Beyond Cake Cutting: Allocating Homogeneous Divisible Goods

Ioannis Caragiannis
Aarhus University
Aarhus, Denmark
iannis@cs.au.dk

Vasilis Gkatzelis
Drexel University
Philadelphia, PA, United States
gkatz@drexel.edu

Alexandros Psomas
Purdue University
West Lafayette, IN, United States
apsomas@cs.purdue.edu

Daniel Schoepflin
Drexel University
Philadelphia, PA, United States
schoep@drexel.edu

ABSTRACT

The problem of fair division known as "cake cutting" has been the focus of multiple papers spanning several decades. The most prominent problem in this line of work has been to bound the query complexity of computing an envy-free outcome in the Robertson-Webb query model. However, the root of this problem's complexity is somewhat artificial: the agents' values are assumed to be additive across different pieces of the "cake" but *infinitely complicated* within each piece. This is unrealistic in most of the motivating examples, where the cake represents a *finite* collection of homogeneous goods.

We address this issue by introducing a fair division model that more accurately captures these applications: the value that an agent gains from a given good depends only on the amount of the good they receive, yet it can be an arbitrary function of this amount, allowing the agents to express preferences that go beyond standard cake cutting. In this model, we study the query complexity of computing allocations that are not just envy-free, but also approximately Pareto optimal among all envy-free allocations. Using a novel flow-based approach, we show that we can encode the ex-post feasibility of randomized allocations via a polynomial number of constraints, which reduces our problem to solving a linear program.

CCS CONCEPTS

• **Theory of computation** → *Mathematical optimization.*

KEYWORDS

Fair division; Cake cutting; Envy-freeness; Pareto efficiency

ACM Reference Format:
Ioannis Caragiannis, Vasilis Gkatzelis, Alexandros Psomas, and Daniel Schoepflin. 2022. Beyond Cake Cutting: Allocating Homogeneous Divisible Goods. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

The "cake-cutting" problem has played a central role in the history of fair division. The commonly used story behind this problem involves a heterogeneous cake with a variety of different toppings (e.g., a region with sprinkles, a region with more icing, or a cherry sitting at the top), and the goal is to fairly divide this cake among a group of children with different preferences. In reality, this is just a metaphor aiming to capture complicated resource allocation settings, where each region of the cake may correspond to a different resource and each agent competing for these resources may have a diverse and complicated valuation function.

The cake is mathematically represented using the interval $[0, 1]$ and a piece of the cake corresponds to a (possibly not contiguous) subset of this interval. The agents' preferences are encoded by a valuation function that maps each sub-interval to a number, and the value of an agent for a set of disjoint intervals is equal to the sum over all these intervals of her value for each interval. So, the agents' values for each sub-interval can be highly complicated, but their value across these intervals is additive.

An important goal in cake-cutting is to find some assignment of pieces to the agents that is "envy-free", i.e., each agent (weakly) prefers her piece over the piece allocated to any other agent. The main obstacle in achieving this goal is the fact that, due to the high complexity of the agents' preferences, they cannot be succinctly reported to the algorithm. In the classic Robertson-Webb query model [23], access to the agents' valuations is provided only through two types of queries: i) *value queries*, that take as input some interval and return the value of the agent for it, and ii) *cut queries*, that take as input a point $\ell \in [0, 1]$ and a value v, and return the leftmost point $r \in [\ell, 1]$ such that the value of the agent for the interval $[\ell, r]$ is equal to v.

A trivial way to achieve envy-freeness is to throw away all of the cake and thus ensure that no agent has anything to be envious about. As it turns out, even the simple restriction that none of the cake can be thrown away is enough to really complicate things. For instances involving up to three agents, this was solved in the 1960s [9] but, up until just a few years ago, we did not even know whether a *bounded* (let alone polynomial) number of queries would be sufficient for instances involving four agents. This was recently settled in the affirmative by Aziz and Mackenzie [3, 4], but the proposed protocols are very complicated, and the required number of queries is superexponential in the number of agents.

Despite the extensive amount of attention that this problem has garnered, its apparent intractability is partly due to the artificial complexity of the model. For example, in many of the motivating application domains, the cake actually comprises a set of distinct

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

homogeneous resources and each agent cares only about *how much* of each resource she receives rather than *which part* of it. In other words, when allocating a piece of the proverbial cherry on top of the cake to an agent, she may care about the size of this piece, but not worry about whether it came from the front or the back side of the cherry (it is all just cherry anyway!). But, introducing this type of structure into the cake-cutting model reduces it to the very special case of *piecewise constant valuations*: the $[0, 1]$ interval is partitioned into sub-intervals, each of which corresponds to one of the resources, and each agent's value is constant within each sub-interval. This captures a limited range of agent preferences and even computing a welfare-maximizing envy-free outcome becomes trivial [14], since the agents can just directly report their (constant) value for each resource. The complexity of the cake-cutting model lies in the assumption that there is no such structure in general, even if the cake is partitioned into arbitrarily small pieces.

In this paper, we propose an alternative model that more accurately captures these applications, while also allowing natural types of agent preferences that are beyond the scope of the cake cutting model (e.g., complementarity and substitutability). Achieving envy-freeness in this model is easy even without discarding any of the resources, just like it is in most motivating examples; one can just divide each resource equally among the agents. We instead aim to compute not just any envy-free allocation, but rather one that is (approximately) Pareto optimal among all envy-free allocations, which is not guaranteed if we just divide each resource equally.

1.1 Results and Techniques

We study the problem of fairly allocating a set of divisible homogeneous resources among a set of agents. Deviating from the cake-cutting literature, we use a model where each agent's value depends only on *how much* of each resource they receive rather than *which part*. We allow each agent's value to be a complicated function of the amount of each resource that they receive, and we use cut and value queries, analogous to the Robertson-Webb query model, to "discover" the structure of these functions.

This model provides a better abstraction for many real-world applications. For example, the resource could represent the shares of a company where the value of an agent may increase dramatically upon receiving more than half of all the shares. Alternatively, if the resource being distributed is energy, then there may be a limit on how much energy an agent can utilize each day, so being allocated more than that would not offer her any additional value. These are utilities that the standard cake cutting model cannot capture.

In Section 3, we explain why, in contrast to the cake-cutting model, the use of randomization in our model can enrich the space of achievable (expected) utility vectors, motivating the use of lotteries over deterministic outcomes. We focus on "fair" lotteries that satisfy *ex-ante* envy-freeness (i.e., envy-freeness with respect to expected utilities) and our main goal is to compute one that is (approximately) Pareto efficient among them. Rather than aiming for *ex-post* Pareto efficiency (which compares the outcome of the lottery to all *deterministic* envy-free outcomes) we instead aim for the stronger guarantee of *ex-ante* Pareto efficiency (which compares the lottery to the richer space of all envy-free lotteries).

In Section 4, we prove that finding an ex-ante envy-free lottery that is ϵ-approximate ex-ante Pareto efficient among all ex-ante envy-free lotteries would require $\Omega(1/\epsilon)$ queries even for instances with just two agents and two resources.

In Section 5, we complement this negative result by providing an algorithm that finds such a lottery using only $O(1/\epsilon^2)$ queries for any constant number of agents and resources. We generalize this to the case of n agents and m resources using $O(nm/\epsilon^2)$ queries. The algorithm begins by issuing a sequence of value queries for each agent-resource pair, providing a discretized approximation of the true valuations. It then computes a lottery by solving a linear program using these approximate valuations. The main technical challenge is to ensure that the lottery computed by this linear program is *ex-post feasible*, i.e., that the randomized allocation we obtain can be decomposed into a distribution over feasible deterministic outcomes. We achieve this through a flow-based technique which may be of independent interest.

Finally, in Section 6, we conclude with some observations and open problems for future research within the new model.

1.2 Related Work

Cake-cutting. A procedure for computing envy-free outcomes with a finite (but potentially unbounded) number of queries in the cake cutting model was found by Brams and Taylor [10]. Kurokawa et al. [19] gave a bounded protocol for any number of agents with piecewise linear valuations. Even for just four agents with general valuations, it was unknown if a protocol producing an envy-free allocation using a bounded number of queries exists, until the result of Aziz and Mackenzie [4]. They then extended their own work to achieve a procedure for n agents using a bounded number of queries, but the number is a tower of exponents of n [3]. Amanatidis et al. [2] improved the number of queries for four agents, but a further improvement on the general n-agent case has remained elusive, so there is still a massive gap between the upper bound and the lower bound of $\Omega(n^2)$ queries found by Procaccia [21].

Computing efficient envy-free allocations. The type of guarantee that we seek in this paper, i.e., to find an envy-free solution that is efficient among other envy-free ones, has also been examined in the context of the standard cake-cutting model. Reijnierse and Potters [22] demonstrated that if the entire valuation function of each agent is reported to a mechanism, then an approximately Pareto efficient allocation among all envy-free allocations can be found by approximating the valuation functions by piecewise constant valuation functions. Cohler et al. [14] extended this result by providing an efficient algorithm to compute an approximately welfare-maximizing allocation among all envy-free allocations. Our work can then be viewed as an analogue to [14] in our model. Whereas their result applies in the standard cake-cutting model and thereby can avoid the use of randomization, ours leverages randomness to apply to more general valuation functions than can be captured by cake-cutting. Bei et al. [5] studied a similar objective of finding welfare-maximizing connected allocations but, instead, among the set of proportional allocations in the standard cake-cutting model.[1]

[1] In standard cake-cutting, any envy-free complete allocation is always proportional, unlike in our model as we demonstrate in Section 3.

Notably, fairness constraints may decrease efficiency. Brams et al. [8] examined the standard cake-cutting model and identified sufficient conditions for a welfare-maximizing envy-free allocation to be Pareto efficient among the set of *all* (not necessarily envy-free) allocations. They also provided examples where no welfare-maximizing envy-free allocation is Pareto efficient among the set of all allocations although the loss in efficiency they show is small. In a similar vein, Caragiannis et al. [13] quantified the extent to which various fairness notions may affect efficiency and demonstrated that enforcing envy-freeness as a hard constraint can greatly decrease the maximum obtainable welfare. Brams et al. [8] also pointed out the question of finding an efficient algorithm to compute welfare-maximizing envy-free allocations which are Pareto efficient with respect to *all* allocations in the standard cake cutting model as an important and challenging open problem even in the simple case of piecewise constant valuations. In Section 6, we pose a related open question in our model which we believe is of comparable difficulty and interest.

Homogeneous goods. While the query model we examine in this paper is directly related to the query model of Robertson and Webb [23], our valuation function model is most similar to the work of Feige and Tennenholtz [16]. In that paper, the authors examine the division of a single homogeneous resource where the agents' utilities are non-linear. However, they examine an alternative notion of an agent's "fair share" (based on the average utility she would receive if all other agents had valuations identical to hers) and agents report allocations rather than reply to queries regarding their valuation functions. Buermann et al. [12] similarly examine the problem of allocating a single homogeneous good but instead assume that the available amount of the resource is unknown but drawn from some known prior distribution and the valuation functions of the agents are fully known to the designer. They then study the computational complexity of finding welfare-maximizing allocations subject to ex-ante envy-freeness (where randomness is with respect to the amount of good) and characterize the loss in efficiency by imposing envy-freeness as a constraint.

Lotteries and ex-post feasibility. The key technical challenge we face in finding ex-ante Pareto efficient solutions which can be derandomized into a distribution over feasible deterministic outcomes has been explored in many contexts in fair division. In the case of one-sided matching, Bogomolnaia and Moulin [7] gave the well-known "probabilistic serial" algorithm which directly computes expected assignments and utilizes the Birkhoff-von Neumann theorem [6, 25] to decompose the expected assignments into a lottery over feasible outcomes. The Birkhoff-von Neumann theorem was further generalized by Budish et al. [11] beyond one-to-one settings. As we discuss in Section 5, the methods of [11] cannot be adapted to our model, so we view our flow-based approach as complementary to these prior results.

2 PRELIMINARIES

We consider a setting with a set $\mathcal{N}$ of n agents and a set $\mathcal{M}$ of m divisible items. Throughout the paper, we use the terms items, goods, and resources interchangeably. An *outcome* x defines for each agent $i \in \mathcal{N}$ and item $k \in \mathcal{M}$ the fraction $x_{ik} \in [0, 1]$ of item k that is allocated to agent i. An outcome is *feasible* if no item is over-allocated, i.e., $\sum_{i \in \mathcal{N}} x_{ik} \leq 1$ for all $k \in \mathcal{M}$. A *lottery* L is a probability distribution over outcomes and it is (ex-post) *feasible* if it assigns positive probability only to feasible outcomes.

For each agent i, her preferences are defined through a set of non-decreasing and Lipschitz[2] valuation functions $\{f_{ik}\}_{k \in \mathcal{M}}$. Her utility for receiving a fraction x_{ik} of item k is $f_{ik}(x_{ik})$ and her utility for an outcome x is $u_i(x) = \sum_{k \in \mathcal{M}} f_{ik}(x_{ik})$. In other words, the utilities are weakly monotone within items and additive across items. The (expected) utility of agent i for a lottery L which assigns probability p_ℓ to each outcome x^ℓ is

$$u_i(L) = \sum_\ell p_\ell \sum_{k \in \mathcal{M}} f_{ik}(x^\ell_{ik}).$$

Let $u_i(x_j)$ denote the utility agent i would receive in outcome x if she were given the allocation of agent j. We then say that an outcome x is *envy-free* if for all agents i, j we have $u_i(x_i) \geq u_i(x_j)$. Similarly, let $u_i(L_j)$ denote the utility i would receive from lottery L if she were given the randomized allocation of agent j. A lottery L is *ex-post envy-free* if it assigns positive probability only to outcomes that are envy-free and it is *ex-ante envy-free* if it is envy-free with respect to the agents' expected utilities, i.e., for all agents i, j we have $u_i(L_i) \geq u_i(L_j)$.

An outcome x is *ϵ-Pareto optimal* with respect to a set of outcomes $\mathcal{X}$ if there exists no alternative outcome $y \in \mathcal{X}$ such that $u_i(y) \geq (1 + \epsilon) \cdot u_i(x)$ for all $i \in \mathcal{N}$ and at least one of these inequalities is strict [17, 18, 24, 26]. We say that a lottery L is *ex-post ϵ-Pareto optimal* with respect to a set of outcomes $\mathcal{X}$ if it assigns positive probability only to outcomes that are ϵ-Pareto optimal with respect to $\mathcal{X}$. A lottery L is *ex-ante ϵ-Pareto optimal* with respect to a set of lotteries $\mathcal{L}$ if there is no alternative lottery $L' \in \mathcal{L}$ with $u_i(L') \geq (1+\epsilon)u_i(L)$ for all $i \in \mathcal{N}$ and at least one strict inequality. For $\epsilon = 0$ we retrieve exact Pareto optimality. Our main result focuses on the class of ex-ante envy-free lotteries $\mathcal{E}$ and, in accordance with Cohler *et al.* [14], we say a lottery L is (approximately) *ex-ante Pareto optimal EF* if it is in $\mathcal{E}$ and also (approximately) ex-ante Pareto optimal with respect to all lotteries in $\mathcal{E}$.

Since the f_{ik} valuation functions can be highly complicated and not succinctly representable, in line with the Robertson-Webb cake-cutting model, we assume that the algorithm can gather parts of this information using two types of queries – *value* and *cut* queries. A value query $\textsc{Value}(f, z)$ takes in a valuation function and a z in $[0, 1]$ and returns the value of $f(z)$, and a cut query $\textsc{Cut}(f, v)$ returns the minimum z in $[0, 1]$ such that $f(z) = v$. Our main algorithm (see Section 5) shows that we can actually compute approximately ex-ante Pareto optimal EF lotteries using only value queries. In Section 6 we also demonstrate that cut queries are useful for computing lotteries in $\mathcal{E}$ that are *exactly ex-post* Pareto optimal with respect to any feasible outcome. Moreover, our lower bound on the query complexity of finding approximately ex-ante Pareto optimal EF lotteries in Section 4 applies to any algorithm that can use both cut and value queries.

[2] A function f is Lipschitz if $|f(x) - f(y)| \leq C|x - y|$ $\forall x, y$ and some constant C.

3 MOVING BEYOND CAKE CUTTING

As we discussed in the introduction, when allocating a set of homogeneous goods (where agents care only about the amount of each good they receive rather than what specific part of the good), the preferences in the cake cutting model are reduced to piecewise constant valuations. This is due to the restrictive assumption that the value of an agent for any two pieces is equal to the sum of her value for each of the pieces. Therefore, the agent's value for half a cherry needs to be exactly half of her value for all of the cherry (otherwise combining two halves would not add up to the whole value). In our model, this implies that every agent's valuation function for a good needs to be linear, which is unnecessarily restrictive.

Our model moves beyond this restriction, allowing us to encode complementarity or substitutability in the agents' preferences. For example, if an agent's valuation function for an item is highly convex, like "type A" in Figure 1, then the agent does not receive much value from that item unless she gets a lot of it. In fact, using an extreme convex function allows us to essentially capture the problem of allocating indivisible items: although the item can still be divided, if agents do not receive any value from it unless they get the whole item then dividing it would amount to discarding it. Similarly to the fair division literature for indivisible items, we use randomization to overcome this issue and achieve ex-ante fairness and efficiency guarantees.

The lack of linearity within an item also means that, unlike in the cake-cutting model, in our model complete allocations which are envy-free are not guaranteed to be proportional. For instance, if two agents have highly convex valuation functions for a single item (e.g., two agents of "type A") then splitting the single item equally between them is envy-free, but not proportional[3]: by convexity, her value for a $1/2$ fraction of the item is less than a $1/2$ fraction of her value for receiving the whole item.

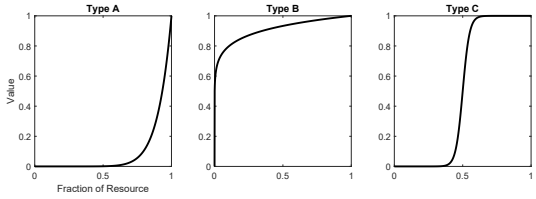

Figure 1: Examples of agent valuation functions for an item.

Efficiency with non-linear valuation functions. When the valuation functions are linear, as in the cake-cutting model, then agents are indifferent between a lottery and a (deterministic) outcome as long as their *expected* allocation of each good in the former is the same as their deterministic allocation in the latter. As a result, there is no reason to use randomization. Moving beyond linear valuation functions, however, introduces several non-trivial opportunities to leverage randomization in order to make the agents happier. On one hand, agents with convex ("type A") valuation functions are risk-seeking and would strictly prefer a lottery that gives them the full item with probability $1/2$ and nothing with probability $1/2$ over a deterministic outcome that always gives them $1/2$ of the item. On the other hand, agents with concave ("type B") valuation functions are risk-averse and would strictly prefer the deterministic outcome.

Even for an instance involving two agents and a single item, how should we allocate the item? Should we use a lottery? If so, what outcomes should the lottery randomize over, and how large does the support of a lottery need to be in order to achieve ex-ante Pareto efficiency among all envy-free lotteries? These are non-trivial questions that we urge the reader to contemplate before proceeding.

Outcomes in the utility space. Although thinking about Pareto optimal lotteries in the allocation space can be quite confusing, visualizing them in the utility space is very convenient. Each outcome defines a utility for each agent, and this point lies in some n-dimensional space *regardless of the number of items*. For example, consider an instance with two "type A" agents and a single resource. Any feasible outcome (i.e., some way of splitting the resource into two pieces) corresponds to a point in two-dimensional space whose two coordinates are the utility of the first agent and the utility of the second agent, respectively, as demonstrated in Figure 2.

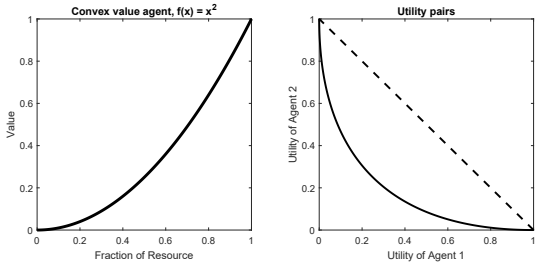

Figure 2: At right is a solid line representing the pairs of utilities of all feasible outcomes when a good is completely allocated among two agents each with the valuation function pictured at left. The dashed line indicates the possible pairs of expected utilities by randomizing between the two endpoint deterministic outcomes.

As Figure 2 illustrates, the set of utility points induced by feasible deterministic outcomes need not be convex and the shape depends on the actual valuation functions of the agents. This is in contrast to the cake-cutting model, where the utility points from feasible outcomes form a convex set [15]. This non-convexity of the set of utility vectors from deterministic outcomes provides some intuition on why adding randomness enriches the space of feasible (expected) utilities in our model, and does not do so in the cake-cutting model. This is because the set of all lotteries randomizes between the feasible outcomes can therefore yield any utility vector in the *convex hull* of the utility points achievable by deterministic outcomes. In short, randomness allows us to "convexify" the outcome space (as the dashed line does in Figure 2). Note that all ex-ante Pareto efficient lotteries lie on a face of an (up to) n-dimensional convex hull. As a result, we can conclude that all utility points on the ex-ante Pareto frontier can be achieved with a lottery of support size at most n.

Things become more complicated when combining an agent with a convex valuation function and one with a concave one, as

[3]Proportionality requires that every agent receives a utility at least $1/n$ times her utility for being allocated everything in full.

in Figure 3. The graph at the bottom of this figure demonstrates the expected utility pairs that can be achieved for these two agents using ex-ante envy-free lotteries. It is worth noting that the "all-or-nothing" lottery (which gives each agent the whole item with probability 0.5 and yields expected utility 0.5 for both of these agents) is clearly Pareto dominated by other ex-ante envy-free lotteries. The points in red correspond to the utility pairs induced by the set of ex-ante Pareto optimal EF lotteries.

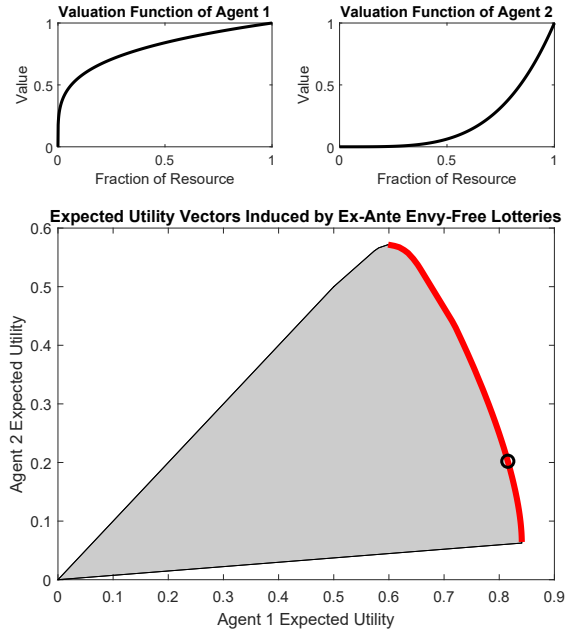

Figure 3: At top left is the concave valuation function of Agent 1. At top right is the convex valuation function of Agent 2. At the bottom, the shaded region contains all the pairs of utilities (for Agent 1 and Agent 2) induced by ex-ante EF lotteries. The red bolded faces form the ex-ante Pareto optimal EF lotteries. For example, the circled point corresponds to the lottery where Agent 1 is allocated approximately 0.75 of the good with probability 0.7 and 0.1 of the good with the remaining 0.3 probability (the allocation of Agent 2 is the complement).

The situation becomes even less clear when agents are neither strictly concave nor convex. For example, how should an item be allocated when some agents have "type C" valuations? Furthermore, things get more complicated with multiple resources. For instance, when all agents have "type A" valuations for all resources, i.e., are risk-seeking and prefer lotteries, but have different "favorite" resources, allocating each agent her favorite resource with probability 1 could Pareto dominate a uniform lottery over the entire bundle of goods. The space of outcomes and lotteries is so vast that it can be difficult to even guess what an ex-ante Pareto efficient EF lottery looks like.

4 QUERY COMPLEXITY

In this section, we study the question of query complexity in our model. Specifically, we are interested in the number of queries necessary for finding an approximately ex-ante Pareto optimal EF lottery. Our main result is a lower bound.

Theorem 4.1. *Computing an ex-ante $\frac{\epsilon}{16}$-Pareto optimal EF lottery with any deterministic adaptive protocol requires at least $\frac{1}{2\epsilon}$ queries, even just for two items and two agents.*

We briefly sketch the main intuition of the proof. Given an algorithm that terminates after k queries we simulate its behavior when the responses to all queries are as if the underlying valuation functions were linear. For every item i, there must be an interval $[x_1, x_2]$, of size at least $1/k$, such that the algorithm only knows $f_i(x_1)$ and $f_i(x_2)$ (which are x_1 and x_2, respectively, since we were responding as if everything was linear). That is, in the true underlying instance, f_i could be linear in (x_1, x_2), or could be equal to x_1 in $(x_1, x_2 - \delta)$ and then increase more rapidly to x_2, or any other non-decreasing function. The crux of the argument is that there is no lottery that is simultaneously $\epsilon/16$-Pareto optimal among all EF lotteries for all different possible instances (three different possibilities suffice).

Proof of Theorem 4.1. Consider two agents $\{1, 2\}$ and two goods $\{a, b\}$. Fix any deterministic (but possibly adaptive) algorithm asking at most $1/(2\epsilon)$ queries. We respond to all queries as if all valuation functions were linear. After $1/(2\epsilon)$ queries the algorithm terminates and outputs some lottery L.

We construct three instances, $\mathcal{I}_1$, $\mathcal{I}_2$ and $\mathcal{I}_3$, consistent with the query responses. Intuitively, in these instances good a will act as a "currency" which the two agents can exchange but good b will have complementarities "hidden" between consecutive, sufficiently-spaced queries. Notice that, by the pigeonhole principle, for item b, there must be an interval $[x_1, x_2]$ of size at least ϵ, where $x_1 > 1/2$, such that the algorithm only "knows" that $f_{2b}(x_1) = x_1$ and $f_{2b}(x_2) = x_2$. Formally, the algorithm has never asked a value query with an endpoint in (x_1, x_2), or a cut query for a value in (x_1, x_2). Our three instances defer only inside this interval.

In $\mathcal{I}_1$ the valuation functions are linear everywhere. In the second instance, $\mathcal{I}_2$, agent 1 has, again, linear valuation functions for both goods, and agent 2 has linear valuation for good a. But, for good b we have that $f_{2b}(x) = x_1 + 2(x - x_1)$ for all $x \in [x_1, x_1 + \epsilon/2)$, and $f_{2b}(x) = x_1 + \epsilon$ for all $x \in [x_1 + \epsilon/2, x_1 + \epsilon)$. In other words, for the first half of the interval $[x_1, x_1 + \epsilon]$ the slope is 2 and for the second half it is 0. f_{2b} is linear otherwise. The third instance, $\mathcal{I}_3$, is symmetric to $\mathcal{I}_2$ but with agents 1 and 2 flipped.

For $\mathcal{I}_1$, we note that any envy-free lottery must give both agents an expected total amount of the resources equal to 1. However, the algorithm cannot distinguish between $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$, so L *must* allocate an expected total amount of resources equal to 1 to both agents.

Observe that in instances $\mathcal{I}_3$ and $\mathcal{I}_2$, the maximum utility that either agent can achieve when receiving a sum of fractions of the resources at most 1 is $1 + \frac{\epsilon}{2}$. This ex-ante utility is obtained for agent 1 when she receives exactly $x_1 + \epsilon/2$ of good b and $1 - (x_1 + \epsilon/2)$ of good a, and for agent 2 at her corresponding fractions in terms

of x_1. But, these two outcomes cannot be achieved simultaneously since $x_1 > 0.5$.

Now, since both agents receive an expected total amount of resources equal to 1, in $\mathcal{I}_2$, L gives agent 1 utility equal to 1 (recall that in $\mathcal{I}_2$ agent 1 is linear everywhere). But, there is an ex-ante envy-free lottery L' for $\mathcal{I}_2$ with $u_1(L') \geq 1 + \frac{\epsilon}{8}$ and $u_2(L') \geq 1 + \frac{3\epsilon}{8}$: give to agent 2 a $x_1 + \frac{\epsilon}{2}$ fraction of resource b and a $(1 - x_1 - \frac{5}{8}\epsilon)$ fraction of resource a and the rest to the agent 1. Since L is ex-ante $\frac{\epsilon}{16}$-Pareto efficient for $\mathcal{I}_2$, it must give agent 2 utility strictly more than $1 + \frac{\epsilon}{4}$. Similarly, L must give agent 1 utility more than $1 + \frac{\epsilon}{4}$ in $\mathcal{I}_3$.

It remains to show that it is impossible for L to allocate to each agent total resources equal to 1, and simultaneously yield utility $1 + \frac{\epsilon}{4}$ in $\mathcal{I}_3$ for agent 1 and $1 + \frac{\epsilon}{4}$ in $\mathcal{I}_2$ for agent 2. Notice that, in each of $\mathcal{I}_2$ and $\mathcal{I}_3$, the expected utility of the non-linear agent is maximized (subject to the allocation constraint) when randomizing between the two outcomes/allocations that give the non-linear agent utility $1 + \frac{\epsilon}{2}$ (the linear agent is always indifferent). But, no matter how this randomization is done, one of the two expected utilities (either u_1 in $\mathcal{I}_3$ or u_2 in $\mathcal{I}_2$) is weakly less than $1 + \frac{\epsilon}{4}$. □

5 APPROXIMATELY OPTIMAL EF LOTTERIES

In this section, we develop the main positive result of the paper. We complement our lower bound from the previous section by giving an algorithm which computes an ex-ante ϵ-Pareto optimal EF lottery for any number of agents and items using only a polynomial number of queries. We do note that there is an asymptotic gap between the number of queries required by our algorithm and the lower bound we showed in the previous section. Assume, for now, that there is a single item. We start by discretizing the item: ask each agent i a series of $1/\epsilon$ value queries[4] $\textsc{Value}(f_i, x)$ for each $x \in \{\epsilon, 2\epsilon, 3\epsilon, \ldots, 1\}$. We do this with the aim of treating $1/\epsilon$ pieces of the resource (of size ϵ each) as atomic. We then seek to find a lottery over the possible allocations of the m/ϵ "indivisible" pieces.

5.1 A Failed First Attempt

We define a variable $p_{i,y}$ denoting the probability that agent i is allocated exactly y pieces of the resource for $y \in [1/\epsilon]$ and write the linear program below.

$$\begin{aligned}
&\text{maximize } \textstyle\sum_{i=1}^{n} \sum_{y=0}^{1/\epsilon} p_{i,y} f_i(y \cdot \epsilon) \\
&\text{subject to } \textstyle\sum_{i=1}^{n} \sum_{y=0}^{1/\epsilon} p_{i,y} \cdot y \cdot \epsilon = 1 \\
&\qquad \textstyle\sum_{y=0}^{1/\epsilon} p_{i,y} = 1, \quad \forall i \in \mathcal{N} \\
&\textstyle\sum_{y=0}^{1/\epsilon} p_{i,y} f_i(y \cdot \epsilon) \geq \sum_{y=0}^{1/\epsilon} p_{i',y} f_i(y \cdot \epsilon), \forall i, i' \in \mathcal{N}
\end{aligned}$$

The first set of constraints requires that the whole item is allocated. The second set restricts the variables $p_{i,y}$ so that they define a probability distribution for each agent i. The third set of constraints encodes the ex-ante envy-freeness requirements. Maximizing the total expected utility of the agents in the resulting lottery yields an ex-ante Pareto efficient EF lottery.

To arrive at an ex-post outcome, one could then allocate to each player i a segment of size $y\epsilon$ with probability $p_{i,y}$. Unfortunately, as the constraints of this linear program only guarantee that the entire resource is allocated in expectation, derandomizing is not guaranteed to produce a lottery over ex-post feasible outcomes. For example, consider the following two-agent instance: agent 1 has value 1 if she receives half of the item, and agent 2 has value 1 if she receives the entire item. Allocating to the first agent half the item with probability 1 and to the second agent the entire item with probability 1/2 is feasible for this linear program (LP). This solution to the LP, of course, cannot be derandomized.

[4]For simplicity, we assume that $1/\epsilon$ is integer throughout this section.

5.2 Corrected Attempt: A Flow-Based Approach

In this section we construct an alternative linear program which randomizes over explicitly feasible ex-post outcomes, thus correcting the issues of the previous LP. We start by describing our approach for the single item case and then describe how to extend the approach to multiple items.

We aim to find a lottery over $1/\epsilon$ pieces, which we treat as indivisible. This time, however, we select an arbitrary ordering over the n agents and construct a directed graph with n columns of $1/\epsilon+1$ vertices each. We add a directed edge from vertex u in the i-th column and j-th row to each vertex v in the $i+1$-th column and the $j+y$-th row for all $y \in \{0, 1, \ldots, 1/\epsilon - j\}$. We also have a source and sink vertex connected to all vertices in the first and the last columns, respectively (see Figure 4). This construction guarantees that a path from the source to the sink vertex corresponds to a feasible allocation of the indivisible pieces of the good to the agents. The residual amounts of the good given to each agent in our arbitrary order is determined by the edges in the path as follows. Suppose the path contains an edge connecting vertex u in the i-th column and j-th row to vertex v in the $i+1$-th column and k-th row; this corresponds to allocating a fraction $\epsilon \cdot (k - j)$ of the item to agent $i + 1$. An edge from the source to vertex u in the first column and j-th row corresponds to allocating a fraction of $\epsilon \cdot j$ of the item to agent 1.

We can then assign to each edge from i, j to $i+1, j+y$ a variable $p_{i,j,y} \in [0, 1]$ corresponding to the probability that a $j \cdot \epsilon$ fraction was allocated up to (and including) agent i and an additional $y \cdot \epsilon$ fraction was given to agent $i + 1$, giving her a value of $f_{i+1}(y \cdot \epsilon)$. A Pareto efficient lottery can then be found by maximizing a linear objective (e.g., the total value of all agents) subject to flow preservation constraints through this graph. To ensure ex-ante envy-freeness, the desired fairness constraint can be encoded in the linear program. Overall, we find an approximately ex-ante Pareto optimal EF lottery which is a randomization over ex-post feasible outcomes (since every path is ex-post feasible).

Although we have only described how to construct the flow graph for a single item, this flow graph approach is easily extensible to multiple items when agents have additive valuations across items. One begins by constructing a separate flow graph for each item. Then, by connecting the sink vertex of the m'-th item graph to the source vertex of the $m'+1$-th item graph for all $m' < m$, we can then obtain an approximately ex-ante Pareto efficient lottery for any number of agents and items in polynomial time by optimizing the linear program over the set of constraints implied by this "chained" flow graph.

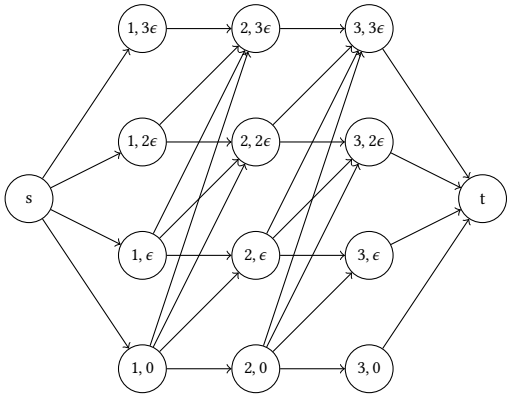

Figure 4: Example flow graph for $\epsilon = 1/3$ and three agents. A vertex v labeled "$i, y\epsilon$" indicates that y pieces of the resource have been consumed along the path from s to v.

We present the linear program for a single item below. Let $\mathcal{P} = \{2, 3, \ldots, n\} \times \{0, 1, 2, \ldots, 1/\epsilon\}$. The objective is again to maximize the total value of the agents. The first set of constraints impose flow preservation and the second constraint requires that a unit of flow is pushed from the source. The third set of constraints encodes ex-ante envy-freeness as agent i must weakly prefer her own lottery to that of i':

$$\begin{aligned}
&\text{maximize } \textstyle\sum_{i=1}^{n} \sum_{j=0}^{1/\epsilon} \sum_{y=j}^{1/\epsilon} p_{i,j,y} f_i((y-j)\cdot\epsilon) \\
&\text{subject to } \textstyle\sum_{k=0}^{1/\epsilon} p_{i-1,k,j-k} = \sum_{\ell=0}^{1/\epsilon} p_{i,j,\ell}, \forall (i,j) \in \mathcal{P} \\
&\qquad \textstyle\sum_{j=0}^{1/\epsilon} p_{0,0,j} = 1 \\
&\textstyle\sum_{j=0}^{1/\epsilon} \sum_{y=j}^{1/\epsilon} p_{i,j,y} f_i((y-j)\epsilon) \geq \sum_{j=0}^{1/\epsilon} \sum_{y=j}^{1/\epsilon} p_{i',j,y} f_i((y-j)\epsilon), \\
&\qquad\qquad\qquad\qquad\qquad\qquad \forall i, i' \in \mathcal{N}
\end{aligned}$$

Theorem 5.1. *The linear program above outputs an ex-ante ϵ-Pareto optimal EF lottery in polynomial time using $O\left(\frac{mn}{\epsilon^2}\right)$ queries for any $\epsilon < \frac{1}{mn}$.*

Proof. First note that ex-ante envy-freeness is implied directly from the constraints of the program and that the lottery implied by the variables is a randomization over ex-post feasible outcomes by the discussion above. Let C denote the maximum Lipschitz constant across all pairs of agents and resources. To demonstrate that the lottery is ex-ante ϵ-Pareto optimal using at most $\frac{Cmn}{\epsilon^2}$ queries, first observe that the lottery which gives each agent all the items with probability $1/n$ is a feasible solution to our program and is consistent with our discretization. On the other hand, as the objective function is optimizing the sum of ex-ante utilities (subject to the constraints), the linear program outputs a lottery which is ex-ante Pareto optimal with respect to the discretization. Therefore, the linear program must give at least one agent utility greater than or equal to $1/n$. The only reason that the solution for our program would not be ex-ante Pareto efficient with respect to the full valuation functions would be if there were jumps in the valuation functions between our queries. However, since all agents have Lipschitz valuation functions with Lipschitz constant at most C, in any ex-post outcome, if we were to give every agent an additional piece of size $\frac{\epsilon^2}{C}$ of every item, the utility of an agent would increase by at most $m\epsilon^2$. For an agent with utility $v \geq 1/n$, we have $v \cdot (1+\epsilon) \geq v + \frac{\epsilon}{n} > v + m\epsilon^2$. So, there is no lottery that improves her utility by a factor $1+\epsilon$, even with respect to the exact valuation functions. Finally, since there are a polynomial number of variables and constraints the program runs in polynomial time. □

We also note that our algorithm for finding ex-ante approximate Pareto efficient EF lotteries can be extended to agents with *non-additive*, bounded gradient valuation functions for a constant number of items. The procedure is largely the same, except we ask $\frac{1}{\epsilon^m}$ value queries for all possible combinations of ϵ sized pieces, e.g., for two items $(0,0), (0,\epsilon), (0,2\epsilon), \ldots, (0,1), (\epsilon,0), (\epsilon,\epsilon), \ldots, (1,1)$. We then appropriately modify the flow graph to include vertices representing all possible tuples for each agent and direct edges accordingly. However, this procedure only works for a constant number of items as the number of queries then depends *exponentially* on the number of items.

Since we discretize the resources into $1/\epsilon$ pieces each and treat these pieces as indivisible items, one could consider using the approach of Budish et al. [11] to find a lottery over ex-post feasible outcomes. However, our valuation functions over the items can be more general than their approach can support since we allow complementarity within a single resource (e.g., by allowing for convex valuation functions). On the other hand, their approach allows for more expressive constraint structures over the feasible allocations of goods to agents. We believe that investigating the degree to which the two approaches can be combined is an interesting question for future work.

6 OBSERVATIONS AND FUTURE WORK

In this section, we provide some auxiliary results in our model and suggest several interesting related open questions. An interesting first question to consider would be to tighten the gap between the upper and lower bounds on the number of queries required to compute an ex-ante ϵ-Pareto efficient EF lottery for a constant number of players. Do $o\left(\frac{1}{\epsilon^2}\right)$ queries suffice? In the case of general m and n, do $o\left(\frac{mn}{\epsilon^2}\right)$ suffice?

While our lower bound suggests that exact ex-ante Pareto efficient lotteries are unattainable using a bounded number of queries, if we simplify our objective from ex-ante Pareto efficiency to ex-post Pareto efficiency then we may find an ex-ante envy-free lottery which is *exactly* ex-post Pareto efficient with respect to the set of all outcomes. We note that our algorithm closely resembles the well-known random serial dictatorship mechanism for one-sided matching markets of Abdulkadiroğlu and Sönmez [1].

Theorem 6.1. *There exists an algorithm that produces an ex-ante envy-free lottery that is exactly ex-post Pareto efficient among the set of all outcomes using a polynomial number of queries.*

Proof. Consider an arbitrary set of n agents $\mathcal{N}$. We uniformly and randomly permute the agents and label them $1, 2, \ldots, n$. We ask agent 1 a value query $\textsc{Value}(f_{1k}, 1)$ for each item $k \in \mathcal{M}$. In

doing so, we determine her value v_{1k} for receiving each good k entirely. We then ask her a cut query $\textsc{Cut}(f_{1k}, v_{1k})$ for each item $k \in \mathcal{M}$, thereby determining how much of each good she needs to receive to obtain her "full value". We continue by allocating to her as much of each item as the corresponding cut query returns. For any remaining pieces of each item, we repeat this process for each successive agent in the ordering. As valuation functions are additive across items, it is easy to verify that this is ex-post Pareto efficient; removing any amount of any of the goods from agent i must decrease her value. This lottery is also ex-ante envy-free, since every agent has the same probability of appearing at any index in the random ordering, and at every point in the algorithm each consecutive agent receives a favorite bundle of goods among the remaining resources. Therefore, no agent envies the lottery of any other agent. □

On the other hand, while ex-post Pareto efficiency is easier to achieve than ex-ante Pareto efficiency, ex-post envy-freeness is a more stringent requirement than ex-ante envy-freeness. For a lottery to be ex-post envy-free, it must be that all outcomes in its support are ex-post envy-free. One may then ask the same questions we explore in this paper through the context of the class of ex-post envy-free lotteries. For instance, can one obtain an ex-ante ϵ-Pareto efficient lottery among the set of all ex-post envy-free lotteries using only a polynomial number of queries?

One can also imagine asking a similar question of finding ex-ante Pareto optimal lotteries among the set of all *ex-ante proportional* lotteries. Ex-ante proportionality is another standard notion of fairness which requires all agents to receive expected utility at least $1/n$. By changing the third set of constraints in our linear program in Section 5 to encode proportionality, one directly obtains the same guarantee of Theorem 5.1, i.e., one obtains an approximately ex-ante Pareto optimal proportional lottery. However, perhaps even more excitingly, one could change the welfare-maximization objective and instead maximize the *leximin* utility by repeatedly maximizing the minimum value (as in, e.g., Kurokawa et al. [20]) and *drop the fairness constraint altogether*. Since the equiprobable lottery is in the space of feasible lotteries, the leximin solution will necessarily give all agents expected utility at least $1/n$. Thus, the lottery would be approximately ex-ante Pareto optimal among *all* lotteries and satisfy ex-ante proportionality!

We view the question of determining if one can find a lottery which is approximately ex-ante Pareto optimal among all lotteries and which is also ex-ante envy-free to be a very interesting and challenging question. In fact, Brams et al. [8] point to the related question of finding a tractable algorithm to compute welfare-maximizing envy-free and Pareto efficient allocations (among the set of all allocations) in the standard cake-cutting model even in the seemingly quite simple case of known piecewise constant valuations as their "most important, and presumably quite challenging, open problem". More broadly, investigating the extent to which different fairness notions change the shape of the ex-ante Pareto frontier is an interesting line of future study.

7 CONCLUSION

In this paper, we propose a new model for allocating homogeneous divisible goods and we show that it is simple in natural ways – envy-freeness is easy to achieve – yet highly complex in others – many descriptive, non-trivial value functions can be captured. We provide an algorithm for finding approximately ex-ante Pareto optimal EF lotteries and complement this result with a lower bound on the query complexity of doing so. Notably, while our algorithm uses only value queries, our lower bound applies to any algorithm using either cut and/or value queries. We also provide an algorithm for finding exactly ex-post Pareto optimal EF lotteries which uses both cut and value queries.

Throughout this paper, we assume that valuation functions are Lipschitz. Therefore, determining if approximate Pareto optimality can be achieved when the Lipschitz assumption is removed would be a first step in extending our results. Perhaps most importantly, removing the assumption that agents are additive across resources is a major open line of further work. We have a way of converting our approach for an arbitrary number of additive items to a computationally efficient method of finding approximately ex-ante Pareto efficient EF lotteries for general valuations if there is a constant number of items. On the other hand, extending this result to arbitrarily many items may require very different techniques.

ACKNOWLEDGMENTS

The second author was supported in part by an NSF CAREER award CCF-2047907 and NSF grant CCF-2008280. The third author was supported in part by a Google Research Scholar Award. The fourth author was supported in part by NSF grants CCF-2008280 and CCF-1755955. Part of this work was done while the last three authors were visiting the Simons Institute for the Theory of Computing.

REFERENCES

[1] Atila Abdulkadiroğlu and Tayfun Sönmez. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 66, 3 (1998), 689–701.
[2] Georgios Amanatidis, George Christodoulou, John Fearnley, Evangelos Markakis, Christos-Alexandros Psomas, and Eftychia Vakaliou. 2018. An improved envy-free cake cutting protocol for four agents. In *Proceedings of the 11th International Symposium on Algorithmic Game Theory (SAGT) (Lecture Notes in Computer Science, Vol. 11059)*. 87–99.
[3] Haris Aziz and Simon Mackenzie. 2016. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 416–427.
[4] Haris Aziz and Simon Mackenzie. 2016. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*. 454–464.
[5] Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. 2012. Optimal proportional cake cutting with connected pieces. In *Twenty-sixth AAAI Conference on Artificial Intelligence*.
[6] Garrett Birkhoff. 1946. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A* 5 (1946), 147–151.
[7] Anna Bogomolnaia and Hervé Moulin. 2001. A new solution to the random assignment problem. *Journal of Economic Theory* 100, 2 (2001), 295–328.
[8] Steven Brams, Michal Feldman, John Lai, Jamie Morgenstern, and Ariel Procaccia. 2012. On maxsum fair cake divisions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26.
[9] S. Brams and A. Taylor. 1996. *Fair Division - From Cake-Cutting to Dispute Resolution*. Cambridge University Press.
[10] Steven J Brams and Alan D Taylor. 1995. An envy-free cake division protocol. *The American Mathematical Monthly* 102, 1 (1995), 9–18.
[11] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *American Economic Review* 103, 2 (2013), 585–623.
[12] Jan Buermann, Enrico H Gerding, and Baharak Rastegari. 2020. Fair allocation of resources with uncertain availability. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 204–212.
[13] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2012. The efficiency of fair division. *Theory of Computing Systems*

50, 4 (2012), 589–610.
[14] Yuga J Cohler, John K Lai, David C Parkes, and Ariel D Procaccia. 2011. Optimal envy-free cake cutting. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*. 626–631.
[15] Lester E Dubins and Edwin H Spanier. 1961. How to cut a cake fairly. *The American Mathematical Monthly* 68, 1P1 (1961), 1–17.
[16] Uriel Feige and Moshe Tennenholtz. 2014. On fair division of a homogeneous good. *Games and Economic Behavior* 87 (2014), 305–321.
[17] Eric J Friedman, Vasilis Gkatzelis, Christos-Alexandros Psomas, and Scott Shenker. 2019. Fair and efficient memory sharing: Confronting free riders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1965–1972.
[18] Nicole Immorlica, Brendan Lucier, E. Glen Weyl, and Joshua Mollner. 2017. Approximate Efficiency in Matching Markets. In *Web and Internet Economics - 13th International Conference, WINE 2017, Bangalore, India, December 17-20, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10660)*, Nikhil R. Devanur and Pinyan Lu (Eds.). Springer, 252–265.
[19] David Kurokawa, John K. Lai, and Ariel D. Procaccia. 2013. How to Cut a Cake before the Party Ends. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)* (Bellevue, Washington) *(AAAI'13)*. AAAI Press, 555–561.
[20] David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. 2018. Leximin Allocations in the Real World. *ACM Trans. Econ. Comput.* 6, 3–4, Article 11 (Oct. 2018), 24 pages. https://doi.org/10.1145/3274641
[21] Ariel D Procaccia. 2009. Thou shalt covet thy neighbor's cake. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*.
[22] J Hans Reijnierse and Jos AM Potters. 1998. On finding an envy-free Pareto-optimal division. *Mathematical Programming* 83, 1-3 (1998), 291–311.
[23] J. M. Robertson and W. Webb. 1998. *Cake-Cutting Algorithms - Be Fair If You Can*. CRC Press.
[24] Günther Ruhe and Bernd Fruhwirth. 1990. ε-optimality for bicriteria programs and its application to minimum cost flows. *Computing* 44, 1 (1990), 21–34.
[25] John Von Neumann. 2016. 1. A Certain Zero-sum Two-person Game Equivalent to the Optimal Assignment Problem. In *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 5–12.
[26] David Zeng and Alexandros Psomas. 2020. Fairness-Efficiency Tradeoffs in Dynamic Fair Division. In *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, Péter Biró, Jason Hartline, Michael Ostrovsky, and Ariel D. Procaccia (Eds.). ACM, 911–912.