is not envious because it was the one that cut the pieces (presumably, equally according to its value). In contrast, agent 1 may prefer the piece of trimmings that agent i received in step 9, but overall agent 1 cannot envy i, because at best i was able to "reconstruct" one of the three original pieces that was trimmed at step 2, which agent 1 values as much as the untrimmed piece it received in step 6.

1: Agent 1 cuts the cake into three equal parts in the agent's value.
2: Agent 2 trims the most valuable of the three pieces such that there is a tie with the two most valuable pieces.
3: Set aside the trimmings.
4: Agent 3 chooses one of the three pieces to keep.
5: Agent 2 chooses one of the remaining two pieces to keep — with the stipulation that if the trimmed piece is not taken by agent 3, agent 2 must take it.
6: Agent 1 takes the remaining piece.
7: Denote by $i \in \{2, 3\}$ the agent which received the trimmed piece, and $j = \{2, 3\} \setminus \{i\}$.
8: Agent j now cuts the trimmings into three equal parts in the agent's value.
9: Agents i, 1, and j choose one of the three pieces to keep in that order.

Algorithm 11: Selfridge-Conway: an envy-free protocol for three agents.

We construct an example by specifying the valuation functions of the agents and their strategies, and arguing that the strategies are in SPNE. The example will have the property that the first two agents receive utilities of 1 (i.e. the maximum value). Therefore, we can safely assume their play is in equilibrium; this will allow us to define the strategies only on a small part of the game tree. In contrast, agent 3 will deviate from its truthful strategy to gain utility, but in doing so will become envious of agent 1.

In more detail, suppose after agent 2 trims the three pieces we have the following.

- Agent 1 values the first untrimmed piece at 1, and all other pieces and the trimmings at 0.
- Agent 2 values the second untrimmed piece at 1, and all other pieces and the trimmings at 0.
- Agent 3 values the untrimmed pieces at $1/7$ and 0, respectively, the trimmed piece at $1/14$, and the trimmings at $11/14$.

Now further suppose that if agent 3 is to take either untrimmed piece and consequently cut the trimmings (i.e. take on the role of j in the protocol), then the first two agents always take the pieces most valuable to agent 3. Thus, if agent 3 takes either untrimmed piece it will achieve a utility of at most $1/7 + (11/14)(1/3) = 17/42$ by taking the first untrimmed piece, and then cutting the trimmings into three equal parts. On the other hand, if agent 3 takes the trimmed piece of worth $1/14$, agent 2 cuts the trimmings into three parts such that one of the pieces is worth 0 to agent 3, and the other two are equivalent in value (i.e. they have values $(11/14)(1/2) = 11/28$). Agents 1 and 3 take these two pieces. Thus, in this scenario, agent 3 receives a utility of $1/14 + 11/28 = 13/28$ which is strictly better than the utility of $17/42$. Agent 3 will therefore choose to take the trimmed piece. However, in this outcome agent 1, from the point of view of agent 3, receives a piece worth $1/7 + 11/28 = 15/28$ and therefore agent 3 will indeed be envious.

The foregoing example shows that envy-freeness is not guaranteed when agents strategize, and so it is difficult to produce envy-free allocations when agents play to maximize their utility. A natural question to ask, therefore, is whether there are any GCC protocols such that all SPNE are envy-free, and existence of SPNE is guaranteed. This remains an open question, but we do give an affirmative answer for the weaker solution concept of NE in the following theorem, whose proof appears in Section 5.7.

Theorem 5.4.1. *There exists a GCC protocol $\mathcal{P}$ such that on every cake cutting instance with strictly positive valuation functions $\texttt{val}_1, \ldots, \texttt{val}_n$, an allocation X is the outcome of a NE of the game induced by $\mathcal{P}$ and $\texttt{val}_1, \ldots, \texttt{val}_n$ if and only if X is an envy-free contiguous allocation that contains the entire cake.*

Crucially, an envy-free contiguous allocation is guaranteed to exist [93], hence the set of NE of protocol $\mathcal{P}$ is nonempty.

Theorem 5.4.1 is a positive result *à la* implementation theory (see, e.g., [68]), which aims to construct games where the NE outcomes coincide with a given specification of acceptable outcomes for each constellation of agents' preferences (known as a *social choice correspondence*). Our construction guarantees that the NE outcomes coincide with (contiguous) envy-free allocations, that is, in this case the envy-freeness criterion specifies which outcomes are acceptable.

That said, the protocol $\mathcal{P}$ constructed in the proof of Theorem 5.4.1 is impractical: its Nash equilibria are unlikely to arise in practice. This further motivates efforts to find an analogous result for SPNE. If such a result is indeed feasible, a broader, challenging open question would be to characterize GCC protocols that give rise to envy-free SPNE, or at least provide a sufficient condition (on the protocol) for the existence of such equilibria.

5.5 Proof of Theorem 5.3.1

Let $\epsilon > 0$, and let $f(n)$ be an upper bound on the number of operations (i.e., on the height of the game tree) of the protocol. Define a grid, $\mathcal{G}_1$, such that every cell on the grid is worth at most $\frac{\epsilon}{2f(n)^2}$ to each agent. For every n, let K denote the maximum number of cut operations, where $0 \leq K \leq f(n)$. For each $i \in \{1, \ldots, K\}$, we define the grid $\mathcal{G}_i$ so that the following properties are satisfied:

- The grids are nested, i.e., $\{0, 1\} \subset \mathcal{G}_1 \subset \mathcal{G}_2 \subset \ldots \subset \mathcal{G}_K$.
- There exists a unique point $z \in \mathcal{G}_{i+1}$ between any two consecutive points $x, y \in \mathcal{G}_i$, such that $x < z < y$ and $z \notin \mathcal{G}_i$, for every $i \in \{1, \ldots, K-1\}$.
- Each cell on $\mathcal{G}_i$ is worth at most $\frac{\epsilon}{2f(n)^2}$ to any agent, for all $i \in \{1, \ldots, K\}$.

Having defined the grids, we compute the backward induction outcome on the discretized cake, where the i-th *Cut* operation can only be made on the grid $\mathcal{G}_i$. We will show that this outcome is an ϵ-SPNE, even though agents could deviate by cutting anywhere on the cake. On the continuous cake, the agents play a perturbed version of the idealized game from the grid $\mathcal{G}$, but maintain a mapping between the perturbed game and the idealized version throughout the execution of the protocol, such that each cut point from the continuous cake is mapped to a grid point that approximates it within a very small (additive) error. Thus when determining the next action, the agents use the idealized grid as a reference. The order of the cuts is the same in the ideal and perturbed game, however the values of the pieces may differ by at most $\epsilon / f(n)$.

We start with the following useful lemma. (For ease of exposition, in the following we refer to $[x, y]$ as the segment between points x and y and $\mathtt{val}_i(x, y)$ as the value of this segment to agent i, regardless of whether $x < y$ or $y \leq x$.)

Lemma 5.5.1. *Given a sequence of cut points $x_1, \ldots, x_k$ and nested grids $\mathcal{G}_1 \subset \ldots \subset \mathcal{G}_k$ with cells worth at most $\frac{\epsilon}{4f(n)^2}$ to each agent, there exists a map $\mathcal{M} : \{x_1, \ldots, x_k\} \to \mathcal{G}_k$ such that:*

1. *For each $i \in \{1, \ldots, k\}$, $\mathcal{M}(x_i) \in \mathcal{G}_i$.*
2. *The map $\mathcal{M}$ is order-preserving. Formally, for all $i, j \in \{1, \ldots, k\}$, $x_i < x_j \iff \mathcal{M}(x_i) < \mathcal{M}(x_j)$ and $x_i = x_j \iff \mathcal{M}(x_i) = \mathcal{M}(x_j)$.*
3. *The piece $[x_i, \mathcal{M}(x_i)]$ is "small", that is: $\mathtt{val}_l(x_i, \mathcal{M}(x_i)) \leq \frac{k\epsilon}{2f(n)^2}$, for each agent $l \in \mathcal{N}$.*
4. *For each $i \in \{1, \ldots, k\}$, $\mathcal{M}(x_i) = 0 \iff x_i = 0$ and $\mathcal{M}(x_i) = 1 \iff x_i = 1$.*

Proof. We prove the statement by induction on the number of cut points k.

Base case: We consider a few cases. If $x_1 \in \mathcal{G}_1$, then define $\mathcal{M}(x_1) := x_1$. Otherwise, let $R(x_1) \in \mathcal{G}_1$ be the leftmost point on the grid $\mathcal{G}_1$ to the right of x_1. If $R(x_1) \neq 1$, define $\mathcal{M}(x_1) := R(x_1)$; else, let $L(x_1)$ denote the rightmost point on $\mathcal{G}_1$ strictly to the left of 1 and define $\mathcal{M}(x_1) := L(x_1)$. To verify the properties of the lemma, note that:

1. $\mathcal{M}(x_1) \in \mathcal{G}_1$.

2. The map $\mathcal{M}$ is order-preserving since there is only one point.
3. $\texttt{val}_l(x_1, \mathcal{M}(x_1)) \leq \frac{\epsilon}{2f(n)^2}$ for each agent $l \in \mathcal{N}$ since the grid $\mathcal{G}_1$ has (by construction) the property that each cell is worth at most $\frac{\epsilon}{2f(n)^2}$ to each agent, and the interval $[x_1, \mathcal{M}(x_1)]$ is contained in a cell.
4. By the definition of L and R, the only time $\mathcal{M}(x_1) = 0$ (resp. $\mathcal{M}(x_1) = 1$) is when $x_1 = 0$ (resp. $x_1 = 1$).

Induction hypothesis: Assume that a map $\mathcal{M}$ with the required properties exists for any sequence of $k-1$ cut points.

Induction step: Consider any sequence of k cut points $x_1, \ldots, x_k$. By the induction hypothesis, we can map each cut point x_i to a grid representative $\mathcal{M}(x_i) \in \mathcal{G}_i$, for all $i \in \{1, \ldots, k-1\}$, in a way that preserves properties 1–4. We claim that the map $\mathcal{M}$ on the points $x_1, \ldots, x_{k-1}$ can be extended to the k-th point, x_k, such that the entire sequence $\mathcal{M}(x_1), \ldots, \mathcal{M}(x_k)$ satisfies the requirements of the lemma. We consider four exhaustive cases.

1. $x_k \in \{0, 1\}$.
   Then define $\mathcal{M}(x_k) = x_k$.
2. There exists $i \in \{1, \ldots, k-1\}$ such that $x_k = x_i$.
   Then define $\mathcal{M}(x_k) := \mathcal{M}(x_i)$.
3. There exists $i \in \{1, \ldots, k-1\}$ such that $x_i < x_k$, but $\mathcal{M}(x_i) \geq x_k$.
   Let x_j be the rightmost cut such that $x_j < x_k$; because $\mathcal{M}$ is order-preserving, it holds that $\mathcal{M}(x_j) \geq x_k$. Let $R(\mathcal{M}(x_j))$ be the leftmost point on $\mathcal{G}_k$ strictly to the right of $\mathcal{M}(x_j)$, and set $\mathcal{M}(x_k) := R(\mathcal{M}(x_j))$.
   Now let us check the conditions. Condition (1) holds by definition. Condition (2) holds because $\mathcal{M}(x_k) > \mathcal{M}(x_j)$, and for every t such that $x_t > x_k$, $\mathcal{M}(x_t) > \mathcal{M}(x_j)$ and $\mathcal{M}(x_t) \in \mathcal{G}_{k-1}$, whereas $\mathcal{M}(x_k)$ uses a "new" point of $\mathcal{G}_k \setminus G_{k-1}$ that is closer to $\mathcal{M}(x_j)$. For condition (3), we have that for every $l \in \mathcal{N}$,
   $$\begin{aligned} &\texttt{val}_l(x_k, \mathcal{M}(x_k)) \\ &\leq \texttt{val}_l\left(x_j, \mathcal{M}(x_k)\right) \\ &= \texttt{val}_l\left(x_j, \mathcal{M}(x_j)\right) + \texttt{val}_l\left(\mathcal{M}(x_j), \mathcal{M}(x_k)\right) \\ &\leq \frac{(k-1)\epsilon}{2f(n)^2} + \frac{\epsilon}{2f(n)^2} \\ &= \frac{k\epsilon}{2f(n)^2}, \end{aligned}$$
   where the third transition follows from the induction assumption. Condition (4) holds vacuously in this case.
4. There exists $i \in \{1, \ldots, k-1\}$ such that $x_i > x_k$, but $\mathcal{M}(x_i) \leq x_k$.
   This case is symmetric to the previous case so we omit its analysis.
5. For every x_i such that $x_i < x_k$, $\mathcal{M}(x_i) < x_k$, and for every x_j such that $x_j > x_k$, $\mathcal{M}(x_j) > x_k$ (and $x_k \notin \{0, 1\}$).
   Let x_i and x_j be the rightmost and leftmost such cuts, respectively; without loss of generality they exist, otherwise our task is even easier.

Let $R(x_k)$ be the leftmost point in $\mathcal{G}_k$ such that $R(x_k) \geq x_k$, and let $L(x_k)$ be the rightmost point in $\mathcal{G}_k$ such that $L(x_k) \leq x_k$. Assume first that $\mathcal{M}(x_j) > R(x_k)$; then set $\mathcal{M}(x_k) := R(x_k)$. This choice obviously satisfies the four conditions, similarly to the base of the induction.

Otherwise, $R(x_k) = \mathcal{M}(x_j)$ (notice that it cannot be the case that $R(x_k) > \mathcal{M}(x_k)$); then set $\mathcal{M}(x_k) := L(x_k)$. Let us check that this choice is order-preserving (as the other three conditions are trivially satisfied). Note that $\mathcal{M}(x_j) \in \mathcal{G}_{k-1}$, so $R(x_k) \in \mathcal{G}_{k-1}$. Therefore, it must hold that $L(x_k) \in G_k \setminus G_{k-1}$ — it is the new point that we have added between $R(x_k)$, and the rightmost point the left of it on $\mathcal{G}_{k-1}$. Since it is also the case that $\mathcal{M}(x_i) \in \mathcal{G}_{k-1}$, we have that $\mathcal{M}(x_i) < \mathcal{M}(x_k) < \mathcal{M}(x_j)$.

By induction, we can compute a mapping with the required properties for k points. This completes the proof of the lemma. ■

Now we can define the equilibrium strategies. Let $x_1, \ldots, x_k$ be the history of cuts made at some point during the execution of the protocol. By Lemma 5.5.1, there exists an order-preserving map $\mathcal{M}$ such that each point x_i has a representative point $\mathcal{M}(x_i) \in \mathcal{G}_i$ and the piece $[x_i, \mathcal{M}(x_i)]$ is "small", i.e.

$$\mathtt{val}_l\,(x_i, \mathcal{M}(x_i)) \leq \frac{k\epsilon}{2f(n)^2} \leq \frac{\epsilon}{2f(n)}$$

for each agent $l \in \mathcal{N}$ — using $k \leq f(n)$.

Consider any history of cuts $(x_1, \ldots, x_k)$. Let i be the agent that moves next. Agent i computes the mapping $(\mathcal{M}(x_1), \ldots, \mathcal{M}(x_k))$. If the next operation is:

- *Choose*: agent i chooses the available piece (identified by the symbolic names of the cut points it contains and their order) which is optimal in the idealized game, given the current state and the existing set of ordered ideal cuts, $\mathcal{M}(x_1), \ldots, \mathcal{M}(x_k)$. Ties are broken according to a fixed deterministic scheme which is known to all the agents.
- *Cut*: agent i computes the optimal cut on $\mathcal{G}_{k+1}$, say at x^*_{k+1}. Then i maps x^*_{k+1} back to a point x_{k+1} on the continuous game, such that $\mathcal{M}(x_{k+1}) = x^*_{k+1}$. That is, the cut x_{k+1} (made in step $k+1$) is always mapped by the other agents to $x^*_{k+1} \in \mathcal{G}_{k+1}$. Agent i cuts at x_{k+1}.

We claim that these strategies give an ϵ-SPNE. The proof follows from the following lemma, which we show by induction on t (the maximum number of remaining steps of the protocol):

Lemma 5.5.2. *Given a point in the execution of the protocol from which there are at most t operations left until termination, it is $\frac{t\epsilon}{f(n)}$-optimal to play on the grid.*

Proof. Consider any history of play, where the cuts were made at $x_1, \ldots, x_k$. Without loss of generality, assume it is agent i's turn to move.

Base case: $t = 1$. The protocol has at most one remaining step. If it is a cut operation, then no agent receives any utility in the remainder of the game regardless of where the cut is made. Thus cutting on the grid ($\mathcal{G}_k$) is optimal. If it is a choose operation, then let $Z = \{Z_1, \ldots, Z_s\}$ be the set of pieces that i can choose from. Agent i's strategy is to map each piece Z_j to its equivalent $\mathcal{M}(Z_j)$ on the grid $\mathcal{G}_k$, and choose the piece that is optimal on $\mathcal{G}_k$. Recall that $\mathtt{val}_q\left(x_j, \mathcal{M}(x_j)\right) \leq \frac{\epsilon}{2f(n)}$ for each agent $q \in \mathcal{N}$. Thus if a piece is optimal on the grid, it is

$\frac{\epsilon}{f(n)}$-optimal in the continuous game (adding up the difference on both sides). It follows that i cannot gain more than $\frac{\epsilon}{f(n)}$ in the last step by deviating from the optimal piece on $\mathcal{G}_k$.

Induction hypothesis: Assume that playing on the grid is $\frac{(t-1)\epsilon}{f(n)}$-optimal whenever there are at most $t-1$ operations left on every possible execution path of the protocol, and there exists one path that has exactly $t-1$ steps.

Induction step: If the current operation is *Choose*, then by the induction hypothesis, playing on the grid in the remainder of the protocol is $\frac{(t-1)\epsilon}{f(n)}$-optimal for all the agents, regardless of i's move in the current step. Moreover, agent i cannot gain by more than $\frac{\epsilon}{f(n)}$ by choosing a different piece in the current step, compared to piece which is optimal on $\mathcal{G}_k$, since $\texttt{val}_i\,(x_l, \mathcal{M}(x_l)) \leq \frac{\epsilon}{2f(n)}$ for all $l \in \{1, \dots, k\}$.

If the current operation is *Cut*, then the following hold:

1. By construction of the grid $\mathcal{G}_{k+1}$, agent i can induce any given branch of the protocol using a cut in the continuous game if and only if the same branch can be induced using a cut on the grid $\mathcal{G}_{k+1}$.
2. Given that the other agents will play on the grid for the remainder of the protocol, agent i can change the size of at most one piece that it receives down the road by at most $\frac{\epsilon}{f(n)}$ by deviating (compared to the grid outcome), since $\texttt{val}_j\,(x_l, \mathcal{M}(x_l)) \leq \frac{\epsilon}{2f(n)}$ for all $l \in \{1, \dots, k+1\}$ and for all $j \in \mathcal{N}$.

Thus by deviating in the current step, agent i cannot gain more than $\frac{t\epsilon}{f(n)}$. ■

Since $t \leq f(n)$, the overall loss of any agent is bounded by ϵ by Lemma 5.5.2. We conclude that playing on the grid is ϵ-optimal for all the agents, which completes the proof of the theorem. ■

5.6 Proof of Theorem 5.3.2

Before we begin, we take this moment to formally introduce the auxiliary concept of a *mediated game* in an abstract sense. We will largely distance ourselves from the specificity of GCC games here and work in a more general model. We do this for two purposes. First, it allows for a cleaner view of the techniques; and second, we believe such general games may be of independent interest. We begin with a few definitions.

Definition 5.6.1. *In an extensive-form game, an* action tuple *is a tuple of actions that describe an outcome of the game. For example, the action tuple $(a_1, \dots, a_r)$ states that a_1 was the first action to be played, a_2 the second, and a_r the last.*

Definition 5.6.2. *Given an action tuple, the k^{th} action is said to be SPNE if the subtree of the game tree rooted where the first $k-1$ actions are played in accordance to the action tuple is induced by some SPNE strategy profile. Furthermore, call such an action tuple k-*SPNE*.*

Note that if the k^{th} action is SPNE, so too are all actions succeeding it in the action tuple. To clarify Definition 5.6.2, note that strategies of an extensive-form game are defined on every possible node of the game tree, so a k-SPNE action tuple can be equivalently defined as being an SPNE of the subgame rooted at the k^{th} action.

With these definitions in hand, we can now describe the games of interest.

Definition 5.6.3. *We call an extensive-form game a* mediated game *if the following conditions hold:*

1. *The set of agents consists of a single special agent, referred to as the* mediator*, and some finite number n of other regular agents. Intuitively, the mediator is an agent who is overseeing the proper execution of a protocol.*
2. *The height h of the game tree is bounded.*
3. *Every agent's utility is bounded.*
4. *Starting from the first or second action, the mediator plays every second action (and only these actions).*
5. *Every action played by the mediator shares the same action space:*

$$\{0,\ldots,n\} \times \left([0,1]^2 \cup 2^{\{1,\ldots,h\}}\right).$$

   This represents the agent who plays next (0 represents ending the game), and the interval which represents their action space or the allowed pieces they may choose from.

6. *The mediator's utility is binary (i.e. it is in $\{0,1\}$) and is described entirely by the notion of* allowed edges*. This is a set of edges in the game tree such that the mediator's utility is 1 iff it plays edges only in this set. Importantly, this set has the property that for every allowed edge, each grandchild subtree (i.e. subtree that represents the next mediator's action) must have at least one allowed edge from its root. Intuitively, these edges are the ones that follow the protocol the mediator is implementing.*
7. *A regular agent's utility is continuous*[2] *in the action tuple.*
8. Allowed-edges-closedness*: given a convergent sequence of action tuples where the mediator plays only allowed edges, the mediator must play only allowed edges in the limit action tuple as well.*

Note that appending meaningless actions (that affect no agent's utility) to a branch of the game tree will not affect the game in any impactful way. Thus, for the sake of convenience, we will assume for any game we consider all leaves of the game occur at the same depth (often denoted by r).

We now give a series of definitions and lemmas that culminate in the main tool used in the proof of Theorem 5.3.2: all mediated games have an SPNE.

Definition 5.6.4. *A sequence of action tuples $\left(a_1^i,\ldots,a_r^i\right)|_i$ is said to be* consistent *if for every j the agent who plays action a_j^i is constant throughout the sequence and, moreover, its action spaces are always subsets of $[0,1]$ or always the same subset of $\{1,\ldots,h\}$ throughout the sequence.*

Lemma 5.6.5. *Let $\left(a_1^i,\ldots,a_r^i\right)|_i$ be a sequence of action tuples in a mediated game. Then there is a convergent subsequence.*

Proof. Due to the finite number of agents and bounded height of the game, we can find an infinite consistent subsequence $\boldsymbol{b}^i|_i = \left(b_1^i,\ldots,b_r^i\right)|_i$. It suffices to show this subsequence has a

[2] The notions of convergence, compactness and continuity, which we will utilize often, necessarily assumes our action spaces are defined as metric spaces. Applicable metrics for the action spaces are not difficult to find, but are cumbersome to describe fully. We therefore will not belabour this point much further.

convergent subsequence of its own. It is fairly clear that we can find a convergent subsequence via compactness arguments, but there is a slight caveat: we must show that the limit action tuple is legal. That is, if the limit action tuple is $(a_1, \ldots, a_r)$ we must show that for every $i < r$ such that the mediator plays action i, action $i + 1$ is played by the agent prescribed by a_i, and within the bounds prescribed by it. We will prove this by induction.

Base hypothesis: First 0 actions have a convergent subsequence — this is vacuously true.

Induction hypothesis: Assume there exists a subsequence such that the first k actions converge legally.

Induction step: We wish to show that there exists a subsequence such that the first $k + 1$ actions converge. By the inductive assumption, there exists a subsequence $\boldsymbol{c}^i \mid_i$ such that the first k actions converge. Now suppose p plays the $k + 1^{th}$ action. If p is the mediator, then the action space is indifferent to actions played previously and is compact. Thus, the $\boldsymbol{c}^i \mid_i$ must have a convergent subsequence such that the $k + 1^{th}$ element of the action tuple converges and so we are done.

Alternatively, if p is a regular agent, the action space is not necessarily indifferent to previous actions. If the action spaces are always the same subset of $\{1, \ldots, h\}$, then we are clearly done. We therefore need only consider the case where the action spaces will be contained in $[0, 1]$. Due to the compactness of this interval, there will be a convergent subsequence of $\boldsymbol{c}^i \mid_i$ such that the $k + 1^{th}$ action converges to some $\gamma \in [0, 1]$. Call this subsequence $\boldsymbol{d}^i \mid_i$.

We argue that γ is in the limit action space of the $k+1^{th}$ action. For purposes of contradiction, assume this is false. Let δ be the length from γ to the closest point in the limit action space (i.e. the action space in the limit given by the k^{th} action played by the mediator). Then there exists some M such that after the M^{th} element in $\boldsymbol{d}^i \mid_i$, the closest point in the $k + 1^{th}$ action space to γ is at least $\delta/2$ away. Moreover, there exists some N such that after the N^{th} element in $\boldsymbol{d}^i \mid_i$ the $k + 1^{th}$ action is no further than $\delta/3$ to γ. Elements of $\boldsymbol{d}^i \mid_i$ after element $\max(M, N)$ then simultaneously must have the $k + 1^{th}$ action space be at least $\delta/2$ away from γ and have a point at most $\delta/3$ away from γ. This is a clear contradiction. ∎

Lemma 5.6.6. *For every k, if we have a convergent sequence of action tuples where the k^{th} action from the end is SPNE, then the k^{th} action from the end for the limit action tuple is also SPNE. That is, for every k, convergent sequences of $(r-k+1)$-SPNE action tuples are $(r-k+1)$-SPNE.*

Proof. We prove the result by induction on k.

Base Case $(k = 0)$*:* This is vacuously true.

Induction hypothesis $(k = m)$*:* Assume convergent sequences of $(r - m + 1)$-SPNE action tuples are $(r - m + 1)$-SPNE.

Induction step $(k = m + 1)$*:* Let $\boldsymbol{a}^i \mid_i = (a^i_1, \ldots, a^i_r) \mid_i$ be a convergent sequence of $(r - m)$-SPNE action tuples with the limit action tuple $(a_1, \ldots, a_r)$. We wish to show that if all actions before the last $m + 1$ actions play their limit actions, then the remaining $m + 1$ actions are SPNE — note that by Lemma 5.6.5 we know that the limit sequence is a valid action tuple.

Let p be the agent that commits the $m + 1^{th}$ action from the end. If p is the mediator, then by the definition of mediated games the desired statement is true (specifically via the allowed-edges-closedness condition). Now suppose instead that p is not the mediator, and simply a regular agent.

We show if the $m+1^{th}$ action from the end took on some other valid value $\alpha \neq a_{r-m}$, there exists SPNE strategies for the remaining m actions such that p achieves a utility no higher than had it stuck with the limit action of a_{r-m}.

So suppose the $m+1^{th}$ action from the end in the i^{th} element of the sequence is α^i such that $\lim_{i\to\infty} \alpha^i = \alpha$. Since $\boldsymbol{a}^i \mid_i$ is a sequence of $(r-m)$-SPNE action tuples, we can construct the sequence:

$$\boldsymbol{b}^i \mid_i = (a_1^i, \ldots, a_{r-m-1}^i, \alpha^i, \tilde{a}_1^i, \ldots, \tilde{a}_m^i) \mid_i$$

where the $\tilde{a}_j^i$ are SPNE actions such that p achieves at most the utility achieved by instead playing a_{r-m}^i. Via Lemma 5.6.5, $\boldsymbol{b}^i \mid_i$ must have a convergent subsequence — call $\boldsymbol{c}^i \mid_i$ and indexed by increasing function σ. That is, $\boldsymbol{c}^i = \boldsymbol{b}^{\sigma(i)}$. $\boldsymbol{c}^i \mid_i$ is then a convergent sequence of $(r-m+1)$-SPNE action tuples and thus, by the inductive assumption, its limit action tuple is also an $(r-m+1)$-SPNE.

Now consider the limit action tuple $(a_1, \ldots, a_r)$ (of $\boldsymbol{a}^i \mid_i$) and the limit action tuple of $\boldsymbol{c}^i \mid_i$ denoted by $(c_1, \ldots, c_r)$. Note that:

1. $\forall i < r-m$: $a_i = c_i$.
2. By the continuity requirement of mediated games (where V_p is the utility function of p):

$$\begin{aligned}
&V_p(a_1, \ldots, a_r)\\
&= \lim_{i\to\infty} V_p(a_1^i, \ldots, a_r^i)\\
&= \lim_{i\to\infty} V_p(a_1^{\sigma(i)}, \ldots, a_r^{\sigma(i)})\\
&\geq \lim_{i\to\infty} V_p(a_1^{\sigma(i)}, \ldots a_{r-m-1}^{\sigma(i)}, \alpha^{\sigma(i)}, \tilde{a}_{r-m+1}^{\sigma(i)}, \ldots, \tilde{a}_r^{\sigma(i)})\\
&= \lim_{i\to\infty} V_p(c_1^i, \ldots, c_r^i)\\
&= V_p(c_1, \ldots, c_r).
\end{aligned}$$

These two points imply that we can set SPNE strategies for the remaining m actions such that the utility of p playing α is less than or equal to if it plays a_{r-m} for the $m+1^{th}$ action from the end (when the actions preceding the $m+1^{th}$ action from the end are those given in the limit action tuple $(a_1, \ldots, a_r)$). As the α was arbitrary, the $m+1^{th}$ action from the end of $(a_1, \ldots, a_r)$ can be made an SPNE action, which completes the proof. ■

Lemma 5.6.7. *All mediated games have an SPNE.*

Proof. We prove the lemma via induction on the height of the game tree. Note that this is possible as mediated games (like extensive-form games) are recursive: the children of a node of a mediated game are mediated games.

Base case (at most 0 actions): This is vacuously true.

Induction hypothesis (at most k actions): Assume we have shown that any mediated game with a game tree of height at most k has an SPNE.

Induction step (at most $k+1$ actions): Let p be the agent that commits the first action. If p is the mediator, any action that is an allowed edge will be SPNE; and if no such action exists, any action will be SPNE (as the mediator is doomed to a utility of 0). Now suppose p is not the mediator.

Assume by the inductive assumption, once p makes its move, all remaining (at most) k actions are SPNE actions. By the definition of a mediated game, p's utility is bounded. Then the least upper bound property of $\mathbb{R}$ implies that p's utility as a function of the first action must have a supremum S. Via the axiom of choice, we construct a sequence of possible actions for the first action that approaches S in p's utility. That is, we have some sequence $x^i \mid_i$ such that if p plays x^i for the first action, it achieves some utility $f(x^i)$ — where $\lim_{i\to\infty} f(x^i) = S$. Moreover, let $g(x^i)$ map the action x^i to a tuple of the remaining actions — which are SPNE. By Lemma 5.6.5 $(x^i, g(x^i)) \mid_i$ must have a convergent subsequence $(y^i, g(y^i)) \mid_i$ that converges to $(y, g(y))$ — where y is a legal first action and $g(y)$ are legal subsequent actions.

Notice that $(y_i, g(y_i)) \mid_i$ is a convergent sequence of 2-SPNE action tuples and thus by Lemma 5.6.6, $(y, g(y))$ is a 2-SPNE action tuple as well. Furthermore, note that by the continuity requirement of mediated games, y must give p a utility of S. Therefore, this must be an SPNE action and so we are done. ■

With this machinery in hand, we are now ready to complete the proof of Theorem 5.3.2. Our main task is to make a formal connection between mediated games and (informed) GCC protocols.

Proof of Theorem 5.3.2. Suppose we have a n-agent GCC protocol $\mathcal{P}$ with a bounded number of steps and and set valuations of the agents $\texttt{val}_1, \ldots, \texttt{val}_n$. Then we wish to prove that there exists an informed GCC protocol $\mathcal{P}'$ that is equivalent to $\mathcal{P}$ up to tie-breaking such that the game induced by $\mathcal{P}'$ and $\texttt{val}_1, \ldots, \texttt{val}_n$ has an SPNE.

Outfit $\mathcal{P}$ as a game M, such that all but the final condition of mediated games are satisfied — that is, the mediator enforces the rules of $\mathcal{P}$ and achieves utility 1 if it follows the rules of $\mathcal{P}$ and 0 otherwise. More explicitly, the mediator plays every second action and upon examination of the history of events (i.e. the ordering of the cuts made thus far, and results of choose queries), decides the next agent to play and their action space based on the prescription of $\mathcal{P}$. To see how all but the last condition is satisfied, we go through them in order.

1. This is by definition.
2. The height of the tree is twice the height of the GCC protocol.
3. The mediator's utility is bounded by 1 by definition, and all other agent's utilities are bounded by 1 as that is their value of the entire cake.
4. This is by definition.
5. When the mediator wishes to ask a *Cut* query to agent i in the interval $[a, b]$, it plays the action $(i, (a, b))$, whereas when it wishes to ask a *Choose* query to agent i giving them the choice between the $x_1^{th}, \ldots, x_k^{th}$ pieces from the left, it plays the action $(i, \{x_1, \ldots, x_k\})$. This method of giving choose queries deviates slightly from the definition given in Section 5.2.1, but the two representations are clearly equivalent.
6. The allowed edges are ones that follow the rules of $\mathcal{P}$.
7. This property is only relevant when considering *Cut* nodes. To establish it, first consider the action in a single *Cut* node, and fix all the other actions. We claim that for every $\epsilon > 0$ there exists $\delta = \delta(\epsilon) > 0$ *that is independent of the choice of actions in other*

nodes such that moving the cut by at most δ changes the values by at most ϵ. Indeed, let us examine how pieces change as the cut point moves. As long as the cut point moves without passing any other cut point, one piece shrinks as another grows. As the cut point approaches another cut point, the induced piece — say k'th from the left — shrinks. When the cut point passes another cut point x, the k'th piece from the left grows larger, or it remains a singleton and another piece grows if there are multiple cut points at x. In any case, it is easy to verify that the sizes of various pieces received in *Choose* nodes change by at most δ if the cut point is moved by δ. Furthermore, note that the number of steps is bounded by r and — since the value density functions are continuous — there is an upper bound M on the value density functions such that if $y - x \leq \delta'$ then $\texttt{val}_i(x, y) \leq M\delta'$ for all $i \in \mathcal{N}$. Therefore, choosing $\delta \leq \epsilon/(Mr)$ is sufficient. Finally, $\texttt{val}_1, \ldots, \texttt{val}_n$ are continuous even in the actions taken in multiple *Cut* nodes, because we could move the cut points sequentially.

We now alter M such that at every branch induced by a comparison of cuts via an *If-Else*, we allow in the case of a tie to follow either branch. Formally, suppose at a branch induced by the statement "*if* $x \leq y$ then A *else* B" we now set in the case of $x = y$ the edges for both A *and* B as allowed. Then we claim the property of allowed-edges-closedness is satisfied.

To see this, let us consider action tuples. An action tuple where the mediator in M only plays on allowed edges can be viewed as a trace of an execution of $\mathcal{P}$ which records the branch taken on every *If-Else* statement — though when there is a tie the trace may follow the "incorrect" branch. A convergent sequence of such action tuples at some point in the sequence must then keep the branches it chooses in the execution of $\mathcal{P}$ constant — unless in the limit, the cuts compared in a branch that is not constant coincide. Thus, we have that in the limit, if a branch is constant, the mediator always takes an allowed edge trivially, and otherwise due to our modification of M the mediator still takes an allowed edge. Furthermore, for all actions of the mediator that are not induced by *If-Else* statements, the mediator clearly still plays on allowed edges and so we have proved the claim.

Now as M is a mediated game, it has an SPNE S by Lemma 5.6.7. Let $\mathcal{P}'$ be the informed GCC protocol equivalent to $\mathcal{P}$ up to tie-breaking such that for every point in the game tree of M that represents the mediator branching on an "*if* $x \leq y$ then A *else* B" statement in the original protocol $\mathcal{P}$, $\mathcal{P}'$ chooses the A or B that S takes in the event of a tie. Then the informedness of the tie-breaking is built into $\mathcal{P}'$ and we immediately see that the SPNE actions of the regular agents in M correspond to SPNE actions in $\mathcal{P}'$. ■

5.7 Proof of Theorem 5.4.1

The proof of the theorem uses the Thieves Protocol given by Algorithm 12. In this protocol, agent 1 first demarcates a contiguous allocation $X = \{X_1, \ldots, X_n\}$ of the entire cake, where X_i is a contiguous piece that corresponds to agent i. This can be implemented as follows. First, agent 1 makes n cuts such that the i-th cut is interpreted as the left endpoint of X_i. The left endpoint of the leftmost piece is reset to 0 by the protocol. Then, the rightmost endpoint of X_i is naturally the leftmost cut point to its right or 1 if no such point exists. Ties among overlapping cut points are resolved in favor of the agent with the smallest index; the corresponding cut point is assumed

```
Agent 1 demarcates a contiguous allocation X of the cake
for i = 2, ..., n, 1 do
  // Verification of envy-freeness for agent i
  Agent i Cuts in {[0, 1]} // @w_i
  Agent i Cuts in {[w_i, 1]} // @ z_i
  for j = 1 to n do
    if ∅ ≠ ([w_i, z_i] ∩ X_j) ⊊ X_j then
      // Agent i steals a non-empty strict subset of X_j
      Agent i Chooses from {[w_i, z_i] ∩ X_j}
      exit // Verification failed: protocol terminates
    end if
  end for
  // Verification successful for agent i
end for
for i = 1 to n do
  Agent i Chooses from {X_i}
end for
```

Algorithm 12: Thieves Protocol: Every NE induces a contiguous envy-free allocation that contains the entire cake and vice versa.

to be the leftmost one. Notice that every allocation that assigns nonempty contiguous pieces to all agents can be demarcated in this way.

After the execution of the demarcation step, X is only a tentative allocation. Then, the protocol enters a verification round, where each agent i is allowed to *steal* some non-empty strict subset of a piece (say, X_j) demarcated for another agent. If this happens (i.e., the if-condition is true) then agent i takes the stolen piece and the remaining agents get nothing. This indicates the failure of the verification and the protocol terminates. Otherwise, the pieces of X are eventually allocated to the agents, i.e., agent i takes X_i.

We will require two important characteristics of the protocol. First, it guarantees that no state in which some agent steals can be a NE; this agent can always steal an even more valuable piece. Second, stealing is beneficial for an envious agent.

Proof of Theorem 5.4.1. Let $\mathcal{P}$ be the Thieves protocol given by Algorithm 12 and $\mathcal{E}$ be any NE of $\mathcal{P}$. Denote by X the contiguous allocation of the entire cake obtained during the demarcation step, where $X_i = [x_i, y_i]$ for all $i \in \mathcal{N}$, and let w_i and z_i be the cut points of agent i during its verification round. Assume for the sake of contradiction that X is not envy-free. Let k^* be an envious agent, where $\mathtt{val}_{k^*}\left(X_{j^*}\right) > \mathtt{val}_{k^*}(X_{k^*})$, for some $j^* \in \mathcal{N}$. There are two cases to consider:

Case 1: Each agent i receives the piece X_i in $\mathcal{E}$. This means that, during its verification round, each agent i selects its cut points from the set $\bigcup_{j=1}^{n}\{x_j, y_j\}$. By the non-envy-freeness condition for X above (and by the fact that the valuation function $\mathtt{val}_{k^*}$ is strictly positive), there exist w'_{k^*}, z'_{k^*} such that $x_{j^*} < w'_{k^*} < z'_{k^*} < y_{j^*}$ and $\mathtt{val}_{k^*}\left(w'_{k^*}, z'_{k^*}\right) > \mathtt{val}_{k^*}(x_{k^*}, y_{k^*})$. Thus, agent k^* could have been better off by cutting at points w'_{k^*} and z'_{k^*} in its verification round, contradicting the assumption that $\mathcal{E}$ is a NE.

Case 2: There exists an agent i that did not receive the piece X_i. Then, it must be the case that some agent k stole a non-empty strict subset $[w''_k, z''_k] = [w_k, z_k] \cap Z_j$ of another piece X_j. However, agent k could have been better off at the node in the game tree reached in its verification round by making the following marks: $w'_k = \frac{x_j + w''_k}{2}$ and $z'_k = \frac{z''_k + y_j}{2}$. Since either $x_j \leq w''_k < z''_k < y_j$ or $x_j < w''_k < z''_k \leq y_j$ (recall that $[w''_k, z''_k]$ is a non-empty strict subset of X_j and the valuation function $\texttt{val}_k$ is strictly positive), it is also true that $\texttt{val}_k\left(w'_k, z'_k\right) > \texttt{val}_k\left(w''_k, z''_k\right)$, again contradicting the assumption that $\mathcal{E}$ is a NE.

So, the allocation computed by agent 1 under every NE $\mathcal{E}$ is indeed envy-free; this completes the proof of the first part of the theorem.

We next show that every contiguous envy-free allocation of the entire cake is the outcome of a NE. Let Z be such an allocation, with $Z_i = [x_i, y_i]$ for all $i \in \mathcal{N}$. We define the following set of strategies $\mathcal{E}$ for the agents:

- At every node of the game tree (i.e., for every possible allocation that could be demarcated by agent 1), agent $i \geq 2$ cuts at points $w_i = x_i$ and $z_i = y_i$ during its verification round.
- Agent 1 specifically demarcates the allocation Z and cuts at points $w_1 = x_1$ and $z_1 = y_1$ during its verification round.

Observe that $[w_i, z_i] \cap Z_j$ is either empty or equal to Z_j for every pair of $i, j \in \mathcal{N}$. Hence, the verification phase is successful for every agent and agent i receives the piece Z_i.

We claim that this is a NE. Indeed, consider a deviation of agent 1 to a strategy that consists of the demarcated allocation Z' (and the cut points w'_1 and z'_1). First, assume that the set of pieces in Z' is different from the set of pieces in Z. Then there is some agent $k \neq 1$ and some piece Z'_j such that the if-condition $\emptyset \subset [x_k, y_k] \cap Z'_j \subset Z'_j$ is true. Hence, the verification round would fail for some agent $i \in \{2, \ldots, k\}$ and agent 1 would receive nothing. So, both Z' and Z contain the same pieces, and may differ only in the way these pieces are tentatively allocated to the agents. But in this case the maximum utility agent 1 can get is $\max_j \texttt{val}_1\left(Z'_j\right)$, either by keeping the piece Z'_1 or by stealing a strict subset of some other piece Z'_j. Due to the envy-freeness of Z, we have:

$$\max_j \texttt{val}_1\left(Z'_j\right) = \max_j \texttt{val}_1\left(Z_j\right) = \texttt{val}_1\left(Z_1\right),$$

hence, the deviation is not profitable in this case either.

Now, consider a deviation of agent $i \geq 2$ to a strategy that consists of the cut points w'_i and z'_i. If both w'_i and z'_i belong to $\bigcup_{j=1}^n \{x_i, y_i\}$, then $[w'_i, z'_i] \cap Z_j$ is either empty or equal to Z_j for some $j \in \mathcal{N}$. Hence, the deviation will leave the allocation unaffected and the utility of agent i will not increase. If instead one of the cut points w'_i and z'_i does not belong to $\bigcup_{j=1}^n \{x_i, y_i\}$, this implies that the condition

$$\emptyset \subset [w'_i, z'_i] \cap Z_j \subsetneq Z_j$$

is true for some $j \in \mathcal{N}$, i.e., agent i will steal the piece $[w'_i, z'_i] \cap Z_j$. However, the utility $\texttt{val}_i\left([w'_i, z'_i] \cap Z_j\right)$ of agent i cannot be greater than $\texttt{val}_i\left(Z_j\right)$, which is at most $\texttt{val}_i\left(Z_i\right)$ due to the envy-freeness of Z. Hence, again, this deviation is not profitable for agent i.

We conclude that $\mathcal{E}$ is a NE; this completes the proof of the theorem. ■

Chapter 6

Existence of Maximin Share Allocations and Their Extensions

6.1 Introduction

In this chapter, we are interested in the *fair* allocation of *indivisible* goods, but to explain the intricacies of this problem we start with a quick re-examining of the case of *divisible* goods (i.e. the continuous analogue) which we have explored in Chapters 4 and 5. In this latter setting, we have seen that we need to divide a heterogeneous cake between agents with different valuation functions (that is, different agents may have different values for the same piece of cake).

When there are only two agents, the *Cut and Choose* protocol provided a compelling method for dividing a cake — and will play an important conceptual role later on. Recall that under this protocol, agent 1 cuts the cake into two pieces that he values equally, and agent 2 subsequently chooses the piece that he prefers, giving the other piece to agent 1. The resulting allocation is fair in the precise, formal sense known as *envy-freeness*: Each agent (weakly) prefers his own allocation to the allocation of the other agent. Envy-free cake divisions exist for any number of agents; today we know exactly how many cuts are needed to achieve such allocations in the worst case [1], and how to constructively find them [20], [9]. Moreover, in the standard cake-cutting setting — envy-freeness implies another natural fairness property called *proportionality*: Each agent in the set of agents $\mathcal{N}$ receives a piece of cake whose value is at least $1/|\mathcal{N}|$ of the agent's value for the entire cake.

Cake cutting is a nice metaphor for real-world problems like land division; the study of cake cutting distills insights about fairness that are useful in related settings, such as the allocation of computational resources [49, 81, 58, 84]. However, typical real-world situations where fairness is a chief concern, for example, divorce settlements and the division of an estate between heirs, involve *indivisible goods* (e.g., houses, cars, and works of art) — which in general preclude envy-free, or even proportional, allocations. As a simple example, if there are several agents and only one indivisible item to be allocated, the allocation cannot possibly be proportional or envy free. Foreshadowing the approach we take below, we note that no allocation can be even *approximately* (in a multiplicative sense) fair according to these notions, because some agents receive an empty allocation of zero value.

So how can we divide an estate without lawyers? Potentially using an intriguing alternative to classical fairness notions, recently presented by Budish [27] (building on concepts introduced by Moulin [76]). Imagine that agent 1 partitioned the items into $|\mathcal{N}|$ bundles, and each agent in $\mathcal{N} \setminus \{1\}$ *adversarially* chose a bundle before agent 1. A smart agent would partition the bundles to maximize his minimum value for any bundle. For the same reason we intuitively view the Cut-and-Choose protocol as fair to agent 1, even before specifying fairness axioms, the allocation that leaves agent 1 with his least desired bundle seems fair to agent 1 — as he is the one who divided the items in the first place. Budish calls the value agent 1 can guarantee in this way his *maximin share (MMS)* guarantee.[1] But an allocation based on the division of agent 1 may make another agent regret the fact that he was not the one to divide the items. The question is: Can we allocate the items in a way that *all* agents receive a bundle worth at least as much as their MMS guarantee? This question was recently addressed by Bouveret and Lemaître [16], and while they were able to answer it for special cases (which we list in Section 6.1.3), they left the general question open.

6.1.1 Model, Conceptual Contribution, and Technical Results

Let us begin with some notation and definitions.

- $\mathcal{N} = \{1, \ldots, n\}$: The set of agents (of which there are n).
- $\mathcal{M}$: The set of indivisible goods/items (of which there are m).
- For all $k \in \mathbb{Z}_{>0}, [k] = \{1, \ldots, k\}$.
- $\mathtt{val}_i : 2^{\mathcal{M}} \to \mathbb{R}_{\geq 0}$: The function taking a subset of the goods and returning i's value for said goods. We simplify notation by writing $\mathtt{val}_i(j)$ instead of $\mathtt{val}_i(\{j\})$ for a single item $j \in \mathcal{M}$. We assume that the valuation functions are *additive* (i.e. $\forall S \subseteq \mathcal{M}$, $\mathtt{val}_i(S) = \sum_{j \in S} \mathtt{val}_i(j)$).
- For any $S \subseteq \mathcal{M}$, $\Pi_k(S)$ is the set of k-partitions of S. That is, the partitions of S that comprise of k sets.
- *k-maximin share (k-MMS) guarantee* of agent $i \in \mathcal{N}$ is given by:

$$\mathtt{MMS}_i(k, S) = \max_{T_1, \ldots, T_k \in \Pi_k(S)} \min_{j \in [k]} \mathtt{val}_i\left(T_j\right).$$

  We call a partition that realizes this value agent i's *k-maximin partition of S*. The valuation function used to determine an agent's MMS guarantee will be clear from the context.

- An *allocation* $A_1, \ldots, A_n \in \Pi_n(\mathcal{M})$ allocates the subset of items A_i to each agent i.
- An allocation $A_1, \ldots, A_n$ is a *maximin share (MMS) allocation* if:

$$\forall i \in \mathcal{N},\ \mathtt{val}_i(A_i) \geq \mathtt{MMS}_i(n, \mathcal{M}).$$

The assumption of additivity may seem somewhat restrictive, but is made in most of the related work on fair division of indivisible goods (see Section 6.1.3), including the paper of

[1]This term should not be confused with the terminology of the systems literature, where max-min fairness simply refers to maximizing the value any agent receives [35] rather than an axiomatic notion of fairness.

Bouveret and Lemaître [16] that studies the maximin share guarantee in the same setting. And more importantly, people find it difficult to specify combinatorial preferences, which is why some deployed implementations of fair division methods (see Section 6.1.2) rely on additive valuation functions. Finally, our positive result does not hold under larger classes of valuation functions, e.g., subadditive and superadditive functions.

For the case of $n = 2$ constructing an MMS allocation can be trivially done by having one agent produce an MMS partition, the other then choosing the better of the two sets (in his view), and the other set going to the producer of the partition. In essence, this is the Cut and Choose protocol in the indivisible good setting. Thus the interesting questions lie in the setting where $n \geq 3$. Our first result — the punchline of Section 6.2 — is negative:

Theorem 6.2.1. *For any set of agents $\mathcal{N}$ such that $n \geq 3$ there exist a set of items $\mathcal{M}$ of size $m \leq 3n + 4$, and (additive) valuation functions, that do not admit an MMS allocation.*

We find this theorem surprising because extensive automated experiments by several groups of researchers (including us) had failed to find a counterexample. Indeed, the counterexamples rely on very intricate constructions. In Section 6.2 we first provide explicit counterexamples for the cases of three and four agents (the latter illustrates the key ideas), and then give the full proof.

While this news may appear somewhat disconcerting, we strive in Sections 6.3, 6.4, and 6.5 to rosy the picture. In Section 6.3 we relax the MMS fairness notion in order to guarantee existence. Unlike other fairness notions such as envy-freeness, the MMS guarantee supports a multiplicative notion of approximation. Our main question is:

> *Is there a value $\gamma > 0$ such that we can always find an allocation $A_1, \ldots, A_n$ that satisfies $\texttt{val}_i(A_i) \geq \gamma \cdot \texttt{MMS}_i(n, \mathcal{M})$ for all i?*

We answer this question in the positive for

$$\gamma = \gamma_n := \frac{2\lfloor n \rfloor_{odd}}{3\lfloor n \rfloor_{odd} - 1}, \text{ or alternatively, } \frac{2\left\lfloor \frac{n+1}{2} \right\rfloor - 1}{3\left\lfloor \frac{n+1}{2} \right\rfloor - 2}$$

where $\lfloor n \rfloor_{odd}$ is the largest odd number that is less than or equal to n. Note that γ_n is always greater than $2/3$, and it is equal to $3/4$ for the important cases of three and four agents. More precisely, we prove the following theorem in Section 6.3.

Theorem 6.3.1. *There always exists an allocation $A_1, \ldots, A_n$ such that for all $i \in \mathcal{N}$, $\texttt{val}_i(A_i) \geq \gamma_n \texttt{MMS}_i(n, \mathcal{M})$. Moreover, for every $\epsilon > 0$, an allocation $A_1, \ldots, A_n$ such that for all $i \in \mathcal{N}$, $\texttt{val}_i(A_i) \geq (1 - \epsilon)\gamma_n \texttt{MMS}_i(n, \mathcal{M})$ can be computed in polynomial time in n and m.*

In Section 6.4 we then give theoretical explanations on why MMS allocations always exist in practice and simulations by showing that under a sensible randomized model, such allocations exist with high probability. Section 6.5 further adds that in the case of $m \leq n + 4$ goods we can always guarantee an MMS allocation.

Finally, in Section 6.6 we examine a slightly tangential fairness notion recently introduced in [30] that applies the concept of MMS at the pairwise level. We improve the best guaranteed approximation factor known for this notion from $2/\left(1 + \sqrt{5}\right) \approx 0.618$ to $\left(\sqrt{17} - 1\right)/4 \approx 0.781$.

6.1.2 Practical Applications of Our Results

The theory of fair division has been extensively studied, as shown, e.g., by the books by Moulin [75] and Brams and Taylor [21]. Despite the abundance of extremely clever fair division algorithms, very few have been implemented. Budish's [27] work is a rare example; his method is currently used for MBA course allocation at the Wharton School of the University of Pennsylvania. Another example is the *adjusted winner* method [21], which assumes that there are exactly two agents (with additive valuation functions). Adjusted winner has been patented by NYU and licensed to *Fair Outcomes, Inc.*

As mentioned in Chapter 1, we are involved in an effort to change this situation by building a fair-division website called *Spliddit* [50], available at `www.spliddit.org`. Spliddit contains implementations of existing mechanisms for the division of rent, credit, taxi/Uber fare, and chores. However, for the fifth application — dividing indivisible goods — we were unable to find satisfactory methods for more than two agents, despite discussions with leading experts on fair division (we survey some existing methods in Section 6.1.3). This provided strong motivation for the theoretical work reported here.

The approach we ultimately implemented relied heavily on Theorem 6.3.1[2]. We consider three "levels" of fairness: envy-freeness, proportionality, and (approximate) MMS guarantee. It is easy to verify that each of these fairness notions implies the ones following it. Users specify their valuation functions by distributing a fixed pool of points between the items. We then find an allocation that maximizes social welfare — $\sum_{i\in\mathcal{N}} \mathtt{val}_i\,(A_i)$ — subject to the strongest feasible fairness constraint (using an integer linear programming formulation, which is solved via CPLEX). For MMS, we maximize the value of γ for which the γ-MMS guarantee is feasible. By Theorem 6.3.1, achieving $\gamma > 2/3$ of the MMS guarantee is always feasible, so the theorem ensures an outcome that is, well, fair enough. By providing rigorous fairness guarantees that are easy to explain, it justifies Spliddit's tagline, "provably fair solutions".

6.1.3 Related Work

Prior work

Motivated by the problem of allocating courses to students, Budish [27] studies a solution concept that he calls *approximate competitive equilibrium from equal incomes (CEEI)*. Budish shows the existence of an approximate CEEI (with certain approximation parameters), even when the preferences of agents are unrestricted (so they may correspond to any combinatorial valuation functions). Roughly speaking, an approximate CEEI guarantees that $\mathtt{val}_i\,(A_i) \geq \mathtt{MMS}_i(n+1, \mathcal{M})$, that is, each of the n agents receives its $(n+1)$-MMS guarantee. However, this result takes advantage of an approximation error in the items that are allocated (some items might be in excess demand or excess supply). The approximation error grows with the overall number of items, and with the number of items demanded by each agent, but not with the number of agents or the number of copies of each item. Therefore, as the two latter parameters go to infinity, the error goes to zero. A large economy, in this sense, is plausible in the context of MBA course allocation, because there are many MBA students, many seats in each course, but relatively few

[2] We have since further refined our approach as discussed in [30].

courses that are offered, and even fewer courses a single student can take. But Budish's results do not provide practical guarantees when there are, say, three or four agents, and (very possibly) only one copy of each item — which is the setting we are interested in.

Like us, Bouveret and Lemaître [16] focus on the division of indivisible goods between agents with additive valuations. They study a hierarchy of fairness properties, of which the maximin share guarantee is the weakest (it is easy to see that allocations satisfying the other properties may not exist). Among other results, they show that MMS allocations exist in the following cases: (i) valuations for items are 0 or 1; (ii) the values different agents assign to items form identical multisets; and (iii) $m \leq n + 3$. They also present results from extensive simulations using different distributions over item values; MMS allocations exist in each and every trial.

Also related is the work of Lipton et al. [65]. Among other results, they give a polynomial-time algorithm that computes approximately envy-free allocations, where the approximation is *additive*. Specifically, they let α be the largest possible increase in value an agent can have from adding one item to his bundle, and produce an allocation such that $\mathtt{val}_i(A_i) \geq \mathtt{val}_i\left(A_j\right) - \alpha$ for all $i, j \in \mathcal{N}$. This interesting result may not be very practical in and of itself; for example, if one of the items is extremely valuable, the agents would not be guaranteed anything. In contrast, assuming items have positive values, an MMS allocation (or any multiplicative approximation thereof) gives some agent a bundle worth zero (if and) only if *any* allocation gives some agent a bundle worth zero.

Hill [55] shows that when valuations are additive, indivisible items can be allocated in a way that a certain value is guaranteed to each agent; and Markakis and Psomas [67] refine this guarantee and construct a polynomial time algorithm that achieves it. However, the guaranteed value is defined using an unwieldy function that depends on the number of agents as well as on the value of the most valuable item, and even for three agents the function's value quickly goes down to zero as the most valuable item becomes more valuable.

When there are exactly two agents, practical methods for dividing indivisible goods are available. For example, recent work by Brams et al. [19] gives a method satisfying several desirable properties, including envy-freeness; its main shortcoming is that it may not allocate all items (it generates a "contested pile" of unallocated items). The *adjusted winner* method [21], mentioned above, is another practical method (which is routinely being used, as discussed in Section 6.1.2) — but it implicitly assumes that the items are divisible and would typically require splitting one of the items. In any case, for more than two agents, one encounters a great many paradoxes when contemplating standard fairness notions [17]. Moreover, generalizing these practical 2-agent protocols is impossible; for example, adjusted winner can be interpreted as a special case of the *egalitarian equivalent* [82] rule (for two agents and additive valuation functions), but the latter method strongly relies on divisibility and may end up splitting all goods.

From an algorithmic viewpoint, our work is related to papers on the problem of allocating indivisible goods to maximize the minimum value any agent has for his bundle (under additive valuation functions) — also known as the *Santa Claus* problem [14, 11, 6]. Woeginger [99] studies the special case of agents with identical valuations, and presents a polynomial time approximation scheme that we leverage below.

Subsequent work

Since the publication of the earliest version of our results [86], several papers have followed up on our work.

The preliminary version of Theorem 6.3.1 [86] achieves a $2/3 - \epsilon$ approximation of the MMS guarantee in polynomial time *only in m*, that is, computational efficiency requires a constant number of agents. The main result of Amanatidis et al. [4] improves the running time of that algorithm: they achieve a $2/3-\epsilon$ fraction of the MMS guarantee in polynomial time for any number of agents. They do this by modifying one of the steps of the original (unintuitive) algorithm of Procaccia and Wang [86]. The current proof of Theorem 6.3.1 is completely different from the original one, and, in particular, immediately leads to an (arguably) intuitive, polynomial-time algorithm. Among other results, Amanatidis et al. [4] also show that a 7/8-MMS allocation can be guaranteed for three agents, improving on our bound of 3/4 for this case.

In a newer paper, Amanatidis et al. [3] design *truthful* approximations algorithms for the MMS guarantee. For the so-called *cardinal model*, where agents report their value for each item, they provide a truthful algorithm that achieves a $\Theta(m)$-approximation of the MMS guarantee. For the case of two agents (where an MMS allocation always exists), they are able to give a truthful 1/2-approximation of the MMS guarantee, and prove that no truthful algorithm can yield a better ratio.

In our very recent work with colleagues [29], we advocate the *Max Nash Welfare* solution, which maximizes the product of utilities, as a method for allocating indivisible goods. We show that this solution, which is clearly Pareto efficient, satisfies an approximate envy-freeness property, and also provides a $\Theta(1/\sqrt{n})$ approximation of the MMS guarantee in theory, and a much better approximation in practice. The new solution was deployed on Spliddit in May 2016.

6.1.4 Open Problems

One obvious question remains open. Theorem 6.2.1 does not provide an upper bound on the the constant $\gamma > 0$ such that γ-MMS allocations always exist, and our constructions in Section 6.2 provide very weak upper bounds. Our lower bound, given by Theorem 6.3.1, is 2/3. Narrowing this gap is, in our view, an important challenge.

As noted above, Budish [27] introduced a different notion of MMS approximation. In its ideal form, we would ask for an allocation such that $\mathtt{val}_i(A_i) \geq \mathtt{MMS}_i(n+1, \mathcal{M})$. We have designed an algorithm that achieves this guarantee for the case of three agents (it is already nontrivial). Proving or disproving the existence of such allocations for a general number of agents remains an open problem; a positive result would provide a compelling alternative to Theorem 6.3.1.

6.2 Nonexistence of Exact MMS Allocations

In this section we will show that, in general, MMS allocations are not guaranteed to exist (even under our assumption of additive valuation functions). But, to give some context for this result, let us briefly discuss a case where they *do* exist. As briefly mentioned previously and first pointed out by Bouveret and Lemaître [16], when there are two agents we can achieve an MMS allocation — essentially via an indivisible analog of the Cut and Choose protocol. First, let agent 1 divide the items according to a 2-maximin partition S_1, S_2 of his, i.e., the partition that maximizes $\min_{j\in[2]} \mathtt{val}_1(S_j)$. Allocate to agent 2 his preferred subset, and give the other subset to agent 1. Agent 1 clearly achieves his MMS guarantee, but what about agent 2? By the additivity of $\mathtt{val}_2$, there exists $j \in [2]$ such that $\mathtt{val}_2(S_j) \geq \mathtt{val}_2(\mathcal{M})/2$. In addition, in any partition S_1', S_2' there exists $k \in [2]$ such that $\mathtt{val}_2(S_k') \leq \mathtt{val}_2(\mathcal{M})/2$, hence $\mathtt{MMS}_2(2, \mathcal{M}) \leq \mathtt{val}_2(\mathcal{M})/2$. It follows that there exists $j \in [2]$ such that $\mathtt{val}_2(S_j) \geq \mathtt{MMS}_2(2, \mathcal{M})$.

In contrast, MMS allocations may not exist when the number of agents is at least three.

Theorem 6.2.1. *For any set of agents $\mathcal{N}$ such that $n \geq 3$ there exist a set of items $\mathcal{M}$ of size $m \leq 3n+4$, and (additive) valuation functions, that do not admit an MMS allocation.*

The case of $n = 3$ is handled separately, in Section 6.2.1. A single construction works for any $n \geq 4$, but because it is rather complex, we first illustrate the main ideas in Section 6.2.2 for the special case of $n = 4$, and then provide the full construction in Section 6.2.3.

6.2.1 Proof of Theorem 6.2.1 for $n = 3$

Let the set of items be $\mathcal{M} = \{(i, j) \mid i \in [3], j \in [4]\}$ (note that $m = 12 < 3n + 4$). The valuation functions of the three agents are defined using the following two matrices:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 17 & 25 & 12 & 1 \\ 2 & 22 & 3 & 28 \\ 11 & 0 & 21 & 23 \end{bmatrix},$$

in conjunction with the three matrices:

$$E^{(1)} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E^{(2)} = \begin{bmatrix} 3 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \quad E^{(3)} = \begin{bmatrix} 3 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

For each item $(i, j) \in \mathcal{M}$, we let

$$\mathtt{val}_k(\{(i, j)\}) = 10^6 \cdot S_{i,j} + 10^3 \cdot T_{i,j} + E^{(k)}_{i,j}.$$

Our first goal is to compute the MMS guarantee of each agent. To this end, we will find it convenient to label each element of T with three of nine possible labels $(1, 2, 3, \alpha, \beta, \gamma, +, -, *)$:

$$\begin{bmatrix} {}^{\alpha}17^{1}_{+} & {}^{\alpha}25^{1}_{-} & {}^{\beta}12^{1}_{+} & {}^{\gamma}1^{1}_{*} \\ {}^{\alpha}2^{2}_{-} & {}^{\beta}22^{2}_{*} & {}^{\gamma}3^{2}_{+} & {}^{\gamma}28^{2}_{-} \\ {}^{\alpha}11^{3}_{*} & {}^{\beta}0^{3}_{-} & {}^{\beta}21^{3}_{*} & {}^{\gamma}23^{3}_{+} \end{bmatrix}$$

T has the following Sudoku-like property: For each label there are exactly four elements with that label, and the sum of these 4 elements is exactly 55. Moreover, any four elements whose sum is 55 must have the same label.

This observation facilitates a straightforward computation of MMS guarantees. Agent 1 can divide the 12 items into three subsets: a subset consisting of the four elements labeled with 1 (the first row), a subset consisting of the four elements labeled by 2 (the second row), and a subset consisting of the four elements labeled by 3 (the third row). For each subset, the sum of its four elements in S, T and $E^{(1)}$ is 4, 55 and 0 respectively. Hence, $\mathtt{MMS}_1(3, \mathcal{M}) = 4 \cdot 10^6 + 55 \cdot 10^3 + 0 = 4055000$. Agent 2's maximin partition is obtained by dividing the items into three subsets according to the labels α, β and γ, and agent 3's maximin partition corresponds to the labels $+, -$ and $*$; all MMS guarantees are 4055000.

We next characterize MMS allocations of $\mathcal{M}$, with the goal of showing that no such allocations exist. First note that a valid MMS allocation of $\mathcal{M}$ must allocate at least four items to each agent. Indeed, for any bundle $X \subseteq \mathcal{M}$ such that $|X| = 3$ and each agent $i = 1, 2, 3$, $\mathtt{val}_i(X) \leq 3 \cdot 10^6 + 76 * 10^3 + 3 < 4055000$. Because there are twelve items, each agent must receive exactly four items.

We now claim that in an MMS allocation each agent must receive four items with the same label. Indeed, as noted above, the only bundles whose values in T add up to 55 consist of four items with identical labels. Suppose that an agent is allocated four items with different labels. Since the sum of all the elements in T is $165 = 55 \times 3$, there must be an agent with four items whose sum in T is less than 55. This agent's value is at most $4 \cdot 10^6 + 54 \cdot 10^3 + 3 < 4055000$.

It is easy to verify that there are only three ways to divide $\mathcal{M}$ into three subsets such that the items in each subset have identical labels:

1. Dividing according to the labels $1, 2, 3$.
2. Dividing according to the labels α, β, γ.
3. Dividing according to the labels $+, -$ and $*$.

All three ways will fail to give some agent his MMS guarantee of 4055000. Indeed, in case (1), there is an agent $i_1 \in \{2, 3\}$ who is allocated items labeled by 2 or 3. The sum of the corresponding elements in $E^{(i_1)}$ is -1, hence the value i_1 obtains is $4 \cdot 10^6 + 55 \cdot 10^3 - 1 = 4054999 < 4055000$. In case (2), an agent $i_2 \in \{1, 3\}$ must be allocated a subset of items labeled with β or γ; and in case (3), an agent $i_3 \in \{1, 2\}$ must be allocated a subset of items labeled with $-$ or $*$. By the same reasoning as in case (1), in cases (2) and (3) agent i_l, $l = 2, 3$, ends up with value at most 4054999. We conclude that it is impossible to satisfy the MMS guarantees of all three agents. ■

6.2.2 Proof of Theorem 6.2.1 for $n = 4$

Because the construction for $n \geq 4$ is somewhat intricate, we start by explicitly providing the special case of $n = 4$ as previously mentioned. To this end, let us define the following two matrices, where ϵ is a very small positive constant ($\epsilon = 1/16$ will suffice).

$$S = \begin{bmatrix} \frac{7}{8} & 0 & 0 & \frac{1}{8} \\ 0 & \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & \frac{1}{8} \end{bmatrix}, \quad T = \begin{bmatrix} 0 & \epsilon^4 & 0 & -\epsilon^4 \\ \epsilon^3 & 0 & -\epsilon^3 + \epsilon^2 & -\epsilon^2 \\ 0 & -\epsilon^4 + \epsilon & 0 & \epsilon^4 - \epsilon \\ -\epsilon^3 & -\epsilon & \epsilon^3 - \epsilon^2 & \epsilon^2 + \epsilon \end{bmatrix}.$$

Let $M = S + T$. Crucially, the rows and columns of M sum to 1. Let $\mathcal{M}$ contain goods that correspond to the nonzero elements of M, that is, for every entry $M_{i,j} > 0$ we have a good (i, j); note that $m = 14 < 3n + 4$.

Next, partition the 4 agents into $P = \{1, 2\}$ and $Q = \{3, 4\}$. Define the valuations of the agents in P as follows where $0 < \tilde{\epsilon} \ll \epsilon$ ($\tilde{\epsilon} = 1/64$ will suffice):

$$M + \begin{bmatrix} 0 & 0 & 0 & -\tilde{\epsilon} \\ 0 & 0 & 0 & -\tilde{\epsilon} \\ 0 & 0 & 0 & -\tilde{\epsilon} \\ 0 & 0 & 0 & 3\tilde{\epsilon} \end{bmatrix}.$$

That is, the values of the rightmost column are perturbed. For example, for $i \in P$, $\mathtt{val}_i(\{(1,4)\}) = 1/8 - \epsilon^4 - \tilde{\epsilon}$. Similarly, for agents in Q, the values of the bottom row are perturbed:

$$M + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\tilde{\epsilon} & -\tilde{\epsilon} & -\tilde{\epsilon} & 3\tilde{\epsilon} \end{bmatrix}.$$

It is easy to verify that the MMS guarantee of all agents is 1 by partitioning the items based off their rows (for agents in Q) or columns (for agents in P). Moreover, our construction ensures the unique MMS partition of the agents in P (where every subset has value 1) corresponds to the columns of M, and the unique MMS partition of the agents in Q corresponds to the rows of M. If we divide the goods by columns, one of the two agents in Q will end up with a bundle of goods worth at most $1 - \tilde{\epsilon}$ — which is less than his MMS guarantee of 1. Similarly, if we divide the goods by rows, one of the agents in P will receive a bundle worth only $1 - \tilde{\epsilon}$. Any other division will certainly fail assuming that $\tilde{\epsilon}$ is sufficiently small. ■

6.2.3 Proof of Theorem 6.2.1 for $n \geq 4$

With the illustrative example of $n = 4$ under our belt, we are now ready for the general case where $n \geq 4$. The crux of the argument is proving the existence of a matrix $M \in \mathbb{R}^{n \times n}$ with the following properties:

1. All entries are non-negative (i.e. $\forall i, j : M_{i,j} \geq 0$).
2. All entries of the last row and column are positive (i.e. $\forall i : M_{i,n}, M_{n,i} > 0$).
3. All rows and columns sum to 1 (i.e. $M\mathbf{1} = M^T\mathbf{1} = \mathbf{1}$).
4. Define M^+ as the set of all positive entries in M. Then if we wish to partition M^+ into n subsets that sum to exactly 1 then our partition must correspond to the rows of M or the columns of M.

To begin, let $S \in \mathbb{R}^{n \times n}$ be the following matrix.

$$\begin{bmatrix} \frac{2^{n-1}-1}{2^{n-1}} & 0 & 0 & \cdots & 0 & 0 & \frac{1}{2^{n-1}} \\ 0 & \frac{2^{n-2}-1}{2^{n-2}} & 0 & \cdots & 0 & 0 & \frac{1}{2^{n-2}} \\ 0 & 0 & \frac{2^{n-3}-1}{2^{n-3}} & \cdots & 0 & 0 & \frac{1}{2^{n-3}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2^{n-1}} & \frac{1}{2^{n-2}} & \frac{1}{2^{n-3}} & \cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{2^{n-1}} \end{bmatrix} \quad \text{i.e. } S_{i,j} = \begin{cases} \frac{2^{n-i}-1}{2^{n-i}} & \text{if } i = j \neq n \\ \frac{1}{2^{n-j}} & \text{if } i = n \text{ and } j \neq n \\ \frac{1}{2^{n-i}} & \text{if } j = n \text{ and } i \neq n \\ \frac{1}{2^{n-1}} & \text{if } i = j = n \\ 0 & \text{otherwise.} \end{cases}$$

Now for $\epsilon \approx 0$ where $\epsilon > 0$, and for all $i \in [n-2]$, let $r_i = \epsilon^{2n-2i-2}$, and $c_i = \epsilon^{2n-2i-3}$. Specifically, this implies:

$$0 < r_1 \ll c_1 \ll r_2 \ll c_2 \ll \ldots \ll r_{n-2} \ll c_{n-2} = \epsilon \approx 0.$$

Furthermore, let $T \in \mathbb{R}^{n\times n}$ be the matrix given by (where we will define x, y, z and the u_i, and v_i below):

$$\begin{bmatrix} 0 & v_1 & 0 & \cdots & 0 & 0 & -r_1 \\ u_1 & 0 & v_2 & \cdots & 0 & 0 & -r_2 \\ 0 & u_2 & 0 & \cdots & 0 & 0 & -r_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & v_{n-2} & -r_{n-2} \\ 0 & 0 & 0 & \cdots & u_{n-2} & 0 & -y \\ -c_1 & -c_2 & -c_3 & \cdots & -c_{n-2} & -x & z \end{bmatrix} \text{ i.e. } T_{i,j} = \begin{cases} u_j & \text{if } i = j+1 \text{ and } j \le n-2 \\ v_i & \text{if } j = i+1 \text{ and } i \le n-2 \\ -r_i & \text{if } j = n \text{ and } i \le n-2 \\ -c_j & \text{if } i = n \text{ and } j \le n-2 \\ -x & \text{if } i = n \text{ and } j = n-1 \\ -y & \text{if } i = n-1 \text{ and } j = n \\ z & \text{if } i = j = n \\ 0 & \text{otherwise.} \end{cases}$$

Note the only nonzero entries are on the first diagonals above and below the main diagonal, and the last row and column.

Now assign positive values to the $u_i, v_i, x, y,$ and z such that all rows and columns sum to zero. A bit of arithmetic then gives:

$$\begin{aligned} u_i &= \left(\sum_{j \le i, j \equiv i \pmod 2} c_j \right) - \left(\sum_{j \le i, j \not\equiv i \pmod 2} r_j \right) \approx c_i \\ v_i &= \left(\sum_{j \le i, j \equiv i \pmod 2} r_j \right) - \left(\sum_{j \le i, j \not\equiv i \pmod 2} c_j \right) \approx r_i \\ x &= v_{n-2} \approx r_{n-2} \\ y &= u_{n-2} \approx c_{n-2} \\ z &= \left(\sum_{j \le n-2, j \equiv n \pmod 2} r_j + c_j \right) \approx c_{n-2}. \end{aligned}$$

Now define $M = S + T$ and $M^+ = \{(i,j) \mid M_{i,j} \neq 0\}$. Moreover, for a set $X \subseteq M^+$, let $\sum X = \sum_{(i,j)\in X} M_{i,j}$. Then we see for sufficiently small ϵ that the following properties hold.

[P1] $M_{i,j} \ge 0$ and if $S_{i,j} \neq 0$ or $T_{i,j} \neq 0$, then $M_{i,j} > 0$.

[P2] $M_{i,j} \approx S_{i,j}$.

[P3] All rows and columns sum to 1 (i.e. $M\mathbf{1} = M^T\mathbf{1} = \mathbf{1}$).

[P4] $\forall i \in [n-1]$ if we have $X \subseteq M^+$ s.t. $(i,i) \in X$ and $\sum X = 1$ then exactly one of the following is true:

(a) $(i,n) \in X$.

(b) $(n,i) \in X$.

(c) $(1,n),(2,n),\ldots,(i-1,n),(n,n) \in X$.

(d) $(n,1),(n,2),\ldots,(n,i-1),(n,n) \in X$.

(e) $\exists j,k < i$ s.t. $(j,n),(n,k) \in X$.

This is easy to see when we take note that $M \approx S$ by [P2].

[P5] If $X \subseteq M^+$ s.t. $\sum X = r_i$, then $X = \{(i,i-1),(i,i+1)\}$.

[P6] If $X \subseteq M^+$ s.t. $\sum X = c_i$, then $X = \{(i-1,i),(i+1,i)\}$.

[P7] If $X \subseteq M^+$ s.t. $\sum X = x$, then $X = \{(n-2,n-1)\}$.

[P8] If $X \subseteq M^+$ s.t. $\sum X = y$, then $X = \{(n-1,n-2)\}$.

We now make a key observation with respect to M.

Lemma 6.2.2. *Suppose $X_1,\ldots,X_n$ is a partition of M^+ such that $\sum X_i = 1$ for all i. Then for sufficiently small ϵ, the partition must correspond to the rows of M or the columns of M.*

Proof. Let us first consider the subset in the partition which includes $(1,1)$. WLOG assume this is X_1. We wish to prove that X_1 is either:

1. the first row = $\{(1,1),(1,2),(1,n)\}$, or
2. the first column = $\{(1,1),(2,1),(n,1)\}$.

By [P4] we see that exactly one of $(n,1)$, $(1,n)$, and (n,n) must be part of X_1.

1. Suppose $(n,n) \in X_1$. Then $\sum X_1 \geq M_{1,1} + M_{n,n} = 1 + z > 1$. This is therefore impossible.
2. Suppose $(1,n) \in X_1$. As $M_{1,1} + M_{1,n} = 1 - r_1$ we see that by [P5] we must have $(1,2) \in X_1$. Then X_1 corresponds to the first row.
3. Suppose $(n,1) \in X_1$. As $M_{1,1} + M_{n,1} = 1 - c_1$ we see that by [P6] we must have $(2,1) \in X_1$. Then X_1 corresponds to the first column.

Now suppose we wish to find a partition as in the lemma's statement such that the first $i-1$ rows are in the partition where $i \in \{2,\ldots,n\}$. Then we claim row i must be in the partition as well. Importantly, this implies that if the first row is to be in the partition, then the partition must be the rows.

We first consider the case where $i \leq n-1$. Let X_i denote the subset in the partition that includes (i,i). By [P4] we see that we must have one of the following.

1. $(i,n) \in X_i$.
   If $i \leq n-2$ we find that $M_{i,i} + M_{i,n} = -r_i$ and so by [P5] we have $(i,i-1),(i,i+1) \in X_i$. We therefore find that $X_i = \{(i,i-1),(i,i),(i,i+1),(i,n)\}$. On the other hand, if $i = n-1$ we find that $M_{i,i} + M_{i,n} = -y$ and so by [P8] we have $(n-1,n-2) \in X_i$. Thus $X_i = \{(n-1,n-2),(n-1,n-1),(n-1,n)\}$. In either case X_i is the i^{th} row.
2. $(n,i) \in X_i$.
   If $i \leq n-2$ we find that $M_{i,i} + M_{n,i} = -c_i$ and so by [P6] we have $(i-1,i) \in X_i$. But $(i-1,i)$ is in a previous row, which by our assumption is already assigned to a subset in the partition. On the other hand, if $i = n-1$ we have $M_{i,i} + M_{n,i} = -x$ and so by [P7] we

have $(n-2, n-1) \in X_i$. Similarly to before, this element is in a previous row and thus is already assigned to a subset in the partition.

3. $(1,n), (2,n), \ldots, (i-1,n), (n,n) \in X_i$.
   As $(1,n)$ is in a previous row it is already assigned to a subset in the partition.
4. $(n,1), (n,2), \ldots, (n,i-1), (n,n) \in X_i$.
   This is impossible because

$$\begin{aligned}\sum X_i &\geq M_{i,i} + M_{n,1} + M_{n,2} + \ldots + M_{n,i-1} + M_{n,n} \\ &= 1 - r_1 - r_2 - \ldots - r_{i-1} + z \\ &= 1 + r_i + r_{i+1} + \ldots + r_{n+2} + y \\ &> 1.\end{aligned}$$

5. $\exists j,k < i$ s.t. $M_{j,n}, M_{n,k} \in X_i$.
   As (j,n) is in a previous row it is already assigned to a subset in the partition.

Next, suppose $i = n$. In this case, since we are only allowed n subsets in this partition, all remaining entries (i.e. the last row) must be in the last set. By [P3] we know this last row sums to 1. We therefore have shown that if the first row is in the partition, then the partition simply corresponds to the rows. A similar argument gives an analogous result for columns. As the first row or first column must be a subset in the partition (namely as X_1) we are done. ■

To show the $n \geq 4$ counterexample, we now consider our construction through the lens of MMS allocations. We first show that there exists a set of $5n-6$ such goods for $n \geq 4$.

Partition the n agents into two groups P and Q such that $|P|, |Q| \geq 2$ and let $\mathcal{M} = M^+$. Note that there are $|M^+| = 5n-6$ such goods. For $k \in P$, we define:

$$\mathtt{val}_k(\{(i,j)\}) = \begin{cases} M_{i,j} & \text{if } j < n \\ M_{i,j} - \tilde{\epsilon} & \text{if } j = n \text{ and } i < n \\ M_{i,j} + (n-1)\tilde{\epsilon} & \text{if } j = n \text{ and } i = n \end{cases}$$

and similarly, for $k \in Q$, let

$$\mathtt{val}_k(\{(i,j)\}) = \begin{cases} M_{i,j} & \text{if } i < n \\ M_{i,j} - \tilde{\epsilon} & \text{if } i = n \text{ and } j < n \\ M_{i,j} + (n-1)\tilde{\epsilon} & \text{if } i = n \text{ and } j = n \end{cases}$$

where $\tilde{\epsilon} > 0$ and is small enough to ensure all $\mathtt{val}_k(\{(i,j)\}) \geq 0$. That is, the agent valuations are defined by the entries of M aside from perturbations on the last column for agents in P and on the last row for agents in Q.

As all agents in P (respectively Q) can partition the goods into columns (respectively rows) such that the value of each subset in the partition is exactly 1, the MMS guarantee of all agents in P (respectively Q) must be 1.

Next, let us consider an allocation of the goods. Lemma 6.2.2 tells us that if the $\mathtt{val}_k(\{(i,j)\})$ were exactly equal to the $M_{i,j}$ there are only two ways to allocate the goods such that every subset in the partition has value 1 (i.e. we get an MMS allocation): via the rows or via the columns. But note that the alteration to the value of a good (i, j) from $M_{i,j}$ is at most $(n-1)\tilde{\epsilon}$ and indeed no subset of goods can have its total value altered by more than $(n-1)\tilde{\epsilon}$ for any agent. Therefore, we claim that if we wish to have any hope of achieving an MMS allocation we must still partition according to the rows or columns (assuming $\tilde{\epsilon}$ is sufficiently small). To see this, define

$$\gamma = \max_{(X_1,\ldots,X_n)\in\mathcal{X}} \min_{i\in\mathcal{N}} \sum X_i$$

where $\mathcal{X}$ is the set of partitions of M^+ excluding the rows and the columns. Importantly, via Lemma 6.2.2 and the finite nature of $\mathcal{X}$ we know that $\gamma < 1$. Now suppose $\tilde{\epsilon} < \frac{1-\gamma}{n-1}$. Then for any allocation that did not correspond to the rows or columns some agent must have value at most $\gamma + (n-1)\tilde{\epsilon} < 1$. This proves the claim.

Now note that if we split via rows the agents of P will believe only the last row is worth at least 1 and all other rows are worth strictly less than 1. As there are at least two agents in P, not all agents can receive their MMS guarantee. A similar issue occurs when we split via the columns for the agents in Q. Therefore, there exists no MMS allocation in this setting.

We have just shown the result for $5n-6$ goods (for $n \geq 4$) and now set our sights on $3n+4$ goods. Let $\tilde{n} = \lceil (n+4)/2 \rceil \geq 4$. We know that we can find $5\tilde{n}-6$ goods that do not admit an MMS allocation for $\tilde{n}$ agents. Take this set of goods, and let there be n agents such that $\lfloor n/2 \rfloor$ agents are in group P and the remaining $\lceil n/2 \rceil$ are in group Q. Finally, add $n-\tilde{n}$ goods each of value 1 to all agents. Note that the number of goods is:

$$m = (5\tilde{n}-6) + (n-\tilde{n}) = 4\tilde{n} + n - 6 = 4\lceil (n+4)/2 \rceil + n - 6 \leq 3n + 4.$$

Further observe that:

$$n - \tilde{n} = n - \lceil (n+4)/2 \rceil = \lfloor n/2 \rfloor - 2.$$

Thus, we have that of the agents who did not receive any of the new $n-\tilde{n}$ items of value 1 there must be at least $|P| - (n-\tilde{n}) \geq 2$ agents in P. Similarly, there must be $|Q| - (n-\tilde{n}) \geq 2$ agents in Q. As we still must have at least 2 agents in both P and Q when we allocate the original $5\tilde{n}-6$ goods no MMS allocation exists. ■

6.3 Existence and Computation of Approximate MMS Allocations

To circumvent Theorem 6.2.1 we introduce a new notion of *approximate* maximin share guarantee: rather than asking for an allocation $A_1, \ldots, A_n$ such that $\mathtt{val}_i(A_i) \geq \mathtt{MMS}_i(n, \mathcal{M})$ for all $i \in \mathcal{N}$, we look for γ-approximate MMS allocations such that $\mathtt{val}_i(A_i) \geq \gamma \cdot \mathtt{MMS}_i(n, \mathcal{M})$ for some $\gamma > 0$.

To this end, recall that for all $k \in \mathbb{N}$, we denote

$$\gamma_k = \frac{2\lfloor k \rfloor_{odd}}{3\lfloor k \rfloor_{odd} - 1}.$$

Our main result is that γ_n-approximate MMS allocations always exist.

Theorem 6.3.1. *There always exists an allocation $A_1, \ldots, A_n$ such that for all $i \in \mathcal{N}$, $\texttt{val}_i(A_i) \geq \gamma_n \texttt{MMS}_i(n, \mathcal{M})$. Moreover, for every $\epsilon > 0$, an allocation $A_1, \ldots, A_n$ such that for all $i \in \mathcal{N}$, $\texttt{val}_i(A_i) \geq (1-\epsilon)\gamma_n \texttt{MMS}_i(n, \mathcal{M})$ can be computed in polynomial time in n and m.*

Paramount to the proof of Theorem 6.3.1 is Algorithm 13. The only part of the algorithm that is not elementary is Step 7(a), which says "repeat until no cycles exist". Intuitively, each time the bundles are rotated along a cycle, the total number of edges in the envy graph decreases, and therefore the cycle elimination process must terminate. This claim is formally established by Lipton et al. [65], who use it to show that an algorithm that essentially coincides with Steps 6 and 7 of Algorithm 13 achieves the α-envy-freeness property discussed in Section 6.1.3.

1. If there is an agent who believes any single item is worth γ_n of his MMS guarantee, give it to him and eliminate him and his item from all further consideration. Repeat until no such agent exists.
2. If only two agents remain, let one of the two agents produce a 2-MMS partition and have the other take his more preferred bundle. The remaining bundle is given to the agent who produced the partition and the algorithm ends.
3. In lexicographic order, give each remaining agent their most favored item not already given away or eliminated (breaking ties between items lexicographically).
4. In *reverse* lexicographic order, give each remaining agent their most favored item not already given away or eliminated (breaking ties between items lexicographically).
5. If a non-eliminated agent believes that his last received item (the one given to him in Step 4), in addition to any two items not already given out or eliminated, is worth γ_n of his MMS guarantee, then:
   (a) Exchange his current two items for these three items (his last received item remains with him).
   (b) Eliminate this agent and his three items.
   (c) Have all other remaining agents (the others who received items in steps 3 and 4) return their items.
   (d) Go to Step 3.
6. Create a directed *envy graph* $G = (V, E)$ where V represents the remaining agents and there is an edge (i, j) iff i believes his current bundle is worth strictly less than that of j.
7. Loop through the following until all items have been allocated.
   (a) If there is a cycle in G then eliminate it by having each agent in the cycle give his bundle to the agent before him in the cycle (and receive the bundle from the agent after him). Update the edges so that (i, j) exists iff i believes his current bundle is worth strictly less than that of j, as before. Repeat until no cycles exist.
   (b) As there is no cycle in G, there exists at least one agent who has no incoming edges. Give one of the items not already given out or eliminated to one of these agents.

Algorithm 13

We prove Theorem 6.3.1 in two steps. In Section 6.3.1, we show that Algorithm 13 produces a γ_n-approximate MMS allocation. The algorithm would clearly run in polynomial time, if it were given an oracle that can compute MMS partitions. In Section 6.3.2, we explain how to convert the algorithm to a polynomial time algorithm, at the cost of decreasing the MMS approximation ratio by ϵ.

6.3.1 Proof of Theorem 6.3.1: Existence

Fix the number of agents n, and denote $\gamma = \gamma_n$. Assume, for the sake of contradiction, that the existence claim in Theorem 6.3.1 is false. In particular, we have a counterexample where agent $i \in \mathcal{N}$ does not achieve the desired γ ratio of his MMS guarantee on the item set $\mathcal{M}$, when Algorithm 13 is executed on this instance. For notational convenience, further assume i's MMS guarantee is 1 in this instance (normalizing if necessary) and that values always refer to those of agent i unless otherwise specified.

Observation 6.3.2. *If i is eliminated at any point, then he achieves a γ fraction of his MMS guarantee (i.e. he receives a value of at least γ).*

Observation 6.3.3. *γ_k is a non-increasing function of k, and is in $(2/3, 3/4]$ for $k \geq 3$.*

Lemma 6.3.4. *If the set of agents eliminated in step 1 of the algorithm in our counterexample (where i fails to achieve a value of γ) is non-empty but does not contain i, then there exists a counterexample where no agents are eliminated in step 1.*

Proof. Let $\tilde{\mathcal{N}}$ be the set of agents remaining after step 1 in our counterexample, and $\tilde{\mathcal{M}}$ the set of items.

Now consider the execution of the algorithm on the set of agents $\tilde{\mathcal{N}}$ and items $\tilde{\mathcal{M}}$. Observe that upon completion of step 1, the executions on this instance and the original (i.e. the execution with $\mathcal{N}$ and $\mathcal{M}$) are equivalent. That is, the agents in $\tilde{\mathcal{N}}$ are given the same items in both instances.

Finally, consider i's value in the altered instance. As each eliminated agent took only one item and because a single item can only occupy a single bundle in an MMS partition, we have that

$$\mathtt{MMS}_i(|\tilde{\mathcal{N}}|, \tilde{\mathcal{M}}) \geq \mathtt{MMS}_i(|\mathcal{N}|, \mathcal{M}). \tag{6.1}$$

Recall that i's value on the altered instance is equal to his value on the original instance, which is less than $\gamma_{|\mathcal{N}|}\mathtt{MMS}_i(|\mathcal{N}|, \mathcal{M}) \leq \gamma_{|\tilde{\mathcal{N}}|}\mathtt{MMS}_i(|\tilde{\mathcal{N}}|, \tilde{\mathcal{M}})$, where the weak inequality follows from Equation (6.1) and Observation 6.3.3. Therefore, i does not achieve the desired $\gamma_{|\tilde{\mathcal{N}}|}$ ratio of his MMS guarantee on the altered instance. ■

Observation 6.3.5. *If the algorithm terminates on step 2, then all agents achieve the desired γ approximation.*

Importantly, Observation 6.3.2 and Lemma 6.3.4 suggest that we may safely assume that no agent is eliminated in step 1 in our counterexample — and, in fact, that i is not eliminated at any point. Observation 6.3.5 further allows us to assume that the algorithm does not terminate at step 2 and so, in addition, $n \geq 3$. With this in mind, we introduce the following notation.

- For $j \in \mathcal{N}$, Φ_j denotes the set of the two most valuable items (in i's view) that j possesses at the beginning of step 6 (breaking ties arbitrarily). In particular, if j is not eliminated in step 5 he has exactly two items, and otherwise he has exactly three.
- For $j \in \mathcal{N}$, Ψ_j denotes the bundle containing Φ_j upon completion of the entire algorithm (it is easy to check that the algorithm will never separate them). Note that j may not receive this bundle upon algorithm completion due to step 7, but all agents receive exactly one of these bundles.
- For $j \in \mathcal{N}$, v_j denotes $\mathtt{val}_i\left(\Phi_j\right)$.
- For $j \in \mathcal{N}$, V_j denotes $\mathtt{val}_i\left(\Psi_j\right)$.
- p denotes i's value for the item i received in the last iteration of step 3.
- q denotes i's value for the item i received in the last iteration of step 4. Note that $p + q = v_i$ and $p \geq q$.
- $\hat{i}$ denotes the index such that upon algorithm completion i receives bundle $\Psi_{\hat{i}}$. Note that we must have $V_{\hat{i}} < \gamma$.

With this notation we are now ready for the following key observations and lemmas.

Observation 6.3.6. *$v_j \leq V_j$ (since $v_j = \mathtt{val}_i\left(\Phi_j\right) \leq \mathtt{val}_i\left(\Psi_j\right) = V_j$).*

Observation 6.3.7. *During steps 6 and 7, i's value is non-decreasing (since i only exchanges his bundle for one he envies).*

Observation 6.3.8. *$p + q = v_i < \gamma$ (since by Observation 6.3.7 i must receive a value of at least v_i upon algorithm completion).*

Observation 6.3.9. *$q < \gamma/2$ (by Observation 6.3.8 and $q \leq p$).*

Lemma 6.3.10. *If $j \neq i$ is eliminated in step 5, then at most one of j's three items has value in $(q, p]$, and the others each have value at most q.*

Proof. When j is eliminated in step 5, he retains the item he received in step 4 and receives two others. The item he received in step 4 is clearly of value at most p, as otherwise i would have taken this in step 3. Similarly, the other two items each must be of value at most q, as i could have taken either in step 4. ■

Corollary 6.3.11. *If $j \neq i$ is eliminated in step 5, then $V_j \leq p + 2q < \gamma + q$ (By Lemma 6.3.10 and Observation 6.3.8).*

Lemma 6.3.12. *If $\Phi_j \subsetneq \Psi_j$ then $v_j < \gamma$ and $V_j < \gamma + q$.*

Proof. If $\Phi_j \subsetneq \Psi_j$ then we must have one of two cases.

1. *j is eliminated in step 5.*
   Lemma 6.3.10 and Observation 6.3.8 shows us that $v_j \leq p + q < \gamma$ and Corollary 6.3.11 shows us that $V_j < \gamma + q$.
2. *During steps 6 and 7, the bundle initially denoted by Φ_j and ending as Ψ_j received at least one item.*
   Let us consider the last time this bundle received an item. i must not have envied whomever held the bundle at the time, and therefore its value to i before the addition of the new item

must be less than γ. Thus, $v_j < \gamma$. Furthermore, the added item must have value at most q (as otherwise i would have selected this item in step 4) and so we have $V_j < \gamma + q$.

■

Corollary 6.3.13. *If $v_j \geq \gamma$, then $\Psi_j = \Phi_j$ and thus $V_j = v_j$ as well (by Lemma 6.3.12).*
Lemma 6.3.14. *It holds that $v_j \leq V_j \leq \max(v_j, \gamma + q)$.*

Proof. $v_j \leq V_j$ is true by Observation 6.3.6. Regarding the second inequality, if $\Phi_j = \Psi_j$ then we clearly have $V_j = v_j \leq \max(v_j, \gamma + q)$. Otherwise Lemma 6.3.12 applies and we see that $V_j < \gamma + q \leq \max(v_j, \gamma + q)$. ■

Lemma 6.3.15. *If $v_j \leq \gamma + q$ we have $V_j \leq \gamma + q$.*

Proof. If $\Phi_j \subsetneq \Psi_j$, then Lemma 6.3.12 applies. Otherwise, $V_j = v_j$ which gives the result. ■

For the following observations, recall that agents choose in lexicographic order (increasing index) in Step 3, and in reverse lexicographic order (decreasing index) in Step 4.
Observation 6.3.16. *If $j \leq i$ is not eliminated, then the more valuable of Φ_j's two items (in i's view) i values less than γ (as otherwise i would have taken this item in step 1) and the other i values at most q (as otherwise i would have taken this item in step 4). This further implies $v_j < \gamma + q$.*
Observation 6.3.17. *If $j > i$ is not eliminated, then each of the two items in Φ_j must have value at most p to i (as otherwise i would have taken one of these items in step 3). This further implies $v_j \leq 2p$.*
Lemma 6.3.18. *For all $j \leq i$ we have $v_j \leq V_j \leq \gamma + q$.*

Proof. If j is eliminated in step 5, then Corollary 6.3.11 applies and we see that $v_j \leq \gamma + q$. Otherwise, by Observation 6.3.16 we still have that $v_j \leq \gamma + q$. Combining this with Lemma 6.3.14 gives the result. ■

Lemma 6.3.19. *For all $j > i$ we have $v_j \leq V_j \leq \max(2p, \gamma + q)$.*

Proof. If j is eliminated in step 5, then Corollary 6.3.11 applies and we see that $v_j \leq \gamma + q$ as before. Otherwise, by Observation 6.3.17 we have that $v_j \leq 2p$. Combining this with Lemma 6.3.14 gives the result. ■

Lemma 6.3.20. *It holds that $p < 1/2$.*

Proof. Assume for contradiction that $p \geq 1/2$. Let $S = \{j \mid v_j > 1\}$. We make the following observations for each $j \in S$:

- $j > i$.
  If $j \leq i$ we have:

$$\begin{aligned} v_j &\leq \gamma + q \text{ (by Lemma 6.3.18)} \\ &< \gamma + (\gamma - p) \text{ (since } p + q < \gamma) \\ &= 2\gamma - p \\ &\leq 2(3/4) - 1/2 \text{ (by Observation 6.3.3 and our assumption } p \geq 1/2) \\ &= 1. \end{aligned}$$

  Thus, if $j \leq i$ we have $v_j \leq 1$ and so we cannot have that $j \in S$.
- $\Psi_j = \Phi_j$.
  This follows from Corollary 6.3.13 and noting that $\gamma < 1$.
- *There are only two items in Ψ_j and each has value at most p.*
  Since $\Psi_j = \Phi_j$, we have that Ψ_j has only two items. Furthermore, as we know that $j > i$, the two items in Ψ_j each have value at most p by Observation 6.3.17.

Now let $T = \left\{ j \mid v_j \in [\gamma, 1] \right\}$. Observe that for all $j \in T$, we have $\Psi_j = \Phi_j$ and so there are only two items in Ψ_j (by Corollary 6.3.13 and noting that $\gamma < 1$). Now consider the following algorithm (which we use only as a tool in our proof and not as a useful algorithm in and of itself).

---

1. Let $P = \{A_1, \ldots, A_n\}$ be some MMS partition for i.
2. Flag the item i receives in the last invocation of step 3 (which is worth p).[3]
3. Flag the $2|S|$ items corresponding to the Ψ_j for $j \in S$.
4. While $T \neq \emptyset$:
   (a) Remove some $t \in T$.
   (b) Denote the two items corresponding to Ψ_t by x and y.
   (c) Denote by G the bundle in P that x belongs to.
   (d) Denote by H the bundle in P that y belongs to.
   (e) Flag x and y.
   (f) If $G \neq H$ replace G and H with $\{x, y\}$ and $(G \cup H) \setminus \{x, y\}$ in P. That is, if $G \neq H$ we replace P with $(P \setminus \{G, H\}) \cup \{\{x, y\}, (G \cup H) \setminus \{x, y\}\}$.

We claim that an invariant of the loop in the algorithm (and therefore holds upon algorithm completion) is that for all $G \in P$:

- if there are zero flagged items in G, then i values G at least at 1.
- if there is exactly one flagged item in G, then i values the non-flagged items of G at least at $1 - p$.

[3]The concept of flagging can be thought of as inclusion in some *flag set*, but we find this approach intuitively clearer.

As initially P is an MMS partition, we know that before we enter the loop for the first time, any bundle of P without any flagged items must have a value of at least i's MMS guarantee, which is 1. Furthermore, as all of the $2|S|+1$ items initially flagged must have value at most p, any bundle with exactly one flagged item must have value at least $1-p$ for the non-flagged items. Our invariant thus holds initially.

During a loop iteration if we have that $G = H$ then since flagging x and y forces $G(= H)$ to have at least two flagged items, our invariant continues to hold vacuously. It therefore only remains to show that when we replace G, H with $\{x,y\}, (G \cup H) \setminus \{x,y\}$ our invariant still holds. As the set $\{x,y\}$ contains two flagged items, we need not show anything of this set. We focus now on the set $(G \cup H) \setminus \{x,y\}$.

During a loop iteration we have the following cases:

- $(G \cup H) \setminus \{x,y\}$ has zero flagged items.
  In this case, both G and H had zero flagged items before the flagging of x and y and therefore, they each have value at least 1. Thus, the non-flagged items in $(G \cup H) \setminus \{x,y\}$ have value at least

  $$1 + 1 - \mathtt{val}_i(\{x,y\}) \geq 1 + 1 - 1 = 1$$

  where we have used the fact that $\mathtt{val}_i(\{x,y\}) = v_t$ for some $t \in T$ and therefore by the definition of T is at most 1.
- $(G \cup H) \setminus \{x,y\}$ has exactly one flagged item.
  In this case, exactly one of $G \setminus \{x\}$ and $H \setminus \{y\}$ has a flagged item (and exactly one flagged item). Then we have that the non-flagged items of $(G \cup H) \setminus \{x,y\}$ have value at least

  $$1 + 1 - p - \mathtt{val}_i(\{x,y\}) \geq 1 + 1 - p - 1 = 1 - p.$$

- $(G \cup H) \setminus \{x,y\}$ has two or more flagged items.
  In this case, we need not prove any property of the bundle.

This proves the loop invariant.

Once this algorithm completes, we have that for all $j \in S \cup T$, both of the two items in Ψ_j are flagged, as is i's first item received (which is worth p). If we let $k = |S \cup T|$, then we have that the total value of all non-flagged items is:

$$\begin{aligned}
-p + \sum_{j \notin S \cup T} V_j &= V_{\hat{i}} - p + \sum_{j \notin S \cup T \cup \{\hat{i}\}} V_j \\
&< \gamma - p + (n-k-1)(\gamma + q) \text{ (by } V_{\hat{i}} < \gamma \text{ and Lemma 6.3.15)} \\
&< \gamma - p + (n-k-1)(\gamma + (\gamma - p)) \text{ (since } p + q < \gamma) \\
&= (2n - 2k - 1)\gamma - (n-k)p.
\end{aligned}$$

We will now contradict this statement by in fact demonstrating that the total value of all non-flagged items must simultaneously be at least $(2n-2k-1)\gamma - (n-k)p$ — thus completing the proof.

Denote by α_j the number of bundles of the final partition with exactly j flagged items. Then the total value of non-flagged items must be at least $\alpha_0 + \alpha_1(1-p)$ due to the loop invariant. Importantly, by counting the number of flagged items, we also have that:

$$\begin{aligned} 2k+1 = \sum_{j \geq 1} j\alpha_j \geq \alpha_1 + 2\sum_{j\geq 2}\alpha_j = \alpha_1 + 2(n - \alpha_0 - \alpha_1) \\ \Rightarrow \alpha_1 \geq 2n - 2k - 1 - 2\alpha_0. \end{aligned}$$

Thus, to prove the desired contradiction it suffices to show that the solution to the following optimization problem is at least 0.

$$\begin{aligned} \min_{\alpha_0,\alpha_1,p} \; & \alpha_0 + \alpha_1(1-p) - (2n-2k-1)\gamma + (n-k)p \\ \text{s.t. } & \alpha_1 \geq 2n - 2k - 1 - 2\alpha_0 \\ & \alpha_0 \geq \max(0, n - 2k - 1). \end{aligned}$$

As $p < \gamma < 1$ we have that $1 - p > 0$ and so it is best to minimize α_1 under the constraint. That is, the constraint should be tight at the optimal solution. We can therefore assume $\alpha_1 = 2n - 2k - 1 - 2\alpha_0$ and with a bit of arithmetic we arrive at the following equivalent optimization problem.

$$\begin{aligned} \min_{\alpha_0,p} \; & \alpha_0(2p-1) + (2n-2k-1)(1-p-\gamma) + (n-k)p \\ \text{s.t. } & \alpha_0 \geq \max(0, n-2k-1). \end{aligned}$$

As $p \geq 1/2$ we have that $2p - 1 \geq 0$ and so it is also best to minimize α_0 — the number of bundles with zero flagged items. Thus, we have that $\alpha_0 = \max(0, n-2k-1)$ and so we have the further reduced optimization problem:

$$\min_p \max(0, n-2k-1) \cdot (2p-1) + (2n-2k-1)(1-p-\gamma) + (n-k)p. \tag{6.2}$$

In regards to the final variable of our optimization, p, we see that our objective is linear and therefore we need only consider the extreme values of $1/2$ and γ. We are left with three cases to analyze — each of which is a matter of straightforward computation.

- $p = 1/2$.
  The objective of (6.2) is

$$\begin{aligned} & (2n-2k-1)(1/2-\gamma) + (n-k)(1/2) \\ & = (2n-2k-1)\left(\frac{1}{2} - \frac{2\lfloor n \rfloor_{odd}}{3\lfloor n \rfloor_{odd} - 1}\right) + (n-k)/2 \\ & = \frac{n\lfloor n \rfloor_{odd} - 3n + \lfloor n \rfloor_{odd} + 1 - k(\lfloor n \rfloor_{odd} - 3)}{2(3\lfloor n \rfloor_{odd} - 1)}. \end{aligned}$$

As the denominator is always greater than 0, to show this is at least 0 it suffices to show the numerator itself is at least 0. The numerator is

$$\begin{aligned}
&n\lfloor n \rfloor_{odd} - 3n + \lfloor n \rfloor_{odd} + 1 - k(\lfloor n \rfloor_{odd} - 3) \\
&\geq n\lfloor n \rfloor_{odd} - 3n + \lfloor n \rfloor_{odd} + 1 \\
&\qquad - (n-1)(\lfloor n \rfloor_{odd} - 3) \text{ (since } \lfloor n \rfloor_{odd} \geq 3 \text{ and } k \leq n-1) \\
&= 2\lfloor n \rfloor_{odd} - 2 \\
&\geq 2(3) - 2 \\
&> 0.
\end{aligned}$$

- *$n \geq 2k+1$ and $p = \gamma$.*
  The objective of (6.2) is

$$\begin{aligned}
&(n-2k-1)(2\gamma - 1) + (2n-2k-1)(1-2\gamma) + (n-k)\gamma \\
&= n - (n+k)\gamma \\
&= n - (n+k)\left(\frac{2\lfloor n \rfloor_{odd}}{3\lfloor n \rfloor_{odd} - 1}\right) \\
&= \frac{n\lfloor n \rfloor_{odd} - 2k\lfloor n \rfloor_{odd} - n}{3\lfloor n \rfloor_{odd} - 1}.
\end{aligned}$$

  As before, it suffices to show the numerator is at least 0. When n is odd we have that the numerator is

$$n^2 - 2kn - n = n(n - (2k+1)) \geq 0.$$

  If, on the other hand, n is even we have that the numerator is

$$\begin{aligned}
n(n-1) - 2k(n-1) - n &= (n-1)(n-(2k+1)) - 1 \\
&\geq (n-1) - 1 \text{ (because } n \text{ is even } n - (2k+1) \geq 1) \\
&\geq 0.
\end{aligned}$$

- *$n \leq 2k$ and $p = \gamma$.*
  The objective of (6.2) is

$$\begin{aligned}
(2n-2k-1)(1-2\gamma) + (n-k)\gamma &= (2n-2k-1) + (-3n+3k+2)\gamma \\
&= (2n-2k-1) + (-3n+3k+2)\left(\frac{2\lfloor n \rfloor_{odd}}{3\lfloor n \rfloor_{odd} - 1}\right) \\
&= \frac{\lfloor n \rfloor_{odd} - 2n + 2k + 1}{3\lfloor n \rfloor_{odd} - 1}.
\end{aligned}$$

  As before, it suffices to show the numerator is at least 0; it is at least

$$\begin{aligned}
(n-1) - 2n + 2k + 1 &= -n + 2k \\
&\geq 0.
\end{aligned}$$

■

Lemma 6.3.21. *It holds that $q > \frac{n}{n-1}(1-\gamma)$.*

Proof. Suppose for purposes of contradiction that $q \le \frac{n}{n-1}(1-\gamma)$. For all $j \in \mathcal{N}$ we have:

$$\begin{aligned}
V_j &\le \max(2p, \gamma + q) \text{ (by Lemmas 6.3.18 and 6.3.19)} \\
&\le \max\left(2(1/2), \gamma + \frac{n}{n-1}(1-\gamma)\right) \text{ (by Lemma 6.3.20 and our assumption on } q) \\
&= \max\left(1, \frac{n-\gamma}{n-1}\right) \\
&= \frac{n-\gamma}{n-1} \text{ (since } \gamma < 1).
\end{aligned}$$

We then see that:

$$\sum_{j=1}^{n} V_j = V_{\hat{i}} + \sum_{j \neq \hat{i}} V_j < \gamma + (n-1)\frac{n-\gamma}{n-1} = n.$$

That $\sum_{j=1}^{n} V_j < n$ clearly contradicts that i's MMS guarantee is 1. ■

Lemma 6.3.22. *n is even.*

Proof. Suppose for purposes of contradiction that n is odd. By Lemma 6.3.21, we must have that:

$$\begin{aligned}
q &> \frac{n}{n-1}(1-\gamma) \\
&= \frac{n}{n-1}\left(1 - \frac{2n}{3n-1}\right) \text{ (by the definition of } \gamma) \\
&= \frac{n}{3n-1} \\
&= \gamma/2.
\end{aligned}$$

This clearly contradicts Observation 6.3.9's statement that $q < \gamma/2$. ■

Corollary 6.3.23. $\gamma = \frac{2(n-1)}{3(n-1)-1}$ *(by the definition of γ and Lemma 6.3.22).*
Lemma 6.3.24. *It holds that $\frac{n}{n-1}(1-\gamma) \ge 1/3$.*

Proof.

$$\begin{aligned}
\frac{n}{n-1}(1-\gamma) &= \frac{n}{n-1}\left(1 - \frac{2(n-1)}{3(n-1)-1}\right) \text{ (by Corollary 6.3.23)} \\
&= \frac{1}{3}\frac{3n^2-6n}{3n^2-7n+4} \\
&\ge \frac{1}{3}\frac{3n^2-6n}{3n^2-7n+n} \text{ (since } n \ge 4 \text{ by Lemma 6.3.22 and } n \ge 3) \\
&= 1/3.
\end{aligned}$$

■

Let us now take this moment to introduce the following notation.

- X: the set of agents who are eliminated (in step 5).
- Y: the set of agents $j \notin X$ and $j < i$ where $v_j \geq \gamma$.
- Z: the set of agents $j \notin X$ and $j > i$ where $v_j \geq \gamma$.
- $x = |X|$, $y = |Y|$, and $z = |Z|$.

Observation 6.3.25. *$i \notin X \cup Y \cup Z$ (since i is not eliminated).*

Observation 6.3.26. *For all $j \notin Z$ we have $V_j \leq \gamma + q$ (by Corollary 6.3.11 and Lemmas 6.3.15 and 6.3.18).*

Observation 6.3.27. *For all $j \in Z$ we have $V_j = v_j \leq 2p < 1$ (by Corollary 6.3.13, Observation 6.3.17, and Lemma 6.3.20).*

Lemma 6.3.28. *$\hat{i} \notin X \cup Y \cup Z$.*

Proof. If $\hat{i} \in X$, then $\Psi_{\hat{i}}$ must go to the eliminated agent $\hat{i}$ (who is thus not i). If $\hat{i} \in Y \cup Z$, then i would receive a value $= V_{\hat{i}} \geq v_{\hat{i}} \geq \gamma$. ■

Lemma 6.3.29. *It holds that $Z = \emptyset$ (i.e. $z = 0$).*

Proof. Assume for purposes of contradiction that $z \geq 1$. Then we have the following.

$$\begin{aligned}
\sum_{j=1}^{n} V_j &= \sum_{j \in Z} V_j + V_{\hat{i}} + \sum_{j \in \mathcal{N} \setminus (Z \cup \{\hat{i}\})} V_j \\
&< \sum_{j \in Z} 1 + \gamma + \sum_{j \in \mathcal{N} \setminus (Z \cup \{\hat{i}\})} (\gamma + q) \text{ (by Observations 6.3.26 and 6.3.27 and } V_{\hat{i}} < \gamma) \\
&= z + \gamma + (n - z - 1)(\gamma + q) \\
&< z + \gamma + (n - z - 1)(\gamma + \gamma/2) \text{ (by Observation 6.3.9)} \\
&= (1 - 3\gamma/2)z + (3n - 1)(\gamma/2) \\
&< (1 - 3\gamma/2) + (3n - 1)(\gamma/2) \text{ (since } \gamma > 2/3 \text{ by Observation 6.3.3 and } z \geq 1) \\
&= \left(1 - \frac{3}{2} \cdot \frac{2(n-1)}{3(n-1) - 1}\right) + \frac{3n-1}{2} \cdot \frac{2(n-1)}{3(n-1) - 1} \text{ (by Corollary 6.3.23)} \\
&= n.
\end{aligned}$$

That $\sum_{j=1}^{n} V_j < n$ clearly contradicts that i's MMS guarantee is 1. ■

Lemma 6.3.30. *It holds that $x + y > n - 3$.*

Proof. Assume for purposes of contradiction that $x + y \leq n - 3$. Let us consider the $n - x - y - 1$ values V_j for $j \in \mathcal{N} \setminus \left(X \cup Y \cup \{\hat{i}\}\right)$. As i was not eliminated in step 5, i must believe the two most valuable items not given out or eliminated at the beginning of step 6 sum to value $< \gamma - q$. This statement, along with the fact $n - x - y - 1 \geq 2$ (since we are assuming $x + y \leq n - 3$), implies i's value for the $n - x - y - 1$ largest items not given out or eliminated at the beginning of step 6 is at most $(n - x - y - 1)(\gamma - q)/2$. Simultaneously, we know for all $j \in \mathcal{N} \setminus (X \cup Y \cup Z \cup \{\hat{i}\}) = \mathcal{N} \setminus (X \cup Y \cup \{\hat{i}\})$ (we have used Lemma 6.3.29 for the equality), the value of the bundle that at step 6 starts as Φ_j and upon algorithm completion becomes Ψ_j before it receives its last item is less than γ (as otherwise, i would envy this bundle). This yields:

$$\sum_{j\in\mathcal{N}\setminus(X\cup Y\cup\{\hat{i}\})} V_j < (n-x-y-1)(\gamma+(\gamma-q)/2).$$

Noting that $V_{\hat{i}} < \gamma$ and for all $j \in X \cup Y$ we have $V_j < \gamma + q$ by Corollary 6.3.11 and Lemma 6.3.18, we then get:

$$\begin{aligned}\sum_{j=1}^{n} V_j &= \sum_{j\in X\cup Y} V_j + V_{\hat{i}} + \sum_{j\in\mathcal{N}\setminus(X\cup Y\cup\{\hat{i}\})} V_j \\ &< (x+y)(\gamma+q) + \gamma + (n-x-y-1)(\gamma+(\gamma-q)/2).\end{aligned}$$

We claim this last quantity, a function which we will call V, is smaller than n for the relevant values of q — i.e. $q \in \left(\frac{n}{n-1}(1-\gamma), \gamma/2\right)$ (the relevant values are determined by Observation 6.3.9 and Lemma 6.3.21). Indeed, observe that V is a linear function in q. Moreover, note that since $\gamma \le 3/4$ (by Observation 6.3.3) we have:

$$\gamma/3 \le (3/4)/3 = 1 - 3/4 \le 1 - \gamma \le \frac{n}{n-1}(1-\gamma).$$

This implies that the domain $\left(\frac{n}{n-1}(1-\gamma), \gamma/2\right)$ is contained in $[\gamma/3, \gamma/2]$. Thus, to show the desired inequality $V < n$ it suffices to show the inequality for $q \in \{\gamma/3, \gamma/2\}$:

$$\begin{aligned}V(\gamma/3) &= (x+y)(4\gamma/3) + \gamma + (n-x-y-1)(\gamma+\gamma/3) \\ &= (4\gamma/3)(n-1/4) \\ &\le (4(3/4)/3)(n-1/4) \text{ (by Observation 6.3.3)} \\ &= n - 1/4 \\ &< n. \\ V(\gamma/2) &= (x+y)(3\gamma/2) + \gamma + (n-x-y-1)(\gamma+\gamma/4) \\ &= (\gamma/4)(5n+x+y-1) \\ &\le (\gamma/4)(5n+(n-3)-1) \text{ (since we are assuming } x+y \le n-3) \\ &= (\gamma/2)(3n-2) \\ &= \frac{3n-2}{2}\frac{2(n-1)}{3(n-1)-1} \text{ (by Corollary 6.3.23)} \\ &= n - \frac{n-2}{3n-4} \\ &< n.\end{aligned}$$

We can therefore conclude that $\sum_{j=1}^{n} V_j < V < n$ — contradicting that i's MMS guarantee is 1. ■

Lemma 6.3.31. *It holds that $x + y \neq n - 2$.*

Proof. Assume for purposes of contradiction that $x + y = n - 2$. Consider the set $\mathcal{H}$ of items composed of the following.

- *The items in all of the Ψ_j for all $j \in X$ (equivalently, the items that go to the agents in X).* There are $3x$ such items, and by Lemma 6.3.10 we know i values all of these items at a value of at most q, except for at most x of them which may have value in $(q, p]$.
- *The items in all of the Ψ_j for all $j \in Y$.* By Corollary 6.3.13 there are $2y$ such items, but we will imagine as if the y largest items (in i's view) are in fact two inseparable items — giving us instead $3y$ such items. Note that each such pair of inseparable items are of value $< \gamma$ and the other y items have value at most q by Observation 6.3.16.
- *The two items in Φ_i (which i values at p and q).*
- *The item i values most (breaking ties arbitrarily) among those not eliminated nor given out at the beginning of step 6.* Let Δ denote i's value of this item. Note that $\Delta \leq q$ as otherwise i would have taken this item in step 4.

Observe that any single item of $\mathcal{H}$ is of value $\leq p$ and any two items have value at most $\max(2p, \gamma) \leq \max(2(\gamma - q), \gamma) = 2(\gamma - q)$.

We are interested in the value of all items aside from these $3x + 3y + 2 + 1 = 3(n - 1)$ items, which we will denote by r. That is, $r = (\sum_{j \notin X \cup Y} V_j) - (p + q + \Delta)$. Now fix $A_1, \ldots, A_n$ to be some MMS partition for i. In each of the following four encompassing cases, we will demonstrate that $r \geq 2\gamma - p - q$.

1. *There exists an A_j that contains no items in $\mathcal{H}$.*

$$
\begin{aligned}
r &\geq \mathtt{val}_i\left(A_j\right) \\
&\geq 1 \text{ (since } i\text{'s MMS value is 1)} \\
&= 2(2(3/4) - 1) \\
&\geq 2(2\gamma - 1) \text{ (by Observation 6.3.3)} \\
&= 2(\gamma - (1 - \gamma)) \\
&\geq 2\left(\gamma - \frac{n}{n-1}(1 - \gamma)\right) \\
&\geq 2(\gamma - q) \text{ (by Lemma 6.3.21)} \\
&= 2\gamma - q - q \\
&\geq 2\gamma - p - q.
\end{aligned}
$$

2. *There exists an A_j that contains exactly one item in $\mathcal{H}$.* In this case there must exist some other A_k with at most two items from $\mathcal{H}$ as $|\mathcal{H}| = 3(n - 1)$. As previously observed, the single item in $A_j \cap \mathcal{H}$ must be of value $\leq p$ and the

two items in $A_k \cap \mathcal{H}$ must be of value $\leq 2(\gamma - q)$. Thus we have:

$$
\begin{aligned}
r &\geq \mathtt{val}_i\left(A_j\right) + \mathtt{val}_i\left(A_k\right) - p - 2(\gamma - q) \\
&\geq 2 - p - 2(\gamma - q) \text{ (since } i\text{'s MMS value is 1)} \\
&= (2\gamma - p - q) + (-4\gamma + 3q + 2) \\
&\geq (2\gamma - p - q) + (-4(3/4) + 3(1/3) + 2) \text{ (since } q \geq 1/3 \text{ by Lemma 6.3.24)} \\
&= 2\gamma - p - q.
\end{aligned}
$$

3. *$n \geq 6$ and all the A_j contain at least two items in $\mathcal{H}$.*
In this case there must be at least three A_j with exactly two items from $\mathcal{H}$ as $|\mathcal{H}| = 3(n-1)$. Without loss of generality, suppose this is true of A_1, A_2, and A_3. In each of these three we must have that the two items from $\mathcal{H}$ have value $\leq 2(\gamma - q)$ as previously mentioned. Thus we have:

$$
\begin{aligned}
r &\geq \mathtt{val}_i\left(A_1\right) + \mathtt{val}_i\left(A_2\right) + \mathtt{val}_i\left(A_3\right) - 3(2(\gamma - q)) \\
&\geq 3 - 3(2(\gamma - q)) \text{ (since } i\text{'s MMS value is 1)} \\
&= (2\gamma - p - q) + (-8\gamma + p + 7q + 3) \\
&\geq (2\gamma - p - q) + (-8\gamma + 8q + 3) \\
&\geq (2\gamma - p - q) + \left(-8\gamma + 8\frac{n}{n-1}(1 - \gamma) + 3\right) \text{ (by Lemma 6.3.21)} \\
&= (2\gamma - p - q) \\
&\qquad + \left(-8\frac{2(n-1)}{3(n-1)-1} + 8\frac{n}{n-1}\left(1 - \left(\frac{2(n-1)}{3(n-1)-1}\right)\right) + 3\right) \text{ (by Corollary 6.3.23)} \\
&= (2\gamma - p - q) + \frac{n^2 - 5n - 4}{(n-1)(3n-4)} \\
&> (2\gamma - p - q) + \frac{n^2 - 5n - n}{(n-1)(3n-4)} \text{ (since we are assuming } n \geq 6) \\
&= (2\gamma - p - q) + \frac{n(n-6)}{(n-1)(3n-4)} \\
&\geq 2\gamma - p - q \text{ (since we are assuming } n \geq 6).
\end{aligned}
$$

4. *$n = 4$ and all the A_j contain at least two items in $\mathcal{H}$.*
In this special case, there is one A_j with exactly three items in $\mathcal{H}$, and three A_j (without loss of generality, say A_1, A_2, and A_3) with exactly two items in $\mathcal{H}$ due to $|\mathcal{H}| = 3(n-1) = 9$. Furthermore, a tedious brute force computation (which we omit) demonstrates that in this case, the six most valuable items are of value at most $2\gamma + p + q$. Thus we have:

$$
\begin{aligned}
r &\geq \mathtt{val}_i\left(A_1\right) + \mathtt{val}_i\left(A_2\right) + \mathtt{val}_i\left(A_3\right) - (2\gamma + p + q) \\
&\geq 3 - (2\gamma + p + q) \text{ (since } i\text{'s MMS value is 1)} \\
&= 4(3/4) - (2\gamma + p + q) \\
&= 4\gamma - (2\gamma + p + q) \text{ (by the definition of } \gamma) \\
&= 2\gamma - p - q.
\end{aligned}
$$

As the four cases above encompass all possible scenarios, we do indeed find that $r \geq 2\gamma - p - q$. We therefore find:

$$\begin{aligned}\sum_{j \notin X \cup Y} V_j &= r + p + q + \Delta \\ &\geq (2\gamma - p - q) + p + q + \Delta \\ &= 2\gamma + \Delta.\end{aligned}$$

However, we know regarding the two $j \notin X \cup Y$ that one of the V_j must go to i (i.e. $j = \hat{i}$) and is therefore of value $< \gamma$, while the other must have value $< \gamma + \Delta$. We thus simultaneously find that:

$$\sum_{j \notin X \cup Y} V_j < \gamma + \gamma + \Delta = 2\gamma + \Delta.$$

This is a clear contradiction. ■

Lemma 6.3.32. *It holds that $x + y \neq n - 1$.*

Proof. Assume for purposes of contradiction that $x + y = n - 1$. In this case $\mathcal{N} \setminus \{X \cup Y\} = \{i\}$. Similarly to Lemma 6.3.31's proof, we introduce a set of interest under the name of $\mathcal{H}$. This is identical to before except it does not include the one item whose value was denoted as Δ. For convenience, we have restated the rest of set's contents here.

- *The items in all of the Ψ_j for all $j \in X$ (equivalently, the items that go to the agents in X).* There are $3x$ such items, and by Lemma 6.3.10 we know i values all of these items at a value of at most q, except for at most x of them which may have value in $(q, p]$.
- *The items in all of the Ψ_j for all $j \in Y$.* By Corollary 6.3.13 there are $2y$ such items, but we will imagine as if the y largest items (in i's view) are in fact two inseparable items — giving us instead $3y$ such items. Note that each such pair of inseparable items are of value $< \gamma$ and the other y items have value at most q by Observation 6.3.16.
- *The two items in Φ_i (which i values at p and q).*

If we again let $A_1, \ldots, A_n$ be an MMS partition for i we see that there exists some A_j that contains at most two of $\mathcal{H}$ since $|\mathcal{H}| = 3x + 3y + 2 = 3n - 1$. $\texttt{val}_i\left(A_j \setminus \mathcal{H}\right)$ must then be at least $\texttt{val}_i\left(A_j\right) - \max(2p, \gamma) \geq 1 - \max(2p, \gamma)$. We therefore find that upon algorithm completion i must receive a value of at least $1 - \max(2p, \gamma) + p + q$. If $2p \leq \gamma$ we have that:

$$\begin{aligned}&1 - \max(2p, \gamma) + p + q \\ &= 1 - \gamma + p + q \\ &\geq 1 - \gamma + q + q \text{ (since } q \leq p) \\ &\geq 1 - 3/4 + 1/3 + 1/3 \text{ (by Observation 6.3.3 and Lemmas 6.3.21 and 6.3.24)} \\ &= 11/12.\end{aligned}$$

Whereas if $2p > \gamma$ we have that:

$$
\begin{aligned}
&1 - \max(2p, \gamma) + p + q \\
&= 1 - 2p + p + q \\
&= 1 - p + q \\
&> 1 - (\gamma - q) + q \text{ (since } p + q < \gamma) \\
&= 1 + 2q - \gamma \\
&\geq 1 + 2(1/3) - 3/4 \text{ (by Observation 6.3.3 and Lemmas 6.3.21 and 6.3.24)} \\
&= 11/12.
\end{aligned}
$$

We therefore find that i must achieve a value of at least $11/12 \geq 3/4 \geq \gamma$. ■

Note that the statements of Lemmas 6.3.30, 6.3.31, and 6.3.32 imply that $x + y = n$. However, as we know that $i \notin X \cup Y$ by Observation 6.3.25, we also see that $x + y < n$. This contradiction concludes the proof that Algorithm 13 must produce a γ-approximate MMS allocation. ■

6.3.2 Proof of Theorem 6.3.1: Polynomial Time

While Algorithm 13 seems rather innocent at first glance, it does make one computational leap by letting agents compute their MMS guarantee, or an MMS partition. It is easy to see that this is **NP**-hard; in fact, even when there are two agents with identical valuations, it is **NP**-hard to determine whether the the MMS guarantee is $\mathtt{val}_i(\mathcal{M})/2$ — this can be shown via an immediate reduction from Partition.

Woeginger [99] studied the problem of computing an MMS partition, albeit under a different name: scheduling jobs on identical machines to maximize the minimum completion time. He gave a polynomial-time approximation scheme (PTAS), and showed that no fully polynomial-time approximation scheme (FPTAS) exists unless $\mathbf{P} = \mathbf{NP}$. Using our terminology, this means that given a constant $\epsilon > 0$ we can compute a partition $A_1, \ldots, A_n$ of the set of items $\mathcal{M}$ so that $\min_{i \in \mathcal{N}} \mathtt{val}_i(A_i) \geq (1 - \epsilon)\mathtt{MMS}_i(n, \mathcal{M})$ in polynomial time.

The modified algorithm is almost identical to Algorithm 13, but for two critical differences.

1. When we need to compute an agent's MMS guarantee, we instead compute a $1 - \epsilon$ approximation via the PTAS.
2. If two agents remain in Step 2, then we compute a $1 - \epsilon$ approximation to an MMS partition via the PTAS.

The analysis of Section 6.3.1 goes through largely unchanged, giving each agent a bundle of value $(1 - \epsilon)\gamma\mathtt{MMS}_i(n, \mathcal{M})$. ■

6.4 Random Valuations

We have now seen that MMS allocations do not always exist but are guaranteed a 2/3 approximation. However, as we have noted previously, constructions where a full MMS allocation does not exist is extremely rare in practice and indeed, our counterexamples to the existence in Section

6.2 are very sensitive: tiny random perturbations are extremely likely to invalidate them. Our goal in this section is to prove MMS allocations do, in fact, exist with high probability, if a small amount of randomness is present.

To this end, let us consider a probabilistic model with the following features:

1. For all $i \in \mathcal{N}$, $\mathcal{D}_i$ denotes a probability distribution over $[0, 1]$.
2. For all $i \in \mathcal{N}, g \in \mathcal{M}$, $\mathtt{val}_i(g)$ is randomly sampled from $\mathcal{D}_i$.
3. The set of random variables $\{\mathtt{val}_i(g)\}_{i \in \mathcal{N}, g \in \mathcal{M}}$ is mutually independent.

We will establish the following theorem:

Theorem 6.4.1. *Assume that for all $i \in \mathcal{N}$, $\mathbb{V}[\mathcal{D}_i] \geq c$ for a constant $c > 0$. Then for all $\epsilon > 0$ there exists $K = K(c, \epsilon)$ such that if $\max(n, m) \geq K$, then the probability that an MMS allocation exists is at least $1 - \epsilon$.*

In words, as long as each $\mathcal{D}_i$ has constant variance, if either the number of agents or the number of goods goes to infinity, there exists an MMS allocation with high probability. In parallel, independent work, Amanatidis et al. [4] establish (as one of several results) a special case of Theorem 6.4.1 where each $\mathcal{D}_i$ is the uniform distribution over $[0, 1]$. Dealing with arbitrary distributions presents significant technical challenges, and is also important in terms of explaining the abovementioned experiments, which cover a wide range of distributions. Yet the result of Amanatidis et al. is not completely subsumed by Theorem 6.4.1, as they carefully analyze the rate of convergence to 1.

Our starting point is a result by Dickerson et al. [38], who study the existence of envy-free allocations. They show that an envy-free allocation exists with high probability as $m \to \infty$, as long as $n \in O(m/\ln m)$, and the distributions $\mathcal{D}_i$ satisfy the following conditions for all $i, j \in \mathcal{N}$:

1. $\mathbb{P}\left[\arg\max_{k \in \mathcal{N}} \mathtt{val}_k(g) = \{i\}\right] = 1/n$.
2. There exist constants μ, μ^* such that

$$\begin{aligned} 0 &< \mathbb{E}\left[\mathtt{val}_i(g) \;\middle|\; \arg\max_{k \in \mathcal{N}} \mathtt{val}_k(g) = \{j\}\right] \\ &\leq \mu < \mu^* \\ &\leq \mathbb{E}\left[\mathtt{val}_i(g) \;\middle|\; \arg\max_{k \in \mathcal{N}} \mathtt{val}_k(g) = \{i\}\right]. \end{aligned}$$

The proof uses a naïve allocation algorithm: simply give each good to the agent who values it most highly. The first condition then implies that each agent receives roughly $1/n$ of the goods, and the second condition ensures that each agent has higher expected value for each of his own goods compared to goods allocated to other agents.

It turns out that, via only slight modifications, their theorem can largely work in our setting. That is, alter their allocation algorithm to give a good g to an agent i who believes g is in the top $1/n$ of their probability distribution $\mathcal{D}_i$. If there are multiple such agents, choose one uniformly at random and if no such agent exists, give it to any agent uniformly at random.

This procedure is fairly straightforward for continuous probability distributions. For example, if agent i's distribution $\mathcal{D}_i$ is uniform over the interval $[0, 1]$ then he believes g is in the top $1/n$ of $\mathcal{D}_i$ if $\mathtt{val}_i(g) \geq (n-1)/n$. However, distributions with atoms require more care. For example, suppose D_i is $1/3$ with probability $7/8$ and uniform over $[1/2, 1]$ with probability $1/8$.

Then if $n = 3$, i believes g is in the top $1/n$ of $\mathcal{D}_i$ if $\texttt{val}_i(g) > 1/3$ or if $\texttt{val}_i(g) = 1/3$ he should believe it is in his top $1/n$ only $1/n - 1/8 = 5/24$ of the time. To implement such a procedure, when sampling from $\mathcal{D}_i$, we should first sample from the uniform distribution over $[0, 1]$. If our sampled value is at least $(n-1)/n$ we will say i has drawn from his top $1/n$. We then convert our sampled value to a sampled value from $\mathcal{D}_i$ by applying the inverse CDF.

Utilizing the observation that any envy-free allocation is also an MMS allocation we can then restate the result of Dickerson et al. [38] as the following lemma, whose proof we will see in Section 6.4.1.

Lemma 6.4.2 ([38]). *Assume that for all $i \in \mathcal{N}$, $\mathbb{V}[\mathcal{D}_i] \geq c$ for a constant $c > 0$. Then for all $\epsilon > 0$ there exists $K = K(\epsilon)$ such that if $m \geq K$ and $m \geq \alpha n \ln n$, for some $\alpha = \alpha(c)$, then the probability that an MMS allocation exists is at least $1 - \epsilon$.*

Note that the statement of Lemma 6.4.2 is identical to that of Theorem 6.4.1, except for two small changes: only m is assumed to go to infinity, and the additional condition $m \geq \alpha n \ln n$. So it only remains to deal with the case of $m < \alpha n \ln n$. We can handle this scenario via consideration of the case $m < n^{8/7}$ — formalized in the following lemma.

Lemma 6.4.3. *For all $\epsilon > 0$ there exists $K = K(\epsilon)$ such that if $n \geq K$ and $m < n^{8/7}$, then the probability that an MMS allocation exists is at least $1 - \epsilon$.*

Note that this lemma actually does not even require the minimum variance assumption, that is, we are proving a stronger statement than is needed for Theorem 6.4.1.

It is immediately apparent that when the number of goods is relatively small, we will not be able to prove the existence of MMS allocations via the existence of envy-free allocations. For example, envy-free allocations certainly do not exist if $m < n$, and are provably highly unlikely to exist if $m = n + o(n)$ [38]. Our approach to this lemma (which we give in Section 6.4.2) is therefore significantly more intricate.

6.4.1 Proof of Lemma 6.4.2

The crux of the proof in [38] relies on the allocation algorithm only satisfying the following two properties.

1. For any good g, if we do not condition on the $\texttt{val}_i(g)$, then every agent has a $1/n$ probability of receiving g.
2. For some constant Δ, we have that

$$\mathbb{E}\left[\texttt{val}_i(g) \mid i \text{ receives } g\right] - \mathbb{E}\left[\texttt{val}_i(g) \mid i \text{ does not receive } g\right] \geq \Delta.$$

We must show that our allocation algorithm implies these two properties in our setting. The first is clear via symmetry and so we turn our attention to the second. We claim that $\Delta = c/16$ suffices (recall that $\mathbb{V}[\mathcal{D}_i] \geq c$).

Let $X \sim \mathcal{D}_i$, $\bar{X} = \mathbb{E}[X]$, $p = \mathbb{P}\left[X < \bar{X}\right]$, and γ represent the value such that $\mathbb{P}[X \geq \gamma] = 1/n$[4]. We first show that $\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma] \geq c/2$.

$$\begin{aligned}
c &\leq \mathbb{V}[X] \\
&= \mathbb{E}\left[(X - \bar{X})^2\right] \\
&\leq \mathbb{E}\left[|X - \bar{X}|\right] \\
&= p\mathbb{E}\left[\bar{X} - X \mid X < \bar{X}\right] + (1-p)\mathbb{E}\left[X - \bar{X} \mid X \geq \bar{X}\right] \\
&= -p\mathbb{E}\left[X \mid X < \bar{X}\right] + (1-p)\mathbb{E}\left[X \mid X \geq \bar{X}\right] + (2p-1)\bar{X}.
\end{aligned}$$

Knowing that $\bar{X} = p\mathbb{E}\left[X \mid X < \bar{X}\right] + (1-p)\mathbb{E}\left[X \mid X \geq \bar{X}\right]$ there are two cases.

1. $\gamma < \bar{X}$.

$$\begin{aligned}
c &\leq -p\mathbb{E}\left[X \mid X < \bar{X}\right] + (1-p)\mathbb{E}\left[X \mid X \geq \bar{X}\right] + (2p-1)\bar{X} \\
&= -2p\mathbb{E}\left[X \mid X < \bar{X}\right] + \bar{X} + (2p-1)\bar{X} \\
&= 2p\left(\bar{X} - \mathbb{E}\left[X \mid X < \bar{X}\right]\right) \\
&\leq 2p\left(\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma]\right) \\
&\leq 2\left(\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma]\right).
\end{aligned}$$

2. $\gamma \geq \bar{X}$.

$$\begin{aligned}
c &\leq -p\mathbb{E}\left[X \mid X < \bar{X}\right] + (1-p)\mathbb{E}\left[X \mid X \geq \bar{X}\right] + (2p-1)\bar{X} \\
&= -\bar{X} + 2(1-p)\mathbb{E}\left[X \mid X \geq \bar{X}\right] + (2p-1)\bar{X} \\
&= 2(1-p)\left(\mathbb{E}\left[X \mid X \geq \bar{X}\right] - \bar{X}\right) \\
&\leq 2(1-p)\left(\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma]\right) \\
&\leq 2\left(\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma]\right).
\end{aligned}$$

In either case, we see that $\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma] \geq c/2$ as desired. Now observe that

$$\mathbb{E}[\mathtt{val}_i(g) \mid i \text{ receives } g] = (1 - 1/n)^n\, \mathbb{E}[X \mid X < \gamma] + (1 - (1-1/n)^n)\, \mathbb{E}[X \mid X \geq \gamma],$$

and

$$\begin{aligned}
\mathbb{E}[\mathtt{val}_i(g) \mid i \text{ does not receive } g] &\leq \mathbb{E}[\mathtt{val}_i(g)] \\
&= (1 - 1/n)\, \mathbb{E}[X \mid X < \gamma] + (1/n)\mathbb{E}[X \mid X \geq \gamma].
\end{aligned}$$

Thus, we have that

$$\begin{aligned}
&\mathbb{E}[\mathtt{val}_i(g) \mid i \text{ receives } g] - \mathbb{E}[\mathtt{val}_i(g) \mid i \text{ does not receive } g] \\
&\geq \left((1-1/n) - (1-1/n)^n\right)\left(\mathbb{E}[X \mid X \geq \gamma] - \mathbb{E}[X \mid X < \gamma]\right) \\
&\geq (1/2 - 1/e)\,(c/2) \\
&\geq c/16.
\end{aligned}$$

[4]As discussed previously, such a γ may not exist in distributions with atoms, but we ignore this possibility purely for ease of exposition.

■

6.4.2 Proof of Lemma 6.4.3

We assume that $m > n$, because an MMS allocation always exists when $m \leq n$ (in fact, when $m \leq n+4$, as Theorem 6.5.1 will show). We will require the following notions and lemma.

Definition 6.4.4. *A* ranking *of the goods $\mathcal{M}$ for some agent $i \in \mathcal{N}$ is the order of the goods by value from most valued to least. Ties are broken uniformly at random. Furthermore, a good g's* rank *for an agent i is the position of g in i's ranking.*

An important observation of the rankings that we will use often throughout this section is that the agents' rankings are independent of each other.

Definition 6.4.5. *Suppose $X \subseteq \mathcal{N}$ and $Y \subseteq \mathcal{M}$ where $|X| \leq |Y|$. Let*

$$s = s(X,Y) = |X| \lceil |Y|/|X| \rceil - |Y|,$$

and $\Gamma = \Gamma(X,Y)$ be the bipartite graph where:

1. *L represents the vertices on the left, and R on the right.*
2. *L is comprised of $\lfloor |Y|/|X| \rfloor$ copies of the first s agents of X and $\lceil |Y|/|X| \rceil$ copies of the remaining agents in X.*
3. *$R = Y$.*
4. *The i^{th} copy of an agent has an edge to a good g iff g's rank is in $((i-1)\Delta, i\Delta]$ in the agent's ranking where $\Delta = \ln^3 n$.*

Note that in this definition $|L| = |R|$ since if we let $x = |X|$ and $y = |Y|$ (and therefore $s = x\lceil y/x \rceil - y$). Then

$$\begin{aligned}|L| &= s\lfloor y/x \rfloor + (x-s)\lceil y/x \rceil \\ &= x\lceil y/x \rceil - s\left(\lceil y/x \rceil - \lfloor y/x \rfloor\right).\end{aligned}$$

If x divides y, then we have that $\lceil y/x \rceil = \lfloor y/x \rfloor = \frac{y}{x}$ and so $|L| = y$. If, on the other hand, x does not divide y, then we have that $\lceil y/x \rceil - \lfloor y/x \rfloor = 1$ and so we have

$$\begin{aligned}|L| &= x\lceil y/x \rceil - s \\ &= x\lceil y/x \rceil - (x\lceil y/x \rceil - y) \\ &= y.\end{aligned}$$

Therefore, in either case, $|L| = y = |Y| = |R|$.

Definition 6.4.6. *Suppose $X \subseteq \mathcal{N}$ and $Y \subseteq \mathcal{M}$ as before. Then the* matched draft *on X and Y is the process of constructing Γ and producing an allocation corresponding to a perfect matching of Γ. That is, if a perfect matching exists then an agent in X is given all goods the copies of it are matched to. In the event that no perfect matching exists, the matched draft is said to fail.*

Lemma 6.4.7. *Suppose of the $m < n^{8/7}$ goods $x = \gamma\lfloor m/n \rfloor$ are randomly chosen and removed, where $\gamma \leq n^{1/3}$, and the remaining $\tilde{m} := m - x$ goods are allocated via a matched draft to $\tilde{n} := n - \gamma$ agents. Then this matched draft succeeds with probability $\to 1$ as $n \to \infty$ (note that as $n \to \infty$, so too do $\tilde{n}, \tilde{m}$).*

Proof. Define d as the minimum degree of a vertex of L in Γ and $D = 2 \lg n \ln n$. Then we have

$$
\begin{aligned}
&\mathbb{P}\,[\text{matched draft fails}] \\
&= \mathbb{P}\,[\text{matched draft fails} \mid d < D]\,\mathbb{P}\,[d < D] \\
&\quad + \mathbb{P}\,[\text{matched draft fails} \mid d \geq D]\,\mathbb{P}\,[d \geq D] \\
&\leq \mathbb{P}\,[d < D] + \mathbb{P}\,[\text{matched draft fails} \mid d \geq D]\,.
\end{aligned}
$$

Let us consider these two terms separately and show they $\to 0$ as $n \to \infty$.

Denoting by p_D^{ij} the probability that the i^{th} of the $\tilde{n}$ remaining agents has less than D of the $\tilde{m}$ remaining goods ranked in positions $((j-1)\Delta, j\Delta]$. Then we have

$$
\begin{aligned}
\mathbb{P}\,[d < D] &\leq \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{\lceil \tilde{m}/\tilde{n} \rceil} p_D^{ij} \\
&= \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{\lceil \tilde{m}/\tilde{n} \rceil} p_D^{11} \text{ (by independence)} \\
&= \tilde{n} \,\lceil \tilde{m}/\tilde{n} \rceil\, p_D^{11} \\
&\leq 2\tilde{m} p_D^{11}.
\end{aligned}
$$

Now let the random variable X denote the number of the x randomly chosen goods ranked in the top Δ for first remaining agent. Then we have

$$
\mathbb{P}\,[d < D] \leq 2\tilde{m} p_D^{11} = 2\tilde{m}\mathbb{P}\,[\Delta - D < X \leq \Delta]\,.
$$

For sufficiently large n and $i \in (\Delta - D, \Delta]$ we further note that $\mathbb{P}\,[X = i] \leq \mathbb{P}\,[X = \Delta/2]$ and so

$$
\begin{aligned}
\mathbb{P}\,[d < D] &\leq 2\tilde{m}\mathbb{P}\,[\Delta - D < X \leq \Delta] \\
&= 2\tilde{m} \sum_{i=\Delta-D+1}^{\Delta} \mathbb{P}\,[X = i] \\
&\leq 2\tilde{m} \sum_{i=\Delta-D+1}^{\Delta} \mathbb{P}\,[X = \Delta/2] \\
&\leq 2\tilde{m} D \mathbb{P}\,[X = \Delta/2] \\
&= 2\tilde{m} D \binom{x}{\Delta/2} \left(\frac{\Delta}{m}\right)^{\Delta/2} \left(1 - \frac{\Delta}{m}\right)^{x-\Delta/2} \\
&\leq 2\tilde{m} D \binom{x}{\Delta/2} \left(\frac{\Delta}{m}\right)^{\Delta/2} \\
&\leq 2\tilde{m} D \left(\frac{ex}{\Delta/2}\right)^{\Delta/2} \left(\frac{\Delta}{m}\right)^{\Delta/2} \text{ (by the standard inequality } \binom{p}{q} \leq \left(\frac{ep}{q}\right)^{q}\text{)} \\
&= 2\tilde{m} D \left(\frac{2ex}{m}\right)^{\Delta/2}.
\end{aligned}
$$

Now substituting in that $\Delta = \ln^3 n$, $\tilde{m} < m < n^{8/7}$, $D = 2\lg n \ln n$, and $x/m \leq \gamma \lfloor m/n \rfloor /m \leq \gamma/n \leq n^{-2/3}$ we then find that the last quantity $\to 0$ as $n \to \infty$.

A quick note for rigor: We have used the probability mass function of the binomial distribution here when we technically require the use of the hypergeometric. However, as the probability of a collision occurring is asymptotically low itself it is insignificant and as the inclusion of this analysis would only greatly convolute the proof, we omit it.

Next let us consider $\mathbb{P}[\text{matched draft fails} \mid d \geq D]$. We would like to appeal to the plethora of results on perfect matchings in bipartite Erdös-Rényi graphs [15] or random bipartite k-out graphs [70], but due to the lack of independence on the edge existences we do not satisfy a crucial assumption of much of this literature, and more importantly its proofs. We will therefore prove this in full here via an approach that allows us to ignore the dependence. We will utilize Hall's theorem and denote by $N(X)$ the set of neighbors of X in the bipartite graph Γ.

$$
\begin{aligned}
&\mathbb{P}[\text{matched draft fails} \mid d \geq D] \\
&= \mathbb{P}[\exists\, X \subseteq L \text{ s.t. } |X| < |N(X)| \mid d \geq D] \\
&\leq \sum_{X \subseteq L} \mathbb{P}[|X| < |N(X)| \mid d \geq D] \\
&\leq \sum_{i=D}^{\tilde{m}} \sum_{\substack{X \subseteq L \\ |X|=i}} \sum_{\substack{Y \subseteq R \\ |Y|=i-1}} \mathbb{P}[N(X) \subseteq Y \mid d \geq D].
\end{aligned}
$$

If the edges of Γ were independent then we would find that for $|X| = i$ and $|Y| = i - 1$,

$$
\mathbb{P}[N(X) \subseteq Y] = \left(\frac{i-1}{\tilde{m}}\right)^{\sum_{x \in X} |N(x)|},
$$

and more importantly

$$
\mathbb{P}[N(X) \subseteq Y \mid d \geq D] \leq \left(\frac{i-1}{\tilde{m}}\right)^{iD}. \tag{6.3}
$$

Via our independence assumptions in our randomized setting there is only one form of dependence in the edges of Γ. Specifically, if we take all copies of any agent $i \in L$, then their neighbors in R never intersect. Though this does indeed introduce dependence into our system, note that we still have that Equation (6.3) as the dependence only lowers the probability of $N(X)$ "fitting" into Y. We therefore find

$$
\begin{aligned}
&\mathbb{P}\,[\text{matched draft fails} \mid d \geq D] \\
&\leq \sum_{i=D}^{\tilde{m}} \sum_{\{X \subseteq L \mid |X|=i\}} \sum_{\{Y \subseteq R \mid |Y|=i-1\}} \left(\frac{i-1}{\tilde{m}}\right)^{iD} \\
&= \sum_{i=D}^{\tilde{m}} \binom{\tilde{m}}{i}\binom{\tilde{m}}{i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} \\
&\leq \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \binom{\tilde{m}}{i}\binom{\tilde{m}}{i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} + \sum_{i=\lceil \tilde{m}/2 \rceil}^{\tilde{m}} \binom{\tilde{m}}{\tilde{m}-i}\binom{\tilde{m}}{\tilde{m}-i+1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} \\
&= \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \binom{\tilde{m}}{i}\binom{\tilde{m}}{i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} + \sum_{j=0}^{\lfloor \tilde{m}/2 \rfloor} \binom{\tilde{m}}{j}\binom{\tilde{m}}{j+1}\left(\frac{\tilde{m}-j-1}{n}\right)^{(\tilde{m}-j)D}.
\end{aligned}
$$

We now show both of these terms separately $\to 0$ as $n \to \infty$.

First,

$$
\begin{aligned}
\sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \binom{\tilde{m}}{i}\binom{\tilde{m}}{i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} &\leq \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{i}\right)^{i}\left(\frac{\tilde{m}e}{i-1}\right)^{i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} \\
&\leq \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{i-1}\right)^{2i-1}\left(\frac{i-1}{\tilde{m}}\right)^{iD} \\
&= \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{i-1}{\tilde{m}}\right)^{i(D-2)+1} e^{2i-1} \\
&= \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\lfloor \tilde{m}/2 \rfloor - 1}{\tilde{m}}\right)^{i(D-2)+1} e^{2i-1} \\
&\leq \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \frac{e^{2i-1}}{2^{i(D-2)+1}} \\
&= \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \frac{1}{2e}\left(\frac{e^2}{2^{D-2}}\right)^{i} \\
&\leq \sum_{i=D}^{\lfloor \tilde{m}/2 \rfloor} \frac{1}{2e}\frac{e^2}{2^{D-2}} \\
&= (\lfloor \tilde{m}/2 \rfloor - D + 1)\left(\frac{e}{2^{D-1}}\right) \\
&\leq m\left(\frac{e}{2^{D-1}}\right).
\end{aligned}
$$

Substituting in that $\tilde{m} \leq m < n^{8/7}$ and $D = 2 \lg n \ln n$ we then find that the last quantity $\to 0$ as $n \to \infty$.

Second,

$$
\begin{aligned}
&\sum_{j=0}^{\lfloor \tilde{m}/2 \rfloor} \binom{\tilde{m}}{j}\binom{\tilde{m}}{j+1}\left(\frac{\tilde{m}-j-1}{\tilde{m}}\right)^{(\tilde{m}-j)D} \\
&\le \tilde{m}\left(\frac{\tilde{m}-1}{\tilde{m}}\right)^{\tilde{m}D} + \sum_{j=1}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{j}\right)^{j}\left(\frac{\tilde{m}e}{j+1}\right)^{j+1}\left(\frac{\tilde{m}-j-1}{\tilde{m}}\right)^{(\tilde{m}-j)D} \\
&\le \tilde{m}\left(1-\frac{1}{\tilde{m}}\right)^{\tilde{m}D} + \sum_{j=1}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{j}\right)^{2j+1}\left(1-\frac{j+1}{\tilde{m}}\right)^{(\tilde{m}-j)D} \\
&\le \tilde{m}e^{-D} + \sum_{j=1}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{j}\right)^{2j+1} e^{-D(j+1)(\tilde{m}-j)/\tilde{m}} \\
&\le \tilde{m}e^{-D} + \sum_{j=1}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}e}{j}\right)^{2j+1} e^{-D(j+1)/2} \\
&\le \tilde{m}e^{-D} + \sum_{j=1}^{\lfloor \tilde{m}/2 \rfloor} \left(\frac{\tilde{m}^2e^2}{j^2e^{D/2}}\right)^{j+1} \\
&\le \frac{n^{8/7}}{n^{2\lg n}} + \sum_{j=1}^{\lfloor n^{8/7}/2 \rfloor} \left(\frac{(n^{8/7})^2e^2}{j^2n^{\lg n}}\right)^{j+1} \\
&\le \frac{n^{8/7}}{n^{2\lg n}} + \left\lfloor n^{8/7}/2 \right\rfloor \left(\frac{(n^{8/7})^2e^2}{n^{\lg n}}\right) \\
&\to 0,
\end{aligned}
$$

where the third inequality follows from $1 + x \le e^x$ for all x.

Thus, we find that as $n \to \infty$ the matched draft succeeds with probability $\to 1$. ■

We are now ready to prove the main lemma.

Proof of Lemma 6.4.3. Recall that we may assume that $m > n$. We will ensure every agent has at most one less good than any other agent. Let s then represent the number of agents that receive one less good than any other agent, that is,

$$s = n\lceil m/n \rceil - m.$$

We consider two separate cases here.

Case 1: $s \le n^{1/3}$. In this scenario we do the following.

1. If possible, give each of the first s agents their top $\lfloor m/n \rfloor$ goods. Otherwise, fail to produce any allocation.
2. Hold a matched draft for the remaining $(n-s)\lceil m/n \rceil$ goods and $n-s$ agents.

We first show that as $n \to \infty$ this procedure successfully produces an allocation with probability $\to 1$.

Consider the probability that the first step of the procedure successfully completes. That is, the first s agents each get their top $\lfloor m/n \rfloor$ goods. Similarly to a birthday paradox like argument we get that this occurs with probability at least

$$\prod_{i=1}^{s\lfloor m/n \rfloor} \left(1 - \frac{i-1}{m}\right) > \left(1 - \frac{sm/n}{m}\right)^{sm/n} \geq \left(1 - \frac{1}{n^{2/3}}\right)^{n^{1/3+8/7-1}} = \left(1 - \frac{1}{n^{2/3}}\right)^{n^{10/21}}.$$

But as

$$\lim_{x\to\infty} \left(1 - \frac{1}{\omega(x)}\right)^x = 1$$

we find that this too goes to 1 as $n \to \infty$.

Now consider the second step of the procedure. By Lemma 6.4.7 with $\gamma = s$, we know that this succeeds with probability 1 as $n \to \infty$. Therefore the entire procedure will successfully complete with the same asymptotic probability guarantee.

To prove the theorem then, it suffices to show that if the procedure successfully completes, then we have an MMS allocation. Since for every agent any MMS partition must include a subset with at most $\lfloor m/n \rfloor$ goods and the first s agents are given their top $\lfloor m/n \rfloor$ goods, they must receive their MMS value.

Let us turn our attention then to the remaining $n - s$ agents. Upon successful completion of the matched draft, we know that all of these agents will receive goods ranked in their top $\Delta \lceil m/n \rceil$. We claim that for sufficiently large n any agent's MMS partition must include a subset of at most $\lceil m/n \rceil$ goods where each good is ranked lower than $\Delta \lceil m/n \rceil$. Suppose this were not true for purposes of contradiction. Then each of the n subsets in an offending agent's MMS partition must include either one of the top $\Delta \lceil m/n \rceil$ goods or $\lceil m/n \rceil + 1$ goods. We then see that for sufficiently large n, the number of such subsets is bounded by

$$\begin{aligned}
&\Delta \lceil m/n \rceil + \frac{m - \Delta \lceil m/n \rceil}{\lceil m/n \rceil + 1} \\
&= \Delta \lceil m/n \rceil + \frac{s(\lceil m/n \rceil - 1) + (n-s)\lceil m/n \rceil - \Delta \lceil m/n \rceil}{\lceil m/n \rceil + 1} \\
&= \frac{\Delta \lceil m/n \rceil^2 + n\lceil m/n \rceil - s}{\lceil m/n \rceil + 1} \\
&\leq \frac{\lceil m/n \rceil}{\lceil m/n \rceil + 1} n + \Delta \lceil m/n \rceil \\
&\leq \frac{n^{1/7}}{n^{1/7} + 1} n + \left(\ln^3 n\right)\left(n^{1/7}\right) \\
&< n.
\end{aligned}$$

Thus the offending agent cannot produce such an MMS partition which proves the claim.

Now note that the $n - s$ agents of interest have MMS partitions with bundles that include the same number of goods they received, but all of which are worth strictly less than every good in their bundle. They therefore must have achieved their MMS value.

Case 2: $s > n^{1/3}$. In this scenario we simply run a matched draft. Similarly to the previous case we know from Lemma 6.4.7 with $\gamma = 0$ that all the agents will receive goods ranked in their top $\Delta \lceil m/n \rceil$ with probability $\to 1$ as $n \to \infty$.

In this case for sufficiently large n any agent's MMS partition must include a subset of at most $\lfloor m/n \rfloor$ goods where each good is ranked lower than $\Delta \lceil m/n \rceil$. Again, suppose this were not true for purposes of contradiction. Then each of the n subsets in an agent's MMS partition must include either one of the top $\Delta \lceil m/n \rceil$ goods or $\lfloor m/n \rfloor + 1 = \lceil m/n \rceil$ goods (in this case $m \not\equiv 0 \pmod{n}$). We then see that for sufficiently large n, the number of subsets is at most

$$\begin{aligned}
&\Delta \lceil m/n \rceil + \frac{m - \Delta \lceil m/n \rceil}{\lceil m/n \rceil} \\
&= \Delta \lfloor m/n \rfloor + \frac{s(\lceil m/n \rceil - 1) + (n-s) \lceil m/n \rceil}{\lceil m/n \rceil} \\
&= n + \Delta \lfloor m/n \rfloor - \frac{s}{\lceil m/n \rceil} \\
&\leq n + \left(\ln^3 n\right)\left(n^{1/7}\right) - \frac{n^{1/3}}{n^{1/7}} \\
&< n.
\end{aligned}$$

Via logic similar to the previous case, we conclude that all agents must have achieved their MMS value. ■

6.5 A Small Number of Goods Guarantees the Existence of an MMS Allocation

In Section 6.2 we saw that even with $m = 3n+4$ goods, we cannot guarantee an MMS allocation. In this section, we consider the dual problem: For what values of m are we assured there are always such desired allocations?

As we have previously mentioned, when $m < n$, the MMS guarantee of each agent is 0, so any allocation is an MMS allocation. If $m = n$, we have that $\mathtt{MMS}_i(n, \mathcal{M}) = \min_{g \in \mathcal{M}} \mathtt{val}_i(g)$. So any allocation that gives a single good to each agent fits the bill. The case of $m = n+1$ is still trivial: the MMS partition of an agent puts his two least desirable goods in one bundle, and every other good in a singleton bundle. Therefore, it is sufficient to let the agents choose a single good each in the order $1, \ldots, n-1$, and allocate to agent n the two remaining goods. Bouveret and Lemaître [16] extend this argument to show that an MMS allocation exists whenever $m \leq n+3$.

In this section, which we view as a bit of an aside, we slightly improve the bound of Bouveret and Lemaître [16] to $m \leq n+4$. While the improvement is not of major excitement, we include it as we believe the approach is quite interesting and its ideas may be used to further hone the bounds.

Theorem 6.5.1. *If $m \leq n+4$ then there exists an MMS allocation.*

Proof. We give a detailed algorithm (with some commentary) to handle the case of $m \leq n+4$ as Algorithm 14, but we highlight some of the nuances here.

while $\exists i \in \mathcal{N}$ s.t. $\exists g \in \mathcal{M}$ where $\texttt{val}_i\,(g) \geq \texttt{MMS}_i(|\mathcal{N}|, \mathcal{M})$ **do**
  Give g to agent i.
  $\mathcal{N} \leftarrow \mathcal{N} \setminus \{i\}$
  $\mathcal{M} \leftarrow \mathcal{M} \setminus \{g\}$
end while
// Note that at this point $|\mathcal{N}| \leq 4$.
For convenience, relabel the agents to $1, 2, \ldots$
if $|\mathcal{N}| = 1$ **then**
  Give all of $\mathcal{M}$ to agent 1.
else if $|\mathcal{N}| = 2$ **then**
  Use an MMS partition for these two agents on the remaining goods $\mathcal{M}$.
else if $|\mathcal{N}| = 3$ **then**
  // $|\mathcal{M}| \in \{6, 7\}$, so every MMS partition of any agent has two subsets of size 2.
  Let X_1, X_2, X_3 be an MMS partition of agent 1 where $|X_1| = |X_2| = 2$.
  if $\exists i \in \{2, 3\}$ s.t. $\texttt{val}_i\,(X_1) < \texttt{MMS}_i(|\mathcal{N}|, \mathcal{M})$ or $\texttt{val}_i\,(X_2) < \texttt{MMS}_i(|\mathcal{N}|, \mathcal{M})$ **then**
    WLOG assume $\texttt{val}_2\,(X_1) < \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$.
    Let Y_1, Y_2, Y_3 be an MMS partition of agent 2.
    // Note that X_1 is not contained in any single one of Y_1, Y_2, Y_3.
    WLOG assume $X_1 \subseteq Y_1 \cup Y_2$.
    $Z := (Y_1 \cup Y_2) \setminus X_1$.
    Let agent 3 take one of X_1, Y_3, Z.
    if agent 3 chooses X_1 **then**
      // As $\texttt{val}_2\,(X_1) < \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$ one of $\texttt{val}_2\,(X_2)\,, \texttt{val}_2\,(X_3) \geq \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$.
      Give agent 2 one of X_2 and X_3 such that he achieves his MMS.
      Give agent 1 the other subset.
    else if agent 3 chooses Y_3 **then**
      Give X_1 to agent 1.
      // Note that $\texttt{val}_2\,(Y_1)\,, \texttt{val}_2\,(Y_2) \geq \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$ and $\texttt{val}_2\,(X_1) < \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$.
      // Thus, $\texttt{val}_2\,(Z) = \texttt{val}_2\,(Y_1) + \texttt{val}_2\,(Y_2) - \texttt{val}_2\,(X_1) > \texttt{MMS}_2(|\mathcal{N}|, \mathcal{M})$.
      Give Z to agent 2.
    else if Agent 3 chooses Z **then**
      Give X_1 to agent 1.
      Give Y_3 to agent 2.
    end if
  else
    Give X_3 to agent 1.
    Give X_1 to agent 2.
    Give X_2 to agent 3.
  end if
else if $|\mathcal{N}| = 4$ **then**
  // $|\mathcal{M}| = 8$, so every MMS partition of any agent has only subsets of size 2.
  Let the agents choose a single good one at a time in the order: $1, 2, 3, 4, 4, 3, 2, 1$.
end if

Algorithm 14: MMS Allocation for $m \leq n + 4$.

Observe that whenever an agent believes a single good is worth at least his MMS value we can give that good to him and in the reduced problem (where there is one less agent and one less good) every agent's MMS value has not decreased. Thus, so long as the reduced problem has an MMS allocation we will have an MMS allocation overall.

Now note that as long as $m < 2n$ every MMS partition of any agent must include a subset with a single good. We can therefore utilize the observation repeatedly until there are at most 4 agents left. In the event that there are at most 2 agents left, we know this is easily handled and so only the cases where there are 3 or 4 agents remaining are of interest.

In the more complex outcome where there are 3 agents we essentially have an intricate case analysis that is best understood via the fully explicit treatment given in Algorithm 14. We therefore only consider the case where there are 4 agents left here. In such an outcome exactly 8 goods remain and in every MMS partition of any agent all subsets of the partition are of size 2 (as otherwise some agent achieves his MMS value with a single good).

We claim then that for any agent i if the goods $g_1, \ldots, g_8$ were sorted by value in that order (i.e. $\mathtt{val}_i\left(g_j\right) \geq \mathtt{val}_i\left(g_k\right)$ for all $j \leq k$) then $\{g_1, g_8\}, \{g_2, g_7\}, \{g_3, g_6\}, \{g_4, g_5\}$ is an MMS partition for i (i.e. the partition where g_j paired with g_{9-j}). Suppose this were not true, then let $j \in \arg\min_{k \leq 4}(\mathtt{val}_i\left(g_k\right) + \mathtt{val}_i\left(g_{9-k}\right))$. Now let S_1, S_2, S_3, S_4 be any MMS partition for i and consider the good g_j is paired with, say g_k. We know that that $k < 9 - j$ as otherwise we would have

$$\begin{aligned}
\mathtt{MMS}_i(|\mathcal{N}|, \mathcal{M}) &= \min_k \mathtt{val}_i\left(S_k\right) \\
&\leq \mathtt{val}_i\left(g_j\right) + \mathtt{val}_i\left(g_k\right) \\
&\leq \mathtt{val}_i\left(g_j\right) + \mathtt{val}_i\left(g_{9-j}\right) \\
&= \min_{k \leq 4}(\mathtt{val}_i\left(g_k\right) + \mathtt{val}_i\left(g_{9-k}\right)).
\end{aligned}$$

Now consider the j goods $g_{9-j}, g_{10-j}, \ldots, g_8$. As we require $\mathtt{MMS}_i(|\mathcal{N}|, \mathcal{M}) = \min_k \mathtt{val}_i\left(S_k\right) > \mathtt{val}_i\left(g_j\right) + \mathtt{val}_i\left(g_{9-j}\right)$ we must have that the goods they are paired with in the MMS partition chosen have value greater than $\mathtt{val}_i\left(g_j\right)$. Unfortunately, there are at most $j - 1$ such goods — a clear contradiction.

Thus, if we allow agents to choose one good at a time, we find that so long as an agent gets to make the j^{th} and $(9-j)^{th}$ choice, he will have his MMS value.

As mentioned above, see Algorithm 14 for the complete approach. ■

6.6 MMS Guarantees on Subsets of $\mathcal{N}$

In the previous sections of this chapter, we have focussed solely on the problem of ensuring each agent i achieves a value of $\mathtt{MMS}_i(n, \mathcal{M})$ (or some approximation to it). [30] introduced another concept of fairness relating to MMS which alters the application granularity of the MMS guarantee.

Definition 6.6.1. *An allocation $A_1, \ldots, A_n$ satisfies a γ approximate pairwise maximin share*

(PMMS) guarantee if

$$\forall i, j \in \mathcal{N} : \mathtt{val}_i(A_i) \geq \gamma \cdot \mathtt{MMS}_i\left(2, A_i \cup A_j\right).$$

Specifically, [30] gives an algorithm that admits a $2/\left(1+\sqrt{5}\right) \approx 0.618$ approximation and they leave it open whether a 1-PMMS allocation always exists no matter the agents, goods, and valuations.

In this section we show the following theorem.

Theorem 6.6.2. *Algorithm 15 achieves a* $\left(\sqrt{17}-1\right)/4 \approx 0.781$ *approximation to PMMS and runs in polynomial time.*

The rest of this section is devoted entirely to the proof of this theorem. As the algorithm's complexity is clearly polynomial, we focus solely on the approximation factor.

For sake of notational convenience throughout this proof we will let $\alpha = \left(3+\sqrt{17}\right)/4$ and $\gamma = \left(\sqrt{17}-1\right)/4$ (the approximation factor). Now let i be any general agent. To prove the theorem, we show that i maintains the desired ratio w.r.t. every other agent throughout the algorithm. We start with two useful observations.

Observation 6.6.3. *If an agent believes the value of his bundle and another bundle is u and v respectively, then he achieves an approximation ratio of at least* $\frac{u}{(u+v)/2} = \frac{2u}{u+v}$ *w.r.t. that bundle.*

Observation 6.6.4. *If i is flagged, then $\forall j \neq i$, we have that at every point of the algorithm after i receives his good in the first phase, at least one of the following hold.*

1. *j has one item.*
2. *i values j's bundle at a value $\leq 1 + 1/\alpha$ times of his own.*

Lemma 6.6.5. *If i is flagged, then $\forall j \neq i$ he achieves the desired ratio w.r.t. j.*

Proof. If at the end of the algorithm, j has only one item then clearly i achieves a ratio of ≥ 1. Otherwise, by Observation 6.6.4, we have that i must value j's bundle $\leq 1 + 1/\alpha$ of his own. Thus by Observation 6.6.3 i achieves a ratio of:

$$\geq \frac{2}{1+1+1/\alpha} = \frac{2}{2+1/\alpha} = \gamma.$$

■

Now for further notational convenience if i is not flagged let $p = p_i$ and $q = q_i$.

Lemma 6.6.6. *If i is not flagged, then from the beginning of the envy-cycle-elimination phase to the end of the algorithm, he achieves the desired ratio w.r.t. Φ_j for all j.*

Proof. There are five cases to consider.

1. *j is flagged.*
   In this case Φ_j has only one good so i must have a ratio of ≥ 1.
2. *j is not flagged and $j = i$.*
   In this case $\mathtt{val}_i\left(\Phi_j\right) = p + q$ and i has at least this value due to the properties of the envy-cycle-elimination phase. Thus, i must have a ratio of ≥ 1.

// Phase 1 (first good)
for $i = 1, \ldots, n$ **do**
  Let g be the most valuable remaining good in i's view.
  Let $j < i$ be the agent whom i believes has the most valuable good so far of the *non-flagged* agents.
  if j's item has value to i at least $((3 + \sqrt{17})/4) \cdot \mathtt{val}_i(g)$ **then**
    Give j's item to i.
    Flag i.
    Swap the indices of i and j (so j is flagged).
    $i \leftarrow i - 1$ (i.e. redo the for loop for the new i).
  else
    Give g to i.
  end if
end for
Let p_i be the value of the good i currently has (to i).
// Phase 2 (reverse lexicographic draft)
// Run a single round draft on the remaining goods for all non-flagged agents in *reverse* lexicographic order. i).
for $i = n, \ldots, 1$ **do**
  if i is not flagged **then**
    Let i take the most valuable good of those remaining.
  end if
end for
For every non-flagged agent i, let q_i be the value of the good i received in this phase (to i).
For every agent i, let Φ_i be the set of (at most two) goods they currently have.
// Phase 3 (envy-cycle-elimination)
Create a directed envy graph $G = (V, E)$ where V represents the agents and there is an edge (i, j) iff i is flagged and believes j's bundle is worth at least his own, or i is not flagged and believes j's bundle is worth at least $\max(i\text{'s value of his own bundle}, \sqrt{2}p_i)$.
while there are unallocated goods **do**
  while there is a cycle in G **do**
    Take any cycle and eliminate it by having each agent in the cycle give his bundle to the agent before him in the cycle (and receive the bundle from the agent after him). Update the edges.
  end while
  As there is no cycle in G, there exists at least one agent who has no incoming edges. Give one of the items not already given out to one of these agents.
end while

Algorithm 15: $\left(\sqrt{17} - 1\right)/4 \approx 0.781$-PMMS

3. *j is not flagged, $j > i$, and i's bundle is a (not necessarily strict) superset of Φ_i.*
   In this case Φ_j has two goods, neither of which are strictly more valuable to i than p and i has at least one good of value p in his own bundle. Thus, i must have a ratio of ≥ 1.
4. *j is not flagged, $j > i$, and i's bundle is not a (not necessarily strict) superset of Φ_i.*
   In this case Φ_j has two goods, neither of which are strictly more valuable to i than p and therefore $\mathtt{val}_i\left(\Phi_j\right) \leq 2p$. As i's bundle is not a superset of Φ_i, he must have taken part in a cycle elimination. Thus i must have a value of at least $\sqrt{2}p$ for his own bundle. By Observation 6.6.3 i must then have a ratio of:

$$\geq \frac{2\sqrt{2}p}{\sqrt{2}p + 2p} = \frac{2}{1+\sqrt{2}} > \gamma.$$

5. *j is not flagged and $j < i$.*
   In this case Φ_j has two goods, one of which i values $\leq \alpha p$ (as otherwise i would have taken it and become flagged) and the other he values $\leq q$ (as otherwise he would have taken it in the reverse lexicographic draft phase). Thus, by Observation 6.6.3 i must achieve a ratio of:

$$\geq \frac{2(p+q)}{p+q+\alpha p+q} = \frac{2(p+q)}{(1+\alpha)p+2q} = \frac{2(1+x)}{1+\alpha+2x}.$$

   If $x = q/p \in [0,1]$. Simultaneously, i must achieve a ratio of:

$$\begin{aligned}
&\geq \frac{i\text{'s value for } i\text{'s bundle}}{i\text{'s value for } i\text{'s bundle} + i\text{'s value for } \Phi_j\text{'s second item}} \\
&\geq \frac{p+q}{p+q+q} \\
&= \frac{p+q}{p+2q} \\
&= \frac{1+x}{1+2x}.
\end{aligned}$$

   So i must achieve a ratio of:

$$\geq \min_{x\in[0,1]} \max\left(\frac{2(1+x)}{1+\alpha+2x}, \frac{1+x}{1+2x}\right).$$

   Clearly as x increases, the first term in the maximization increases while the second decreases. Furthermore, after a bit of arithmetic we see that they intersect only at $x = (\alpha-1)/2$. Thus substituting this value of x into either of the terms in the maximization give the result of the minimum which, after some further arithmetic, yields that the minimum is:

$$\geq \frac{1+\alpha}{2\alpha} = \gamma.$$

■

Lemma 6.6.7. *If i is not flagged, then from the beginning of the envy-cycle-elimination phase to the end of the algorithm, he achieves the desired ratio w.r.t. the bundle which contains Φ_j for all j.*

Proof. If Φ_j is equivalent to this bundle then Lemma 6.6.6 applies so we can safely assume Φ_j is a strict subset of this bundle. If u is the value of i's bundle to himself, then $\mathtt{val}_i$ (this bundle) $\leq \max\left(u, \sqrt{2}p\right) + q$. Thus, by Observation 6.6.3 i must achieve a ratio of:

$$\geq \frac{2u}{u + \max\left(u, \sqrt{2}p\right) + q} = \min\left(\frac{2u}{u+u+q}, \frac{2u}{u+\sqrt{2}p+q}\right).$$

Note however that (as $p + q \leq u$ and $q \leq p$ imply $q \leq u/2$)

$$\frac{2u}{u+u+q} \geq \frac{2u}{u+u+u/2} = \frac{4}{5} > \gamma,$$

and

$$\frac{2u}{u+\sqrt{2}p+q} \geq \frac{2(p+q)}{p+q+\sqrt{2}p+q} \geq \frac{2(p+q)}{p+q+\sqrt{2}p+\sqrt{2}q} = \frac{2}{1+\sqrt{2}} > \gamma.$$

■

Combining the results of Lemmas 6.6.5 and 6.6.7 complete the proof of Theorem 6.6.2.

Bibliography

[1] N. Alon. “Splitting necklaces”. In: *Advances in Mathematics* 63 (1987), pages 241–253 (page 71).

[2] N. Alon, F. Fischer, A. D. Procaccia, and M. Tennenholtz. “Sum of Us: Strategyproof Selection from the Selectors”. In: *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. 2011, pages 101–110 (pages 7, 17).

[3] G. Amanatidis, G. Birmpas, and E. Markakis. “On Truthful Mechanisms for Maximin Share Allocations”. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*. Forthcoming. 2016 (page 76).

[4] G. Amanatidis, E. Markakis, A. Nikzad, and A. Saberi. “Approximation Algorithms for Computing Maximin Share Allocations”. In: *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP)*. 2015, pages 39–51 (pages 76, 99).

[5] O. Arzi, Y. Aumann, and Y. Dombb. “Throw One’s Cake — and Eat It Too”. In: *Proceedings of the 4th International Symposium on Algorithmic Game Theory (SAGT)*. 2011, pages 69–80 (pages 3, 47).

[6] A. Asadpour and A. Saberi. “An approximation algorithm for max-min fair allocation of indivisible goods”. In: *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*. 2007, pages 114–121 (page 75).

[7] Y. Aumann, Y. Dombb, and A. Hassidim. “Computing Socially-Efficient Cake Divisions”. In: *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2013, pages 343–350 (pages 49, 51).

[8] H. Aziz, O. Lev, N. Mattei, J. S. Rosenschein, and T. Walsh. “Strategyproof Peer Selection”. In: *arXiv preprint arXiv:1604.03632* (2016) (pages 20, 30, 31).

[9] H. Aziz and S. Mackenzie. *A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents*. CoRR abs/1604.03655. 2016 (pages 3, 35, 37, 71).

[10] E. Balkanski, S. Brânzei, D. Kurokawa, and A. D. Procaccia. “Simultaneous Cake Cutting”. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*. 2014, pages 566–572 (page 49).

[11] N. Bansal and M. Sviridenko. “The Santa Claus problem”. In: *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*. 2006, pages 31–40 (page 75).

[12] X. Bei, N. Chen, X. Hua, B. Tao, and E. Yang. “Optimal Proportional Cake Cutting with Connected Pieces”. In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*. 2012, pages 1263–1269 (pages 36, 47, 49).

[13] D. Berga and R. Gjorgjiev. *Impartial Social Rankings*. Manuscript. 2014 (pages 8, 21).

[14] I. Bezáková and V. Dani. “Allocating Indivisible Goods”. In: *SIGecom Exchanges* 5.3 (2005), pages 11–18 (page 75).

[15] B. Bollobás. *Random Graphs*. 2nd. Cambridge University Press, 2001 (page 104).

[16] S. Bouveret and M. Lemaître. "Characterizing conflicts in fair division of indivisible goods using a scale of criteria". In: *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2014, pages 1321–1328 (pages 72, 73, 75, 76, 108).

[17] S. J. Brams, P. H. Edelman, and P. C. Fishburn. "Fair division of indivisible items". In: *Theory and Decision* 55.2 (2003), pages 147–180 (page 75).

[18] S. J. Brams, M. Feldman, J. Morgenstern, J. K. Lai, and A. D. Procaccia. "On Maxsum Fair Cake Divisions". In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*. 2012, pages 1285–1291 (pages 36, 41, 49).

[19] S. J. Brams, M. Kilgour, and C. Klamler. "Two-Person Fair Division of Indivisible Items: An Efficient, Envy-Free Algorithm". In: *Notices of the AMS* 61.2 (2014), pages 130–141 (page 75).

[20] S. J. Brams and A. D. Taylor. "An Envy-Free Cake Division Protocol". In: *The American Mathematical Monthly* 102.1 (1995), pages 9–18 (pages 3, 35, 54, 71).

[21] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996 (pages 74, 75).

[22] S. J. Brams, A. D. Taylor, and W. S. Zwicker. "A moving-knife solution to the four-person envy-free cake-division problem". In: *Proceedings of the American Mathematical Society* 125.2 (1997), pages 547–554 (page 54).

[23] S. Brânzei, I. Caragiannis, and D. Kurokawa A. D. Procaccia. *An Algorithmic Framework for Strategic Fair Division*. 2016 (page 3).

[24] S. Brânzei and P. B. Miltersen. "A Dictatorship Theorem for Cake Cutting". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2015, pages 482–488 (pages 49, 51).

[25] S. Brânzei and P. B. Miltersen. "Equilibrium Analysis in Cake Cutting". In: *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2013, pages 327–334 (pages 49, 51, 54, 56).

[26] S. Brânzei, A. D. Procaccia, and J. Zhang. "Externalities in Cake Cutting". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2013, pages 55–61 (page 49).

[27] E. Budish. "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes". In: *Journal of Political Economy* 119.6 (2011), pages 1061–1103 (pages 3, 72, 74, 76).

[28] C. Busch, M. S. Krishnamoorthy, and M. Magdon-Ismail. "Hardness Results for Cake Cutting". In: *Bulletin of the EATCS* 86 (2005), pages 85–106 (page 36).

[29] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. "The Unreasonable Fairness of Maximum Nash Product". In: *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*. Forthcoming. 2016 (page 76).

[30] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. "The Unreasonable Fairness of Maximum Nash Welfare". In: *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*. 2016, pages 305–322 (pages 3, 73, 74, 110, 111).

[31] I. Caragiannis, J. K. Lai, and A. D. Procaccia. "Towards more expressive cake cutting". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2011, pages 127–132 (pages 36, 49).

[32] Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia. "Truth, Justice, and Cake Cutting". In: *Games and Economic Behavior* 77 (2013). Preliminary version in AAAI'10, pages 284–297 (pages 36, 37, 41, 44, 47, 49, 51).

[33] G. de Clippel, H. Moulin, and N. Tideman. “Impartial division of a dollar”. In: *Journal of Economic Theory* 139 (2008), pages 176–191 (pages 8, 21).

[34] Y. J. Cohler, J. K. Lai, D. C. Parkes, and A. D. Procaccia. “Optimal Envy-Free Cake Cutting”. In: *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*. 2011, pages 626–631 (pages 36, 47, 49).

[35] A. Demers, S. Keshav, and S. Shenker. “Analysis and simulation of a fair queueing algorithm”. In: *Proceedings of the ACM Symposium on Communications Architectures & Protocols (SIGCOMM)*. 1989, pages 1–12 (page 72).

[36] X. Deng, Q. Qi, and A. Saberi. “Algorithmic Solutions for Envy-Free Cake Cutting”. In: *Operations Research* 60.6 (2012), pages 1461–1476 (page 51).

[37] X. Deng, Q. Qi, and A. Saberi. *On the Complexity of Envy-Free Cake Cutting*. CoRR abs/0907.1334. 2009 (page 37).

[38] J. P. Dickerson, J. Goldman, J. Karp, A. D. Procaccia, and T. Sandholm. “The Computational Rise and Fall of Fairness”. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*. 2014, pages 1405–1411 (pages 99, 100).

[39] J. Douceur. “Paper rating vs. paper ranking”. In: *Operating Systems Review* 43 (2009), pages 117–121 (page 8).

[40] L. E. Dubins and E. H. Spanier. “How to cut a cake fairly”. In: *American Mathematical Monthly* 68.1 (1961), pages 1–17 (page 50).

[41] B. Edelman, M. Ostrovsky, and M. Schwarz. “Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords”. In: *American Economic Review* 97.1 (2007), pages 242–259 (page 54).

[42] J. Edmonds and K. Pruhs. “Balanced allocations of cake”. In: *Proceedings of the 47th Symposium on Foundations of Computer Science (FOCS)*. 2006, pages 623–634 (page 51).

[43] J. Edmonds and K. Pruhs. “Cake cutting really is not a piece of cake”. In: *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2006, pages 271–278 (pages 36, 51).

[44] S. Even and A. Paz. “A note on cake-cutting”. In: *Discrete Applied Mathematics* 7 (1984), pages 285–296 (page 50).

[45] F. Fischer and M. Klimm. “Optimal Impartial Selection”. In: *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*. 2014, pages 803–820 (page 8).

[46] P. C. Fishburn. “Condorcet social choice functions”. In: *SIAM Journal on applied Mathematics* 33.3 (1977), pages 469–489 (page 20).

[47] C. Fong. “Social preferences, self-interest, and the demand for redistribution”. In: *Journal of Public Economics* 82.2 (2001), pages 225–246 (page 1).

[48] Y. Gal, M. Mash, A. D. Procaccia, and Y. Zick. *Which Is the Fairest (Rent Division) of Them All?* Manuscript. 2016 (page 3).

[49] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. “Dominant Resource Fairness: Fair Allocation of Multiple Resource Types”. In: *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI)*. 2011, pages 24–37 (page 71).

[50] J. Goldman and A. D. Procaccia. “Spliddit: Unleashing Fair Division Algorithms”. In: *SIGecom Exchanges* 13.2 (2014), pages 41–46 (pages 1, 49, 74).

[51] R. Haenni. “Aggregating Referee Scores: an Algebraic Approach”. In: *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC)*. 2008, pages 277–288 (page 8).

[52] C. Harris. "Existence and Characterization of Perfect Equilibrium in Games of Perfect Information". In: *Econometrica* 53.3 (1985), pages 613–628 (page 57).

[53] G. A. Hazelrigg. *Dear Colleague Letter: Information to Principal Investigators (PIs) Planning to Submit Proposals to the Sensors and Sensing Systems (SSS) Program October 1, 2013, Deadline*. Retrieved on June 17, 2014. 2013. URL: http://www.nsf.gov/pubs/2013/nsf13096/nsf13096.jsp?WT.mc_id=USNSF_25#reference1 (pages 5, 8).

[54] H. Hellwig and W. Leininger. "On the Existence of Subgame-Perfect Equilibrium in Infinite-Action Games of Perfect Information". In: *Journal of Economic Theory* 45 (1987), pages 55–75 (page 57).

[55] T. Hill. "Partitioning general probability measures". In: *Annals of Probability* 15.2 (1987), pages 804–813 (page 75).

[56] R. Holzman and H. Moulin. "Impartial nominations for a prize". In: *Econometrica* 81.1 (2013), pages 173–196 (pages 7, 8).

[57] A. Kahng, Y. Kotturi, C. Kulkarni, D. Kurokawa, and A. D. Procaccia. *Randomized Algorithms for Impartial Peer Ranking*. Manuscript. 2017 (page 2).

[58] I. Kash, A. D. Procaccia, and N. Shah. "No Agent Left Behind: Dynamic Fair Division of Multiple Resources". In: *Journal of Artificial Intelligence Research* 51 (2014), pages 579–603 (page 71).

[59] D. Kurokawa, J. K. Lai, and A. D. Procaccia. "How to cut a cake before the party ends". In: *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*. 2013, pages 555–561 (pages 3, 49, 51).

[60] D. Kurokawa, O. Lev, J. Morgenstern, and A. D. Procaccia. "Impartial Peer Review". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2015, pages 582–588 (page 2).

[61] D. Kurokawa, A. D. Procaccia, and N. Shah. "Leximin Allocations in the Real World". In: *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*. 2015, pages 345–362 (pages 3, 49).

[62] D. Kurokawa, A. D. Procaccia, and J. Wang. *Fair Enough: Guaranteeing Approximate Maximin Shares*. Manuscript. 2017 (page 3).

[63] D. Kurokawa, A. D. Procaccia, and J. Wang. "When can the maximin share guarantee be guaranteed?" In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. 2016, pages 523–529 (page 3).

[64] M. Li, J. Zhang, and Q. Zhang. "Truthful Cake Cutting Mechanisms with Externalities: Do Not Make Them Care for Others Too Much!" In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2015, pages 589–595 (page 49).

[65] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. "On approximately fair allocations of indivisible goods". In: *Proceedings of the 6th ACM Conference on Economics and Computation (EC)*. 2004, pages 125–131 (pages 37, 75, 84).

[66] A. Mackenzie. *Impartiality and symmetry*. Manuscript. 2014 (pages 8, 21).

[67] E. Markakis and C.-A. Psomas. "On Worst-Case Allocations in the Presence of Indivisible Goods". In: *Proceedings of the 5th Conference on Web and Internet Economics (WINE)*. 2011, pages 278–289 (page 75).

[68] E. Maskin. "Nash Equilibrium and Welfare Optimality". In: *Review of Economic Studies* 66 (1999). First circulated in 1977, pages 23–38 (page 58).

[69] A. Maya and N. Nisan. "Incentive Compatible Two Player Cake Cutting". In: *Proceedings of the 8th Conference on Web and Internet Economics (WINE)*. 2012, pages 170–183 (page 51).

[70] C. J. H. McDiarmid. "Clutter percolation and random graphs". In: *Mathematical Programming Study* 13 (1980), pages 17–25 (page 104).

[71] M. Merrifield and D. Saari. "Telescope Time Without Tears: a Distributed Approach to Peer Review". In: *Astronomy and Geophysics* 50.4 (2009), pages 2–6 (pages 5, 8, 17).

[72] M. Mitzenmacher. *NSF Reviewing Trial Run*. Retrieved on June 17, 2014. 2013. URL: http://mybiasedcoin.blogspot.com/2013/06/nsf-reviewing-trial-run.html (page 5).

[73] D. Monderer and M. Tennenholtz. "Strong mediated equilibrium". In: *Artificial Intelligence* 173.1 (2009), pages 180–195 (page 57).

[74] E. Mossel and O. Tamuz. "Truthful fair division". In: *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT)*. 2010, pages 288–299 (page 51).

[75] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003 (page 74).

[76] H. Moulin. "Uniform externalities: Two axioms for fair allocation". In: *Journal of Public Economics* 43.3 (1990), pages 305–326 (page 72).

[77] A. Nicolò and Y. Yu. "Strategic divide and choose". In: *Games and Economic Behavior* 64.1 (2008), pages 268–289 (pages 51, 54).

[78] O. Nierstrasz. "Identify the champion". In: *Pattern Languages of Program Design*. Edited by N. Harrison, B. Foote, and H. Rohnert. Volume 4. Addison-Wesley, 2000, pages 539–556 (page 8).

[79] A. Othman, C. H. Papadimitriou, and A. Rubinstein. "The complexity of fairness through equilibrium". In: *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*. 2014, pages 209–226 (page 3).

[80] A. Othman, T. Sandholm, and E. Budish. "Finding approximate competitive equilibria: Efficient and fair course allocation". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2010, pages 873–880 (page 3).

[81] D. C. Parkes, A. D. Procaccia, and N. Shah. "Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities". In: *Proceedings of the 13th ACM Conference on Economics and Computation (EC)*. 2012, pages 808–825 (page 71).

[82] E. Pazner and D. Schmeidler. "Egalitarian equivalent allocations: A new concept of economic equity". In: *Quarterly Journal of Economics* 92.4 (1978), pages 671–687 (page 75).

[83] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. "Tuned models of peer assessment in MOOCs". In: *arXiv preprint arXiv:1307.2579* (2013) (page 19).

[84] A. D. Procaccia. "Cake Cutting: Not Just Child's Play". In: *Communications of the ACM* 56.7 (2013), pages 78–87 (pages 52, 71).

[85] A. D. Procaccia. "Thou Shalt Covet Thy Neighbor's Cake". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 2009, pages 239–244 (pages 36, 37, 49, 51).

[86] A. D. Procaccia and J. Wang. "Fair Enough: Guaranteeing Approximate Maximin Shares". In: *Proceedings of the 14th ACM Conference on Economics and Computation (EC)*. 2014, pages 675–692 (page 76).

[87] Ariel D. Procaccia. *NSF (actually) reviewing via social choice*. Retrieved on June 17, 2014. 2013. URL: http://agtb.wordpress.com/2013/06/10/nsf-actually-reviewing-via-social-choice/ (page 5).

[88] J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998 (pages 36, 50–53).

[89] M. Roos, J. Rothe, and B. Scheuermann. “How to Calibrate the Scores of Biased Reviewers by Quadratic Programming”. In: *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*. 2011, pages 255–260 (page 8).

[90] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008 (page 54).

[91] H. Steinhaus. “The problem of fair division”. In: *Econometrica* 16 (1948), pages 101–104 (page 35).

[92] W. Stromquist. “Envy-free cake divisions cannot be found by finite protocols”. In: *The Electronic Journal of Combinatorics* 15 (2008), #R11 (pages 3, 37).

[93] W. Stromquist. “How to cut a cake fairly”. In: *American Mathematical Monthly* 87.8 (1980), pages 640–644 (page 58).

[94] F. E. Su. “Rental harmony: Sperner’s lemma in fair division”. In: *American Mathematical Monthly* 106.10 (1999), pages 930–942 (page 37).

[95] H. K. Suen. “Peer assessment for massive open online courses (MOOCs)”. In: *The International Review of Research in Open and Distributed Learning* 15.3 (2014) (page 19).

[96] S. Tamura and S. Ohseto. “Impartial nomination correspondences”. In: *Social Choice and Welfare* 43 (2014), pages 47–54 (page 8).

[97] W. Thomson. “Fair Allocation Rules”. In: *Handbook of Social Choice and Welfare*. Edited by K. J. Arrow, A. Sen, and K. Suzumura. Volume 2. North-Holland, 2010. Chapter 21 (page 20).

[98] R. V. Vohra. *A Mechanism Design Approach to Peer Review*. Retrieved on June 17, 2014. 2013. URL: http://theoryclass.wordpress.com/2013/06/06/a-mechanism-design-approach-to-peer-review/ (page 5).

[99] G. J. Woeginger. “A polynomial-time approximation scheme for maximizing the minimum machine completion time”. In: *Operations Research Letters* 20.4 (1997), pages 149–154 (pages 75, 98).

[100] G. J. Woeginger and J. Sgall. “On the complexity of cake cutting”. In: *Discrete Optimization* 4 (2007), pages 213–220 (pages 36, 51).

[101] L. Zhou. “Inefficiency of strategy-proof allocation mechanisms in pure exchange economies”. In: *Social Choice and Welfare* 8 (1991), pages 247–254 (page 51).