On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources

Umang Bhaskar[1], AR Sricharan[2], and Rohit Vaish[3]

[1]Tata Institute of Fundamental Research, umang@tifr.res.in
[2]Chennai Mathematical Institute, arsricharan@cmi.ac.in
[3]Tata Institute of Fundamental Research, rohit.vaish@tifr.res.in

Abstract

We study fair allocation of undesirable indivisible items (or *chores*) and make three contributions: First, we show that determining the existence of an envy-free allocation is NP-complete even when agents have *binary additive* valuations. Second, we provide a polynomial-time algorithm for computing an allocation that satisfies envy-freeness up to one chore (EF1) under *monotone* valuations, correcting a existing proof of the same claim in the literature. A straightforward modification of our algorithm can be used to compute an EF1 allocation for *doubly monotone* instances (wherein each agent can partition the set of items into objective goods and objective chores). Our third and most important result applies to a *mixed resources* model consisting of indivisible chores and a divisible, desirable heterogenous resource (metaphorically, a cake). We show that there always exists an allocation that satisfies envy-freeness for mixed resources (EFM) in this setting, complementing an analogous recent result of Bei et al. (AAAI 2020). We also show a similar result in the flipped setting consisting of indivisible goods and a divisible "bad" cake.

1 Introduction

The question of how to fairly divide a set of resources among agents is of central importance in economics, mathematics, computer science, and political science. Such problems arise in a wide variety of real-world settings such as border settlements, assigning credit among contributing individuals, rent division, and, most relevant to the present times, in the distribution of vaccines and essential medical supplies [PSÜY20]. The theoretical study of fair resource allocation—or *fair division*—has classically focused on *divisible* resources (such as land or clean water), most prominently in the *cake-cutting* literature [BT96, RW98, Pro15].[1] A well-established concept of fairness in this setup is *envy-freeness* [Fol67] which stipulates that no agent prefers the share of any other agent to its own. An envy-free division of a cake is known to exist under general settings [Str80, Alo87, Su99, AM16], and for a wide range of utility functions, such an allocation can also be efficiently computed [CLPP11, KLP13, AY14, BR20].

By contrast, an envy-free solution can fail to exist when the resources are discrete or *indivisible*; important examples include the assignment of course seats at universities [OSB10, BCKO17] and the allocation of public housing units [BCH$^+$20]. This has motivated the formulation of relaxations such as *envy-freeness up to one good* (EF1) which requires that pairwise envy can be eliminated by removing some good from the envied bundle [LMMS04, Bud11]. The EF1 notion has enjoyed a rare combination of theoretical as well as practical success: On the theoretical side, there exist efficient algorithms for computing an EF1 allocation under general, monotone valuations [LMMS04]. At the same time, EF1 has also found impressive practical appeal on the popular fair division website *Spliddit* [GP15] and in course allocation applications [Bud11, BCKO17].

[1]Here, cake is a metaphor for a heterogenous resource that can be fractionally allocated.

Our focus in this work is on fair allocation of indivisible resources that are negatively valued or undesirable (also known as *chores*). The chore division problem, introduced by Gardner [Gar78], models scenarios such as distribution of household tasks (e.g., cleaning, cooking, etc.) or the allocation of responsibilities for controlling carbon emissions among countries [Tra02]. For indivisible chores, too, an envy-free allocation could fail to exist, and one of our contributions is to show that determining the existence of such outcomes is NP-complete even under highly restricted settings (Theorem 1). This negative result prompts us to explore the corresponding relaxation of *envy-freeness up to one chore*, also denoted by EF1.[2]

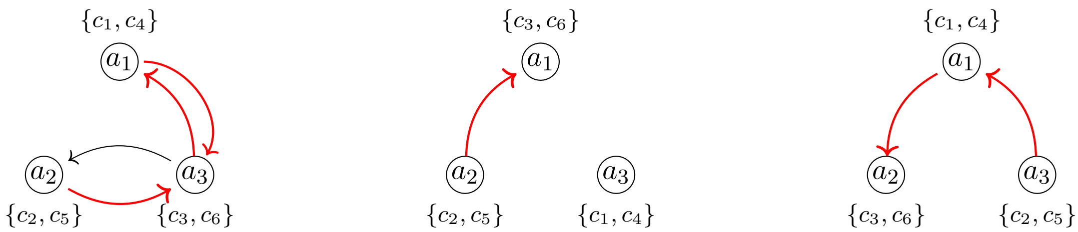
On first glance, the chore division problem might appear to be the 'opposite' of the goods problem, and therefore intuitively, one might expect natural adaptation of algorithms designed to compute EF1 for goods to also work for chores. This, however, turns out to not be the case, as we discuss below.

Goods vs chores: Let us consider the well-known *envy-cycle elimination* algorithm of Lipton et al. [LMMS04] for computing an EF1 allocation of indivisible goods. Briefly, the algorithm works by iteratively assigning goods to an agent that is not envied by anyone else. The existence of such an agent is guaranteed by means of resolving cycles in the underlying envy graph.[3] When adapted to the chores problem, the algorithm assigns a chore to a "non-envious" agent that has no outgoing edge in the envy graph. Interestingly, contrary to an existing claim in the literature [ACIW18], we observe that this algorithm could fail to find an EF1 allocation even when agents have additive valuations.

Example 1 (Envy-cycle elimination algorithm fails EF1 for additive chores). Consider the following instance with six chores $c_1, \dots, c_6$ and three agents a_1, a_2, a_3 with additive valuations:

	c_1	c_2	c_3	c_4	c_5	c_6
a_1	-1	-4	-2	-3	0	-1
a_2	-2	-1	-2	-2	-3	-1
a_3	-1	-3	-1	-1	-3	-10

Suppose the algorithm considers the chores in the increasing order of their indices (i.e., c_1, then c_2, and so on), and breaks ties among agents in favor of a_1, then a_2, and then a_3. It is easy to verify that no directed cycles appear at any intermediate step during the execution of the algorithm on the above instance. Thus, the resulting allocation, say A, is given by $A_1 = \{c_1, c_4\}$, $A_2 = \{c_2, c_5\}$, and $A_3 = \{c_3, c_6\}$ (shown as circled entries in the above table). Notice that A is EF1 and its envy graph is as shown in Figure 1a.



(a) *Envy graph of A.* **(b)** *Envy graph of X.* **(c)** *Envy graph of Y.*

Figure 1: *Envy graphs of various allocations in Example 1. The red edges denote the most envied bundle.*

Each node in the envy graph of A has an outgoing edge (Figure 1a). Therefore, if the algorithm were to allocate another chore after this, it would have to make a choice between resolving one of two envy cycles, namely $\{a_1, a_3\}$ and $\{a_2, a_3\}$. Let X and Y denote the allocations obtained by resolving the cycles $\{a_1, a_3\}$ and $\{a_2, a_3\}$, respectively (the corresponding envy graphs are shown in Figures 1b and 1c). Notice that although both envy graphs are *acyclic* (and thus admit a "sink" agent), only the allocation X satisfies EF1; in particular, the pair $\{a_1, a_3\}$ violates EF1 for Y. □

[2]EF1 for chores entails that any pairwise envy can be addressed by removing some chore from the envious agent's bundle

[3]The *envy graph* of an allocation is a directed graph whose vertices correspond to the agents and there is an edge (i, j) if agent i envies agent j.

The above example highlights an important contrast between indivisible goods and chores: For indivisible goods, resolving arbitrary envy cycles (until the envy graph becomes acyclic) is known to preserve EF1. However, for indivisible chores, the choice of *which* envy cycle is resolved matters.

A key insight of our work is that there always exists a specific envy cycle—the *top-trading envy cycle*—that can be resolved to compute an EF1 allocation of chores. Like Lipton et al. [LMMS04], our result also holds for *monotone* valuations (Theorem 2). Furthermore, a simple modification of this algorithm computes an EF1 allocation for *doubly monotone* instances (Theorem 3), which are instances where each agent can partition the items into 'goods' and 'chores', i.e., items with non-negative and negative marginal utility, respectively, for the agent [ACIW18].[4]

Motivated by this positive observation, we study a *mixed* model consisting of both divisible as well as indivisible resources. This is a natural model for settings where agents are paid to carry out a set of tasks; here the "indivisible" part comprises of the tasks and the "divisible" part corresponds to the monetary payments. Although the use of payments in fair allocation of indivisible resources has been explored in several works [Mas87, ADG91, Ara95, Su99, Kli00, MPR02, HRS02, HS19, Azi20, BDN+20, CI20], the most general formulation of a model with mixed resources is due to Bei et al. [BLL+20], who considered the fair division of a divisible *heterogenous* resource (i.e., a cake) and a set of indivisible goods.

Generalizing the set of resources calls for revising the fairness benchmark—note that while exact envy-free still remains out of reach in the mixed model, EF1 can be "too permissive" when only the divisible resource is present. Bei et al. [BLL+20] remedy this by proposing a fairness concept called *envy-freeness for mixed goods* (EFM) for indivisible goods and divisible cake, which evaluates fairness with respect to EF1 if the envied bundle only contains indivisible goods, but switches to exact envy-freeness if any amount of divisible resource is also present. Interestingly, they show that an EFM allocation always exists for a mixed instance when agents have additive preferences (across indivisible items and across resource types).

Our main contribution is to establish an analogous existence result in a model with *indivisible chores* and *a divisible cake* (Theorem 4). Specifically, we consider an appropriate extension of the EFM notion wherein agents who only own indivisible resources (i.e., chores) experience bounded envy towards others, while those in possession of any amount of the divisible resource (i.e., cake) are not envied by anyone else. We show that the aforementioned modification of EFM, which we refer to by the umbrella term *envy-freeness for a mixed resource* but continue to denote by EFM, always exists for any number of agents when the valuation function is additive over the indivisible items. Furthermore, we show a similar result for the flipped setting consisting of *indivisible goods* and a *divisible negatively-valued resource* (Corollary 1).

It is relevant to note that although we study a closely related model to that of Bei et al. [BLL+20], the techniques used by the two sets of results are quite different. For the indivisible goods and divisible cake model, Bei et al. [BLL+20] start with an EF1 allocation of the indivisible resources computed via the envy-cycle elimination algorithm [LMMS04]. At each subsequent step, they maintain a partial EFM allocation by making envy-free assignments of the cake among a maximal set of agents (an *addable set*) who (a) do not envy each other and (b) whose bundles are strictly less valued to any agent outside the set. Just like in the standard setting with only indivisible goods [LMMS04], the existence of such a set is guaranteed by the absence of envy-cycles.

By contrast, when the resources involve indivisible chores, we encounter several new challenges. First, we cannot first allocate the entire cake and then run the top-trading variant of the envy-cycle algorithm for the indivisible chores as this cannot, in general, prevent a fixed agent from being envied throughout the algorithm. Second, we also cannot first allocate the indivisible chores using our top-trading-style algorithm and then allocate the cake, since there may not be any *sources* in the envy-graph at the end of chores allocation (recall that the top-trading algorithm does not resolve *all* envy cycles).

To address these challenges, we turn to the classical round-robin algorithm, and define a variant of it that decides the ordering of agents in each round *based on the envy relations so far.* We call this

[4] This class has also been referred to as *itemwise monotone* in the literature [CL20].

the *augmented round-robin* algorithm, and show that under additive valuations, the *generalized envy-graph* of the resulting allocation (i.e., an envy-graph with both envy as well as equality edges) does not contain any generalized envy cycles (i.e., a cycle with at least one envy edge) (Lemma 6). Not only does this provide another way of computing an EF1 allocation of additive indivisible chores, but it also allows us to allocate the entire divisible cake first and still return an EFM allocation.

Our Contributions

1. We show that determining the existence of an envy-free allocation in a chores instances is (strongly) NP-complete even when agents have *binary* valuations, i.e., when for all agents $i \in [n]$ and items $j \in [m]$, $v_{i,j} \in \{-1, 0\}$ (Theorem 1).[5] This observation complements a known (weak) NP-completeness result for the same problem when agents have identical (but not necessarily binary) valuations via a reduction from Partition problem.

2. When the fairness goal is relaxed to envy-freeness up to one chore (EF1), we establish efficient computation for *monotone chores* instances (Theorem 2), and *doubly monotone* instances with goods and chores (Theorem 3).

3. With a further restriction to additive valuations, we show the existence of an allocation that satisfies the stronger fairness guarantee called *envy-freeness up to a mixed item* (EFM) for a mixed instance consisting of indivisible chores and a divisible cake (Theorem 4). Additionally, an easy modification of this algorithm also establishes a similar guarantee for the complement setting consisting of indivisible goods and a divisible bad cake (Corollary 1).

2 Related Work

As mentioned previously, fair division has been classically studied for *divisible* resources. For a *heterogenous*, desirable resource (i.e., a cake), the existence of envy-free solutions is known under mild assumptions [Str80, Alo87, Su99, AM16]. In addition, efficient algorithms are also known for restricted preferences [CLPP11, KLP13, AY14, BR20] and for computing ε-envy-free divisions [Pro15]. For an undesirable heterogenous resource (a "bad" cake), too, the existence of an envy-free division is known [PS09], along with a discrete and bounded procedure for finding such a division [DFHY18]. The case of non-monotone or "mixed" cake (i.e., a real-valued divisible heterogenous resource) has gained attention recently, and the existence of envy-free outcomes has been shown for specific values of the number of agents parameter [SH18, MZ19, AK19, AK20]. To the best of our knowledge, the existence question remains open for an arbitrary number of agents.

The special case of divisible *homogenous* resources has been extensively studied. For goods-only instances, the celebrated Eisenberg-Gale convex program establishes efficient computation of a competitive equilibrium [EG59]. Bogomolnaia et al. [BMSY19, BMSY17] study competitive equilibria of chores-only and "mixed manna" instances (goods and chores together). They show that the space of competitive utility profiles can be exponentially large, and discuss some barriers to the applicability of standard computational techniques in this setting. Subsequent work has provided systematic procedures for computing competitive equilibria for a broad class of utilities [CGMM21], special-case tractability results for chores-only [BS19] and mixed manna instances [GM20, CGMM21], and also considered optimization over the space of competitive utility profiles [CGMM20].

Turning to the *indivisible* setting, we note that the sweeping result of Lipton et al. [LMMS04] on EF1 for indivisible *goods* has inspired considerable work on establishing stronger existence and computation guarantees in conjunction with other well-studied economic properties [CKM+19, BKV18a, BKV18b, FSVX19, BCIZ20, CGM20, AMN20, FSV20]. The case of *indivisible chores* has been similarly well studied for a variety of solution concepts such as maximin fair share [ACL19, ARSW17, ALW19,

[5]The analogous problem for indivisible goods with binary valuations is known to be NP-complete [AGMW15, HSV+20].

HL19], equitability [FSVX20, Ale20b], competitive equilibria with general incomes [SH20], and envy-freeness [Ale18, BS19, FSV20]. In particular, Brânzei and Sandomirskiy [BS19] show that two distinct relaxations of EF1 for chores—denoted by EF_1^1 and Prop1—can be simultaneously achieved in conjunction with Pareto optimality. Freeman et al. [FSV20] strengthen this result by showing that there always exists a randomized allocation satisfying these properties ex post alongside ex-ante group fairness [CFSV19].

Aziz et al. [ACIW18, ACIW19] study a model containing both indivisible goods and chores. They show that a variant of the classical round-robin algorithm computes an EF1 allocation[6] under additive utilities, and also claim that a variant of the envy-cycle elimination algorithm [LMMS04] returns such allocations for doubly monotone instances (we revisit the latter claim in Example 1). Other fairness notions such as approximate proportionality [ACIW18, ACIW19, AMS20], maximin fair share [KMT20], approximate jealously-freeness [Ale20b], and weaker versions of EF1 [FSV20] have also been studied in this model.

Bérczi et al. [BBKB$^+$20] formalize variants of EF1 and its strengthening *envy-freeness up to any item* (EFX) for non-monotone instances with indivisible items. They show that for the special case of two agents, an EFX allocation may not exist even under identical valuations,[7] but an EF1 allocation always exists even when the two agents are not necessarily identical. They also prove the existence of an EFX allocation for chores-only instances with identical valuations. Aleksandrov [Ale20a] observes that EF1 and Pareto optimality can be incompatible for two agents in the non-monotone setting. Several other special-case existence results for EF1 and EFX (and their variants thereof) are known, such as for an arbitrary number of agents with identical valuations [CL20, AR20], or agents with boolean $\{0, +1\}$ and negative boolean $\{-1, 0\}$ valuations [BBKB$^+$20], or ternary valuations (i.e., $v_{i,j} \in \{-\alpha_i, 0, +\beta_i\}$) [AR20, AW20].

Finally, we note that the model with *mixed resources* comprising of both indivisible and (heterogenous) divisible parts has been recently formalized by Bei et al. [BLL$^+$20], although, a special case of their model where the divisible resource is homogenous and desirable (e.g., money) has been extensively studied [Mas87, ADG91, Ara95, Su99, Kli00, MPR02, HRS02, HS19, Azi20, BDN$^+$20, CI20]. Bei et al. [BLL$^+$20] showed that when there are indivisible goods and a divisible cake, an allocation satisfying *envy-freness for mixed goods* (EFM) always exists. Subsequent work considers the maximin fairness notion in the mixed model [BLLW20].

3 Preliminaries

In this paper, we deal with two kinds of instances. One with purely indivisible items, and the other with a mixture of divisible and indivisible resources. We will start with the preliminaries for instances with purely indivisible items.

Instances with Only Indivisible Resources

Problem Instance An *instance* $\langle N, M, \mathcal{V}\rangle$ of the fair division problem is defined by a set N of $n \in \mathbb{N}$ *agents*, a set M of $m \in \mathbb{N}$ *indivisible items*, and a *valuation profile* $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ that specifies the preferences of every agent $i \in N$ over each subset of the items in M via a *valuation function* $v_i : 2^M \to \mathbb{R}$.

Additive valuations: We say that the valuation functions are *additive* if the value of any subset of items is equal to the sum of the values of individual items in the set, i.e., for any agent $i \in N$ and any set of items $S \subseteq M$, $v_i(S) := \sum_{j\in S} v_i(\{j\})$, where we assume that $v_i(\emptyset) = 0$. For simplicity, we will write $v_i(j)$ or $v_{i,j}$ to denote $v_i(\{j\})$.

[6]When both goods and chores are present, Aziz et al. [ACIW18, ACIW19] define *envy-freeness up to an item* (EF1) as envy bounded by the removal of some good from the envied bundle or some chore from the envious agent's bundle.

[7]The non-existence of EFX for non-monotone instances has also been noted in [Ale20a, Ale20b].

Marginal valuations: For any agent $i \in N$ and any set of items $S \subseteq M$, the *marginal valuation* of the set $T \subseteq M \setminus S$ is given by $v_i(T|S) := v_i(S \cup T) - v_i(S)$. When the set T is a singleton (say $T = \{j\}$), we will write $v_i(j|S)$ instead of $v_i(\{j\}|S)$ for simplicity.

Monotone instances: We say that the valuation functions are *monotone non-decreasing* if for any sets $S \subseteq T \subseteq M$ and any agent $i \in N$, we have $v_i(T) \geq v_i(S)$, and *monotone non-increasing* if for any sets $S \subseteq T \subseteq M$ and any agent $i \in N$, we have $v_i(S) \geq v_i(T)$. The valuation functions are said to be *monotone* if all valuations are either monotone non-increasing or all are monotone non-decreasing. A *monotone instance* is one where all agents have monotone valuations.

Goods and chores: Given an agent $i \in N$ and an item $j \in M$, we say that j is a *good* for agent i if for every subset $S \subseteq M \setminus \{j\}$, $v_i(j|S) \geq 0$. We say that j is a *chore* for agent i if for every subset $S \subseteq M \setminus \{j\}$, $v_i(j|S) \leq 0$, with one of the inequalities strict.

Doubly monotone instances: We say that an instance is *doubly monotone* if each agent i can partition the items as $M = G_i \uplus C_i$, where G_i contains all of her goods, and C_i contains all of her chores.

Allocation An *allocation* $A := (A_1, \ldots, A_n)$ is an n-partition of the set of items M, where $A_i \subseteq M$ is the *bundle* allocated to the agent i (note that A_i can be empty). Given an allocation A, the *value* of agent $i \in M$ for the bundle A_i is $v_i(A_i)$. An allocation is said to be *complete* if it assigns all items in M, and is called *partial* otherwise.

Envy Graph The *envy graph* G_A of an allocation A is a directed graph on the vertex set N, with a directed edge from agent i to agent k if $v_i(A_k) > v_i(A_i)$, i.e. i prefers A_k over A_i in the allocation A.

Top-trading Graph The *top-trading graph* T_A of an allocation A is a directed graph on the vertex set N, with a directed edge from agent i to agent k if $v_i(A_k) = \max_{j \in N} v_i(A_j)$ and $v_i(A_k) > v_i(A_i)$, i.e. A_k is the most preferred bundle for agent i in the allocation A, and she prefers A_k over her own bundle.

Generalized Envy Graph The *generalized envy graph* E_A of an allocation A is a directed graph on the vertex set N, with a directed edge from agent i to agent k if $v_i(A_k) \geq v_i(A_i)$. If $v_i(A_k) = v_i(A_i)$, then we refer to the edge (i, k) as an *equality edge*, otherwise we call it an *envy edge*. A *generalized envy cycle* in this graph is a cycle C that contains at least one envy edge.

Cycle-swapped allocation Given an allocation A and a directed cycle C in an envy graph, the *cycle-swapped allocation* A^C is obtained by reallocating bundles backwards along the cycle. For each agent i in the cycle, define i^+ as the agent that she is pointing to in C. Then,

$$A_i^C = \begin{cases} A_i & \text{if } i \notin C, \\ A_{i^+} & \text{if } i \in C. \end{cases}$$

Envy-freeness and its relaxations An allocation A is said to be

- *envy-free* (EF) if for every pair of agents $i, k \in N$, we have $v_i(A_i) \geq v_i(A_k)$;
- *envy-free up to one item* (EF1) if for every pair of agents $i, k \in N$ such that $A_i \cup A_k \neq \emptyset$, there exists an item $j \in A_i \cup A_k$ such that $v_i(A_i \setminus \{j\}) \geq v_i(A_k \setminus \{j\})$, and
- *envy-free up to any item* (EFX) if for every pair of agents $i, k \in N$ such that $A_i \cup A_k \neq \emptyset$, for every item $j \in A_i \cup A_k$ such that $v_i(A_i \setminus \{j\}) \geq v_k(A_k \setminus \{j\})$.

The notions of EF, EF1, and EFX were proposed in the context of goods allocation by Foley [Fol67], Budish [Bud11] and Caragiannis et al. [CKM+19] respectively. An earlier work by Lipton et al. [LMMS04] studied a weaker approximation of envy-freeness for goods, but their algorithm is known to compute an EF1 allocation. The extensions of these notions to a model with goods and chores together have been studied before [ACIW19, BBKB+20, Ale20a].

Pareto Optimality An allocation A is said to Pareto dominate (or Pareto improve over) another allocation B if for every agent $i \in N$, $v_i(B_i) \geq v_i(A_i)$, and for some agent $k \in N$, $v_k(B_k) > v_k(A_k)$. A Pareto optimal allocation is one that is not Pareto dominated by any other allocation.

Instances with Divisible And Indivisible Resources

We will now describe the setting with *mixed resources* consisting of both divisible and indivisible parts. This model was recently studied by Bei et al. [BLL+20], who introduced the notion of *envy-freeness for mixed goods* (EFM) in the context of a model consisting of indivisible goods and a divisible cake (i.e., a desirable heterogenous resource). We generalize this notion to a setting with both goods and chores.

Mixed Instance A mixed instance $\langle N, M, \mathcal{V}, \mathcal{C}, \mathcal{F} \rangle$ is defined by a set of n agents, m indivisible items, a valuation profile $\mathcal{V}$ (over the indivisible resource), a divisible resource $\mathcal{C}$ represented by the interval $[0, 1]$, and a family of *density functions* over the divisible resource. The valuations for the indivisible items are as described above. For the divisible resource, each agent has a *density function* $f_i : [0, 1] \rightarrow \mathbb{R}$ such that for any measurable subset $S \subset [0, 1]$, agent i values it at $v_i(S) \coloneqq \int_S f_i(x)dx$. When the density function is non-negative for every agent (i.e., for all $i \in N$, $f_i : [0, 1] \rightarrow \mathbb{R}_{\geq 0}$), we will call the divisible resource a “cake”, and for non-positive densities (i.e., for all $i \in N$, $f_i : [0, 1] \rightarrow \mathbb{R}_{\leq 0}$), we will use the term “bad cake”. Finally, we will use the term “mixed cake” to talk about real-valued density functions.

Allocation An *allocation* $A \coloneqq (A_1, \ldots, A_n)$ is given by $A_i = M_i \cup C_i$, where $(M_1, \ldots, M_n)$ is an n-partition of the set of indivisible items M, and $(C_1, \ldots, C_n)$ is an n-partition of the divisible resource $\mathcal{C} = [0, 1]$. where A_i is the *bundle* allocated to the agent i (note that A_i can be empty). Given an allocation A, the *utility* of agent $i \in N$ for the bundle A_i is $v_i(A_i) \coloneqq v_i(M_i) + v_i(C_i)$.

Envy-freeness for mixed resources We will now discuss the notion of envy-freeness for mixed resources that was formalized by Bei et al. [BLL+20] in the context of indivisible goods and divisible cake, and also describe its extensions to related settings where the indivisible part consists of chores and/or the divisible part is bad cake. All four definitions below are based on the following idea: Any agent who owns cake should not be envied, any agent who owns bad cake should not envy anyone else, and subject to these conditions, any pairwise envy can be bounded in the same way as EF1.

- *Indivisible Goods and Divisible Cake* [BLL+20]: An allocation A is EFM if for every pair of agents $i, k \in N$,
  - if agent k owns a non-zero amount of divisible cake (i.e., if $C_k \neq \emptyset$), then agent i does not envy agent k (i.e., $v_i(A_i) \geq v_i(A_k)$), otherwise
  - $v_i(A_i) \geq v_i(A_k \setminus \{g\})$ for some indivisible good $g \in A_k$ (assuming $A_k \neq \emptyset$).
- *Indivisible Chores and Divisible Cake*: An allocation A is EFM if for every pair of agents $i, k \in N$,
  - if agent k owns a non-zero amount of divisible cake (i.e., if $C_k \neq \emptyset$), then agent i does not envy agent k (i.e., $v_i(A_i) \geq v_i(A_k)$), otherwise
  - $v_i(A_i \setminus \{c\}) \geq v_i(A_k)$ for some indivisible chore $c \in A_i$ (assuming $A_i \neq \emptyset$).
- *Indivisible Goods and Divisible Bad Cake*: An allocation A is EFM if for every pair of agents $i, k \in N$,
  - if agent i owns a non-zero amount of divisible bad cake (i.e., if $C_i \neq \emptyset$), then agent i does not envy agent k (i.e., $v_i(A_i) \geq v_i(A_k)$), otherwise
  - $v_i(A_i) \geq v_i(A_k \setminus \{g\})$ for some indivisible good $g \in A_k$ (assuming $A_k \neq \emptyset$).
- *Indivisible Chores and Divisible Bad Cake*: An allocation A is EFM if for every pair of agents $i, k \in N$,

– if agent i owns a non-zero amount of divisible bad cake (i.e., if $C_i \neq \emptyset$), then agent i does not envy agent k (i.e., $v_i(A_i) \geq v_i(A_k)$), otherwise

– $v_i(A_i \setminus \{g\}) \geq v_i(A_k)$ for some indivisible chore $c \in A_i$ (assuming $A_i \neq \emptyset$).

Query Model We describe the *Robertson-Webb (RW) query model* [RW98], a standard complexity model for queries in cake cutting. We are allowed two types of queries about the cake:

- $\text{EVAL}_i(x, y)$ returns the value $v_i([x, y])$ of agent i for the piece $[x, y]$.
- $\text{CUT}_i(x, \alpha)$ returns a point $y \geq x$ on the cake such that $v_i([x, y]) = \alpha$.

We assume that each query in the RW model takes unit time.

4 Envy-Freeness for Binary Valued Chores

Our first result shows that determining the existence of an envy-free allocation is NP-complete even when agents have binary valuations, i.e., when, for all agents $i \in N$ and items $j \in M$, $v_{i,j} \in \{-1, 0\}$ (Theorem 1). It is relevant to note that without the binary valuations assumption, the problem is still known to be (weakly) NP-complete even for identical agents via a straightforward reduction from PARTITION. By contrast, our result establishes strong NP-completeness.

Theorem 1. *Determining whether a given chores instance admits an envy-free allocation is* NP-complete *even for binary utilities.*

Proof. Membership in NP follows from the fact that given an allocation, checking whether it is envy-free can be done in polynomial time.

To show NP-hardness, we will show a reduction from SET SPLITTING which is known to be NP-complete [GJ79] and asks the following question: Given a universe U and a family $\mathcal{F}$ of subsets of U, does there exist a partition of U into two sets U_1, U_2 such that each member of $\mathcal{F}$ is *split* by this partition, i.e., no member of $\mathcal{F}$ is completely contained in either U_1 or U_2?

Construction of the reduced instance: Let $q := |U|$ and $r := |\mathcal{F}|$ denote the cardinality of the universe U and the set family $\mathcal{F}$, respectively. Let $r' := \max\{q, r\}$. We will find it convenient to refer to the universe as a set of 'vertices', the members of the set family $\mathcal{F}$ as a set of 'hyperedges', and the membership in U_1 or U_2 as each vertex being 'colored' 1 or 2.

We will construct a fair division instance with $m = r' + q$ chores and $n = r' + 2$ agents. The set of chores consists of r' *dummy* chores $D_1, \ldots, D_{r'}$ and q *vertex* chores $V_1, \ldots, V_q$. The set of agents consists of r' *edge* agents $e_1, \ldots, e_{r'}$, and two *color* agents c_1, c_2. When $r' = r$ (i.e., $r \geq q$), each edge agent should be interpreted as corresponding to a hyperedge, and otherwise if $r < r'$, then we will interpret the first r edge agents $e_1, \ldots, e_r$ as corresponding to the hyperedges while each of the remaining edge agents $e_{r+1}, \ldots, e_{r'}$ will be considered as an "imaginary hyperedge" that is adjacent to the entire set of vertices (and therefore does not impose any additional constraints on the coloring problem).

Preferences: The valuations of the agents are specified as follows: Each dummy chore is valued at -1 by all (edge and color) agents. Each vertex chore V_j is valued at -1 by those edge agents e_i whose corresponding hyperedge $E_i \in E$ is adjacent to the vertex $v_j \in V$, and at 0 by all other edge agents. The color agents value all vertex chores at 0. This completes the construction of the reduced instance. We will now argue the equivalence of the solutions.

($\Rightarrow$) Suppose there exists a partition of the universe U that splits all member of $\mathcal{F}$ (equivalently, a feasible 2-coloring of the corresponding hypergraph such that each hyperedge sees both colors). Then, an envy-free allocation can be constructed as follows: The r' dummy chores are evenly distributed among the r' edge agents. In addition, if the vertex v_j is assigned the color $\ell \in \{1, 2\}$, then the vertex chore V_j is assigned to the color agent c_ℓ.

The aforementioned allocation is feasible as it assigns each item to exactly one agent. Furthermore, it is also envy-free for the following reason: Each color agent only receives vertex chores and has utility 0, and therefore it does not envy anyone else. The utility of each edge agent e_i is -1 because of the dummy chore assigned to it. However, e_i does not envy any other edge agent e_ℓ since the latter is also assigned a dummy chore. Furthermore, e_i also does not envy any of the color agents since, by the coloring condition, each of them receives at least one chore that is valued at -1 by e_i.

($\Leftarrow$) Now suppose that there exists an envy-free allocation, say A. Then, it must be that none of the color agents receive a dummy chore. This is because assigning a dummy chore to a color agent c_ℓ would give it a utility of -1, and in order to compensate for the envy, it would be necessary to assign *every* other agent at least one chore that c_ℓ values at -1. This, however, is impossible since the number of agents other than c_ℓ is $r' + 2$, which strictly exceeds the number of chores that c_ℓ values at -1, namely r'. Thus, all dummy chores must be allocated among the edge agents.

We will now show that no edge agent receives more than one dummy chore under the allocation A. Suppose, for contradiction, that the edge agent e_i is assigned two or more dummy chores. Then, due to envy-freeness, every other agent must get at least two chores that e_i values at -1. There are $r' + 2$ agents in total excluding e_i, which necessitates that there must be at least $2r' + 4$ chores valued at -1 by e_i. However, the actual number of chores valued at -1 by e_i that are available for allocation is at most $(r' - 2) + q$, which, by the choice of r', is strictly less than $2r' + 4$, leading to a contradiction. Thus, each edge agent receives at most one dummy chore, and since the number of edge agents equals that of dummy chores, we get that the r' dummy chores are, in fact, evenly distributed among the r' edge agents.

Since each edge agent e_i receives a dummy chore, in order to compensate for the envy the allocation A must assign each color agent at least one vertex chore that e_i values at -1. The desired coloring for the hypergraph (equivalently, the desired partition of the universe U) can now be naturally inferred; in particular, the elements of U corresponding to the vertex chores whose assignment is not forced by the aforementioned remark can be put in an arbitrary partition. This completes the proof of Theorem 1. $\square$

5 EF1 For Doubly Monotone Instances

In light of the intractability result in the previous section, we will now explore whether one can achieve approximate envy-freeness (specifically, EF1) for indivisible chores. To that end, we note that the well-known round-robin algorithm (wherein, in each round, agents take turns in picking their favorite available chore) computes an EF1 allocation when agents have additive valuations. In the following, we will provide an algorithm for computing an EF1 allocation for the much more general class of *monotone valuations*. Thus, our result establishes the analogue of the result of Lipton et al. [LMMS04] from the goods-only model for indivisible chores.

5.1 An Algorithm for Monotone Chores

As previously mentioned, the algorithm of Lipton et al. [LMMS04] computes an EF1 allocation for indivisible goods under monotone valuations. Recall that the algorithm works by assigning, at each step, an unassigned good to an agent who is not envied by anyone else (such an agent is a "source" agent in the underlying envy graph). The existence of such an agent is guaranteed by resolving arbitrary envy-cycles in the envy graph until it becomes acyclic.

To design an EF1 algorithm for indivisible chores, prior work [ACIW18, ACIW19] has proposed the following natural adaptation of this algorithm (Algorithm 1): Instead of a "source" agent, an unassigned chore is now allocated to a "sink" (i.e., non-envious) agent in the envy graph. The existence of such an agent is once again guaranteed by means of resolving envy cycles. However, the key point to note—as illustrated in Example 1 in the Introduction—is that resolving *arbitrary* envy cycles could destroy the EF1 property. The reason has to do with the fact that when evaluating EF1 for chores, a chore is removed from the envious agent's bundle. In the envy cycle resolution step, if a cycle is

ALGORITHM 1: Naïve envy-cycle elimination algorithm

Input: An instance $\langle N, M, \mathcal{V} \rangle$ with non-increasing valuations
Output: An allocation A

Initialize $A \leftarrow (\emptyset, \emptyset, \ldots, \emptyset)$
for $c \in M$ **do**
  Choose a sink i in the envy graph G_A
  Update $A_i \leftarrow A_i \cup \{c\}$
  while *G_A contains a directed cycle C* **do**
    $A \leftarrow A^C$
return A

ALGORITHM 2: Top-trading cycle elimination algorithm

Input: An instance $\langle N, M, \mathcal{V} \rangle$ with non-increasing valuations
Output: An allocation A

Initialize $A \leftarrow (\emptyset, \emptyset, \ldots, \emptyset)$
for $c \in M$ **do**
  if *there is no sink in G_A* **then**
    $C \leftarrow$ any cycle in T_A ▷ if G_A has no sink, then T_A must have a cycle
    $A \leftarrow A^C$
  Choose a sink k in the graph G_A
  Update $A_k \leftarrow A_k \cup \{c\}$
return A

chosen without caution, then it is possible for an agent to acquire a bundle that, although strictly more preferable, contains no chore that is large enough to compensate for the envy on its own.

To address this gap, we propose to resolve a specific envy cycle that we call the *top-trading envy cycle.*[8] Specifically, given a partial allocation A, we consider a subgraph of the envy-graph G_A that we call the *top-trading graph* T_A whose vertices denote the agents, and an edge (i, k) denotes that agent i's (weakly) most preferred bundle is A_k.

It is easy to observe that if the envy-graph does not have a sink, then the top-trading graph T_A has a cycle (Lemma 2). Thus, resolving top-trading cycles (instead of arbitrary envy cycles) also guarantees the existence of a sink agent in the envy graph. More importantly, though, resolving a top-trading cycle *preserves* EF1. Indeed, every agent involved in the top-trading exchange receives its most preferred bundle after the swap, and therefore does not envy anyone else in the next round. The resulting algorithm is presented in Algorithm 2.

Theorem 2. *For a monotone instance with indivisible chores, Algorithm 2 returns an* EF1 *allocation.*

We defer the proof of Theorem 2 to the next subsection, where we generalize the top-trading cycle elimination algorithm to *doubly monotone* instances containing both indivisible goods as well as indivisible chores.

5.2 An Algorithm for Doubly Monotone Instances

For a doubly monotone instance with indivisible items, we now give an algorithm (Algorithm 3) that returns an EF1 allocation. The algorithm runs in two phases. The first phase is for all the items that are a good for at least one agent. For these items, we run the envy-cycle elimination algorithm of Lipton et al. [LMMS04] on the subgraph of agents who consider the item a good. In the second phase, we

[8]The nomenclature is inspired from the celebrated top-trading cycles algorithm [SS74] for finding a core-stable allocation that involves cyclic swaps of the most preferred objects.

ALGORITHM 3: An EF1 algorithm for doubly monotone indivisible instances

Input: An instance $\langle N, M, \mathcal{V}\rangle$ with doubly monotone utilities and indivisible items
Output: An allocation A

1 **for** *each agent* $i \in N$ **do**
2     $G_i \leftarrow \{o \in M \mid v_i(o|S) \geq 0$ for all $S \subseteq M \setminus \{o\}\}$ ▷ G_i contains all items that agent i considers a good
3     $C_i \leftarrow M \setminus G_i$ ▷ C_i contains all items that agent i considers a chore
4     $A_i \leftarrow \emptyset$

// Goods Phase
5 **for** *each item* $g \in \cup_i G_i$ **do**
6     $V^g = \{i \in N \mid g \in G_i\}$ ▷ V^g contains all agents for whom g is a good
7     $G_A^g =$ the envy graph G_A restricted to the vertices V^g
8   Choose a source k in the graph G_A^g
9   Update $A_k \leftarrow A_k \cup \{g\}$
10   **while** G_A *contains a directed cycle* C **do**
11         $A \leftarrow A^C$

// Chores Phase
12 **for** *each item* $c \in \cap_i C_i$ **do**
13   **if** *there is no sink in* G_A **then**
14         $C \leftarrow$ any cycle in T_A ▷ if G_A has no sink, then T_A must have a cycle
15         $A \leftarrow A^C$
16   Choose a sink k in the graph G_A
17   Update $A_k \leftarrow A_k \cup \{c\}$
18 **return** A

allocate items that are chores to everybody by running the top-trading cycle elimination algorithm. For a monotone chores-only instance, we recover Algorithm 2 as a special case of Algorithm 3.

Theorem 3. *For a doubly monotone instance with indivisible items, Algorithm 3 returns an* EF1 *allocation.*

Define a time step as a stage of the algorithm where either an item gets added to a bundle, or a cycle gets resolved. We maintain the invariant that at every step of the algorithm, the allocation remains EF1. Briefly, during the goods phase, any envy created from i to j can always be removed by dropping a good $g \in A_j$. In the chores phase, any new envy created by adding a chore can be removed by dropping the newly added chore. If we resolve top-trading cycles, then no one inside the cycle envies anyone outside it since they now have their most preferred bundle. For any agent i outside the cycle, any envy can be removed by either removing a chore from i or a good from the envied bundle, since i's allocation is unchanged and the bundles remain unbroken.

Lemma 1. *After every step of the goods phase, the partial allocation remains* EF1. *Further, the goods phase terminates in polynomial time.*

Proof. The proof closely follows the arguments of Lipton et al. [LMMS04]. For completeness, we present a self-contained proof below.

Clearly the empty allocation at the beginning is EF1. Suppose before time step t, our allocation A is EF1 (i.e., any envy from agent i to agent j can be eliminated by removing an item from A_j). Denote the allocation after time step t by A'. We will argue that A' is EF1, and any envy from agent i to agent j can be eliminated by removing an item from A'_j.

Suppose at time step t, we had a line 8/9 step. Every time we reach line 5 and enter the loop with an item g, the graph G_A is acyclic. This is because either this is the first time we reach line 5, in which case it is trivially true, or we just reached line 5 after passing through lines 10/11, which eliminates all envy-cycles. Thus, the subgraph G_A^g is acyclic as well, where G_A^g is the graph G_A restricted to the agents for whom g is a good.

Then after time t, our allocation A' will be $A'_k = A_k \cup \{g\}$, and $A'_j = A_j$ for all $j \neq k$, where k is a source in G^g_A. Pick two agents i and j such that i envies j in A'. If i did not envy j in A, then clearly $j = k$ and $i \in V^g$. In this case, removing g from A_k removes i's envy as well. Suppose i envied j in A as well, and the envy was eliminated by removing g' from A_j. If $j = k$ then $i \notin V^g$ since k was a source in G^g_A. Then removing g' eliminates the envy in A' as well, since $v_i(A_k \cup \{g\} \setminus \{g'\}) \leq v_i(A_k \setminus \{g'\})$. If $j \neq k$, then since j's bundle remains the same and $v_i(A'_i) \geq v_i(A_i)$, the envy can again be eliminated by removing g' from A'_j.

Suppose at time t we had a line 10/11 step. Let A be the allocation before time t, C be the cycle along which the swap happens, and $A' = A^C$ the allocation obtained by swapping backwards along the circle. Pick two agents i and j such that i envies j in A'. Let i' and j' be the agents such that $A'_i = A_{i'}$ and $A'_j = A_{j'}$. Since $v_i(A'_i) \geq v_i(A_i)$, i envied j' in the allocation A before the swap. Suppose this envy was eliminated by removing g' from $A_{j'}$. Then $v_i(A'_i) \geq v_i(A_i) \geq v_i(A_{j'} \setminus \{g'\})$, and thus removing g' from A'_j eliminates the envy in A'.

To show that the algorithm terminates in polynomial time, we show that the while loop in lines 10/11 will be executed at most a polynomial number of times for each item. Consider a single while loop for an item, where a cycle swap occurs on the cycle C. Since the bundles remain unbroken, all agents outside the cycle have the same outdegree in $G_{A'}$ as in G_A. An agent i inside the cycle has strictly lesser outdegree in $G_{A'}$ compared to G_A, since the (i, i^+) edge in G_A does not translate into a (i, i) edge in $G_{A'}$ (since i gets i^+'s bundle). Thus the number of envy edges goes down by $|C|$ during each cycle swap, and the while loop terminates in polynomial time. □

We now consider the chores phase of the algorithm. For lines 13-14, we show that if there is no sink in G_A, then there is a cycle in T_A.

Lemma 2. *If there are no sinks in G_A, then T_A contains a cycle. Further, if A' is the allocation obtained by resolving a cycle in T_A, then there is at least one sink in $G_{A'}$.*

Proof. Since G_A has no sinks, every vertex in G_A has outdegree at least one. Thus for all agents i, $i \notin \arg\max_k v_i(A_k)$. So even in the top-trading graph T_A, each vertex has outdegree at least one. We start at an arbitrary agent and follow an outgoing edge from each successive agent. This gives us a cycle in T_A. Note that each agent points to its favorite bundle in T_A. Thus after resolving a cycle in T_A, all agents who participated in the cycle-swap now have their most preferred bundle in A' and do not envy any other agent. These agents are sinks in the graph $G_{A'}$. □

Lemma 3. *At every step of the chores phase, the allocation remains* EF1*, and the chores phase terminates in polynomial time.*

Proof. By Lemma 1, the allocation at the beginning of the chores phase is EF1. Suppose before time step t, our allocation A was EF1, and the allocation after time step t is A'. We show that A' is EF1 as well.

Suppose at time step t, we had a line 16/17 step. Every time we reach line 16, we have a sink in the graph G_A. This is because either there was already a sink in the graph when we entered the loop at line 12, or we passed through lines 13-15, in which case all the agents who were a part of the cycle C do not envy anyone after the cycle-swap, and are sinks in the next line 16/17 step.

Then after time t, our allocation A' will be $A'_k = A_k \cup \{c\}$, and $A'_j = A_j$ for all $j \neq k$, where k is a sink in G_A. Pick two agents i and j such that i envies j in A'. If i did not envy j in A, then clearly $i = k$. In this case, removing c from A_i removes i's envy. Suppose i envied j in A as well, and the envy was eliminated by removing $o \in A_i \cup A_j$. Then $i \neq k$ since k was a sink in the graph G_A, and so $v_i(A_i) = v_i(A'_i)$. If $o \in A_i$, then $v_i(A'_i \setminus \{o\}) \geq v_i(A_j) \geq v_i(A'_j)$. If $o \in A_j$, then $v_i(A'_i) \geq v_i(A_j \setminus \{o\}) \geq v_i(A_j \cup \{c\} \setminus \{o\})$, since c is a chore for all agents.

Suppose at time t we had a line 14/15 step. Let A be the allocation before time t, C be the cycle along which the swap happens, and $A' = A^C$ the allocation obtained by swapping backwards along the circle. Pick two agents i and j such that i envies j in A'. Since every agent in the cycle obtains their favourite bundle, $i \notin C$. Thus $A_i = A'_i$. Let j' be the agent such that $A'_j = A_{j'}$. Since $v_i(A'_i) = v_i(A_i)$, i envied j' before the swap which could be eliminated by removing $o \in A_i \cup A_{j'}$. If $o \in A_i$, then $v_i(A'_i \setminus \{o\}) \geq v_i(A'_j)$. If $o \in A_{j'}$, then $v_i(A'_i) \geq v_i(A'_j \setminus \{o\})$. Thus removing $o \in A_i \cup A_{j'}$ eliminates the envy in A'.

Thus after every step of the chores phase, the allocation remains EF1. By Lemma 2, finding a cycle in T_A takes only polynomial time. Since the while loop executes only once for each chore, the chores phase terminates in polynomial time. □

The proof of Theorem 3 follows immediately, since by Lemma 3 the allocation at the end of the chores phase is EF1. Thus Algorithm 3 returns an EF1 allocation for a doubly monotone instance. Specialized to the case of monotone non-increasing valuations, we obtain Theorem 2 as a corollary.

6 EFM for Indivisible Chores and Divisible Cake

6.1 Augmented Round Robin Algorithm

While Algorithm 3 returns an EF1 allocation, there might be unresolved envy cycles even in the final allocation. A generalized envy cycle in the generalized envy graph E_A implies an obvious Pareto improvement that we may not be able to perform because it might destroy the EF1 property. Recall that a generalized envy cycle is a cycle in E_A that contains at least one envy edge. For a monotone instance with additive valuations and indivisible chores, we give an algorithm that returns an EF1 allocation that is generalized envy cycle free.

We work with the generalized envy graph E_A since working with the envy graph G_A is not sufficient to obtain EFM allocations [BLL+20], and we use Algorithm 4 as a subroutine in the next subsection to obtain EFM allocations. The problem with obtaining an EF1 allocation without envy cycles for chores was the fact that in Algorithm 3, one could not remove any envy cycle of their choice. For the case of additive chores, we present a modified version of the round robin algorithm where we maintain the invariant that after each round of the algorithm, the allocation is generalized envy cycle free. In each round, we move from the sink strongly connected components (referred to henceforth as components) all the way up to the source, and perform a maximum weight perfect matching in the bipartite graph $H_j = (S_j \cup M, E)$, with the vertices of the component S_j on one side and the remaining unallocated items on the other. The weight of an edge from agent i to chore c is its value $v_i(c)$ for the chore. We show that this returns an EF1 allocation without generalized envy cycles.

At the start of each round, we first find a topological sorting of the components (see Section 22.5 of [CLRS09]). Let ComponentToposort$(\cdot)$ be the subroutine that takes in a directed graph G and returns $S = (S_1, S_2, S_3, \dots S_\ell)$, the components of G in topological order. There are no new cycles created within a component after the round since that would contradict maximality of the matching, and there are no new cycles between components either because we allocated chores in reverse topological order. Thus the allocation is generalized envy cycle free, and properties of the round robin algorithm assure us that this allocation is EF1.

To make analysis easier, we assume that the number of items is a multiple of the number of agents. If not, we add virtual items that are valued at 0 by every agent, and remove them at the end of the algorithm. Note that this preserves the EF1 property.

We will show that ARRA$((\emptyset, \dots, \emptyset), N, M, \mathcal{V})$ returns an EF1 allocation without generalized envy cycles. Call each recursive call of the algorithm as a *round* of the algorithm. In each round, an agent receives exactly one item.

Lemma 4. *Let A and A' be the allocations at the start and end of a round respectively. Suppose $S = (S_1, S_2, \dots S_\ell)$ was the topological sorting of the components of E_A. Then $E_{A'}$ has no generalized envy cycles, and for every component S'_k in $E_{A'}$, $S'_k \subseteq S_j$ for some j.*

Proof. If $i > j$, then we show that there are no new edges created from an agent in S_i to an agent in S_j during the round. Since S_i comes after S_j in the topological sort, no agent in S_i has an envy or

ALGORITHM 4: Augmented Round Robin Algorithm

Input: $\langle A, N, M, \mathcal{V}\rangle$ where $A = (A_1, A_2, \ldots, A_n)$ is a partial allocation, N a set of agents, M a set of unallocated chores, and $\mathcal{V}$ a valuation function

Output: An allocation A

if $M = \emptyset$ **then**
    return A
else
    $S \leftarrow$ ComponentToposort(E_A) ▷ `Topological sorting of the components of` E_A
    $\ell = |S|$
    `// Go through the components in reverse topological order`
    for $j = \ell, \ell - 1, \ldots, 1$ **do**
        $H_j = (S_j \cup M, S_j \times M)$ ▷ `Weighted bipartite graph of agents and unallocated items`
        $w(i, c) = v_i(c)$ for all $i \in S_j, c \in M$ ▷ `weights for` H_j `are the value of the chore for the agent`
        $N \leftarrow$ maximum weight perfect matching in H_j
        for $i \in S_j$ **do**
            $A_i \leftarrow A_i \cup \{N(i)\}$
            $M \leftarrow M \setminus \{N(i)\}$
    return ARRA$(A, N, M, \mathcal{V})$

equality edge to any agent in S_j before the round. Thus every agent in S_i strongly prefers their bundle over any bundle in S_j. Note that since all agents in S_i pick their items in this round before any agent in S_j, they all prefer their new items weakly over any new item allocated to an agent in S_j. Thus if $i > j$, there are no new envy or equality edges created from S_i to S_j during the round. This also implies that there cannot be a component S'_k in $E_{A'}$ with vertices from different components of E_A, because then a new envy or equality edge should have been created from an agent in S_i to an agent in S_j with $i > j$.

We claim that there is no generalized envy cycle created inside S_j during the round. Let N be the maximum weight perfect matching in H_j, and let $N(i)$ be the item allocated to agent $i \in S_j$ during this round. Suppose there was a generalized envy cycle C created in S_j after adding the items $\{N(i) \mid i \in S_j\}$. Recall that i^+ is the agent in the cycle C that i points to. For all $i \in C$,

$$v_i(A_i) \geq v_i(A_{i^+})$$
$$v_i(A_{i^+} \cup \{N(i^+)\}) \geq v_i(A_i \cup \{N(i)\})$$

Putting both of these together, we get that $v_i(N(i^+)) \geq v_i(N(i))$. Since there is an envy edge (p, p^+) in the generalized envy cycle C, we get that at least one inequality is strict, $v_p(N(p^+)) > v_p(N(p))$. Then the matching M with $M(i) = N(i)$ if $i \in S_j \setminus C$ and $M(i) = N(i^+)$ if $i \in C$ is a perfect matching with higher weight than N, contradicting maximality of N.

Thus $E_{A'}$ has no generalized envy cycles, and for every component S'_k in $E_{A'}$, $S'_k \subseteq S_j$ for some j. □

Lemma 5. *Every agent weakly prefers the item she gets in round t over any item any agent gets in round $t + 1$.*

Proof. This is trivial to see, since if there is an item $c \in M$ that is unallocated that she strongly prefers over the item c' that she obtains in round t, then getting matched to c instead of c' gives a matching with higher weight than N, a contradiction. □

Lemma 6. *For an additive instance with indivisible chores, Algorithm 4 returns an* EF1 *allocation that contains no generalized envy cycles.*

Proof. From Lemma 4, we know that the allocation at the end of the last round has no generalized envy cycles. We now show that the allocation is EF1. Since m is a multiple of n (by adding 0 valued virtual goods if necessary), and since every agent obtains one item in each round, all agents have the same number of items at the end. Denote the number of items each agent has at the end by $\alpha = \frac{m}{n}$. Take any two agents i and j. Suppose i envies j in the allocation A obtained using Algorithm 4. Then we claim that if we remove the last item c_i^α allocated to i, $v_i(A_i \setminus \{c_i^\alpha\}) \geq v_i(A_j)$. For each round t, we know from Lemma 5 that $v_i(c_i^t) \geq v_i(c_j^{t+1})$. Thus

$$v_i(A_i \setminus \{c_i^\alpha\}) = \sum_{r=1}^{\alpha-1} v_i(c_i^r) \geq \sum_{r=1}^{\alpha-1} v_i(c_j^{r+1}) = v_i(A_j \setminus \{c_j^1\}) \geq v_i(A_j)$$

as required. Thus the allocation is EF1. □

6.2 An EFM Algorithm for Indivisible Chores and Divisible Cake

The fairness notion of EFM for a mixed instance $\langle N, M, \mathcal{V}, \mathcal{C}\rangle$ with additive indivisible goods and divisible cake was defined by Bei et al. [BLL+20]. EFM is an extension of EF1 where an envied agent should not have any divisible cake in their bundle. The algorithm of Bei et al. for finding an EFM allocation for additive indivisible goods and divisible cake is as follows: First obtain an EF1 allocation of the indivisible goods using the standard envy-cycle elimination algorithm on E_A. Thus there are no generalized envy cycles in E_A after allocating all the indivisible goods. Define a (source) addable set to be a maximal set of agents who cannot be reached by envy edges. Now find the largest prefix of the cake that can be allocated perfectly to the agents in the source addable set before some agent outside the source addable set has a new equality edge to an agent in the source addable set. Recall that a perfect allocation is an allocation of the cake such that each agent values all pieces of the allocation equally. Now keep resolving any generalized envy cycles that form, and perfectly allocate the cake until all the cake has been allocated.

This directly generalizes to the instance where the valuation function on the indivisible goods is monotone, since their algorithm does not rely on additivity of the valuation function for the indivisible items. For the case of monotone indivisible chores and divisible bad cake, we can work with the generalized top-trading graph as in Algorithm 3, and with sink addable sets instead of source addable sets to obtain an EFM allocation. For the case of monotone indivisible goods and divisible bad cake, we can run the envy-cycle elimination algorithm to obtain an EF1 allocation of the goods, then switch to the top-trading cycle algorithm for adding bad cake to the sink addable sets.

However when we have indivisible chores and divisible cake, this style of algorithm does not work. We cannot first allocate the cake then use the top-trading cycles algorithm for the indivisible chores (similar to Algorithm 3) since none of the envy graph algorithms can prevent an agent from being envied throughout the algorithm. We cannot allocate the indivisible chores first using the top-trading cycles algorithm then allocate the cake, since there may not be any sources in the graph at the end of chore allocation. Thus this case is interesting as it necessarily requires a technique different from the previous ones.

We give an algorithm to obtain an EFM allocation of indivisible chores and divisible cake when the valuation on the indivisible chores is additive. Since the definition of EFM is not affected by rescaling the valuations of individual agents, we assume without loss of generality that $v_i(\mathcal{C}) = 1$ for all i in the input instance. The idea for the algorithm is as follows: If there are agents who have net non-negative utility for the entirety of $M \cup \mathcal{C}$, then find the agent who has zero utility with all the chores and the least prefix $[0, x_i]$ of the cake. No one envies this agent once $M \cup [0, x_i]$ is allocated to her, and she in turn values all bundles at zero utility. Now an EF allocation of the remaining cake ensures that the entire allocation is EF, and so EFM as well.

Else, everyone has negative utility for $M \cup \mathcal{C}$. For each agent j, define S_j^t to be the set of his t favourite chores. Also define k_j to be the first index such that $v_j(S_j^{k_j} \cup \mathcal{C})$ becomes less than 0. We allocate $S_i^{k_i} \cup \mathcal{C}$ to the agent i who has the maximum value of k_j, and run Algorithm 4 on the remaining unallocated chores to obtain an EFM allocation.

ALGORITHM 5: Indivisible Chores and Divisible Cake Algorithm

Input: An instance $\langle N, M, \mathcal{V}, \mathcal{C}, \mathcal{F}\rangle$ with additive indivisible chores M and a divisible cake $\mathcal{C}$ such that $v_i(\mathcal{C}) = 1$ for all $i \in N$

Output: An allocation A

1 $A \leftarrow (\emptyset, \emptyset, \dots \emptyset)$

2 Order the agents such that $v_1(M) \geq v_2(M) \dots \geq v_n(M)$

3 **if** $v_1(M) \geq -1$ **then**

4 $\quad S \leftarrow \{j \in N \mid v_j(M) \geq -1\}$

5 $\quad x_j \leftarrow \textsc{cut}_j(0, -v_j(M))$ for all $j \in S$

6 $\quad i \in \arg\min_{j \in S} x_j$

7 $\quad A_i \leftarrow M \cup [0, x_i]$

8 $\quad B = \text{EFCut}([x_i, 1], N)$

9 $\quad A_j \leftarrow A_j \cup B_j$ for all $j \in N$

// Else $v_i(M) < 1$ for all agents i

10 **else**

11 $\quad S_j^t \leftarrow$ set of t favourite chores of agent j for all $j \in N$, and for all $t \in [m]$

12 $\quad k_j \leftarrow$ least index t such that $v_j(S_j^t) < -1$

13 $\quad i \in \arg\max_{j \in N} k_j$

14 $\quad A_i \leftarrow S_i^{k_i} \cup \mathcal{C}$

15 $\quad A = \text{ARRA}(A, N, M \setminus S_i^{k_i}, \mathcal{V})$

16 **return** A

In line 5, we use an RW cut query to find the prefix $[0, x_j]$ of the cake such that agent j values $M \cup [0, x_j]$ at 0. In line 8, B is an envy-free allocation of $[x_i, 1]$ to all the agents in N, where each agent obtains a contiguous piece of the cake [Su99]. Though Algorithm 4 was only stated for indivisible chores, we show that instantiating it with the mixed allocation A in line 15 satisfies the same round invariants mentioned in Section 6.1, giving us an EFM allocation.

Theorem 4. *For a mixed instance with additive indivisible chores and divisible cake, Algorithm 5 returns an* EFM *allocation.*

Since lines 3-9 and lines 10-15 are mutually exclusive, we can go through their analysis separately.

Lemma 7. *If $v_1(M) \geq -1$, then Algorithm 5 returns an* EF *allocation.*

Proof. Let $S = \{j \mid v_j(M) \geq -1\}$ be the set of all agents who value $M \cup \mathcal{C}$ non-negatively. For each agent $j \in S$, let $x_j = \inf\{x \mid v_j([0, x]) \geq -v_j(M)\}$ be the minimum prefix of cake required such that $v_j(M \cup [0, x_j]) = 0$. Note that this is the same x_j we obtain from our cut query in line 5. For our analysis, define $x_j = 1$ for all $j \notin S$. Note that if $j \notin S$, then $v_j(M) + v_j([0, 1]) < 0$. Since $x_i = \min_{j \in S} x_j = \min_{j \in N} x_j$, we have that for each $j \in N$,

$$v_j(M) + v_j([0, x_i]) \leq v_j(M) + v_j([0, x_j]) \leq 0$$

with equality for agent i. Let B be an EF allocation of $[x_i, 1]$. Then if we allocate $A_i = M \cup [0, x_i] \cup B_i$, and $A_j = B_j$ for all $j \neq i$, then by envy-freeness of B, we have for all $j \neq i$,

$$v_i(A_j) = v_i(B_j) \leq v_i(B_i) = v_i(A_i)$$
$$\text{and } v_j(A_i) \leq v_j(B_i) \leq v_j(B_j) = v_j(A_j)$$

showing that the allocation A is EF. ◻

Note that we never cut the cake in lines 11-15 of the algorithm. Since we can get a connected EF division of $[x_i, 1]$ in line 8, we can obtain an EFM allocation where everyone except agent i gets a connected piece while agent i gets two connected pieces. Interestingly, we only need one call to an EF oracle here while in the case of indivisible goods and divisible cake, the algorithm of Bei et al. [BLL+20] might need multiple queries to a perfect allocation oracle.

Lemma 8. *The allocation given in line 14 of Algorithm 5 is* EFM.

Proof. First we will show that no agent envies i. Suppose j envied i. Then $v_j(S_i^{k_i}) \geq -1$. Since S_j^t consists of j's t favourite chores, we have that $v_j(S_j^{k_i}) \geq v_j(S_i^{k_i}) \geq -1$. By definition of k_j and the property of k_i above, we have that $k_j > k_i$, contradicting the maximality of k_i.

Now we claim that i's envy for other agents is EFM. Suppose i envies j. Note that j does not have any cake. Since k_i was the least index such that $v_i(S_i^{k_i}) < -1$, we have that $v_i(S_i^{k_i-1}) \geq -1$. Thus if $\{c\} = S_i^{k_i} \setminus S_i^{k_i-1}$, then $v_i(S_i^{k_i} \cup \mathcal{C} \setminus \{c\}) \geq 0 = v_i(A_j)$, and so this envy is EFM. □

Lemma 9. *The allocation given in line 15 of Algorithm 5 is* EFM.

Proof. By Lemma 8, the allocation in line 14 is EFM. We initialize Algorithm 4 with $A_i = S_i^{k_i} \cup \mathcal{C}$ and $A_j = \emptyset$ for all $j \neq i$. Note that the corresponding graph E_A is generalized envy cycle free, since i envies all other agents j, no other agent j envies i, and all agents other than i value their bundles equally. Further, i's envy disappears if she drops the item $\{c\} = S_i^{k_i} \setminus S_i^{k_i-1}$. Agent i prefers her current bundle to any item left in $M \setminus S_i^{k_i}$, since $v_i(S_i^{k_i} \cup \mathcal{C}) \geq v_i(c) \geq v_i(c')$ for all items $c' \in M \setminus S_i^{k_i}$. Since every other agent j was initialized with $A_j = \emptyset$, they prefer their initial items to any item they receive in the later rounds trivially.

Since i is in a source component at the beginning of the algorithm, we can start our component topological sort from the component $\{i\}$ in each round. Thus agent i will be considered last in each round of the algorithm, and no other agent has any directed edges to i at any stage of the algorithm. Thus the final allocation is EFM. □

Putting together Lemma 7 and Lemma 9, we get that Algorithm 5 returns an EFM allocation.

For a mixed instance with additive indivisible goods and divisible bad cake, we can use an adaptation of Algorithm 5 to obtain an EFM allocation. Once again, we assume without loss of generality that $v_i(\mathcal{C}) = -1$ for all i. If all agents have net non-positive utility for the entirety of $M \cup \mathcal{C}$, then find the agent i who has zero utility with all the goods and the highest prefix $[0, x_i]$ of the cake. No one envies this agent once $M \cup [0, x_i]$ is allocated to her, and she in turn values all bundles at zero utility. Now an EF allocation of the remaining cake ensures that the entire allocation is EF, and so EFM as well. Else, some agents have positive utility for $M \cup \mathcal{C}$. For an agent j, define S_j^t to be the set of his t favourite goods. Also define k_j to be the first index such that $v_j(S_j^{k_j} \cup \mathcal{C})$ becomes higher than 0. We allocate $S_i^{k_i} \cup \mathcal{C}$ to the agent i who has the minimum value of k_j, and run an adaptation of Algorithm 4 for goods on the remaining unallocated goods to obtain an EFM allocation. Note that Lemma 5, which was crucial in proving that Algorithm 4 returns an EF1 allocation, did not depend on whether the items being allocated are goods or chores. While the agent with cake remained a source throughout the algorithm in our earlier setting, the agent with bad cake now remains a sink throughout the algorithm, which is what we need to ensure that our final allocation is EFM. This gives us the following corollary:

Corollary 1. *For a mixed instance with additive indivisible goods and divisible bad cake, there exists an* EFM *allocation.*

Acknowledgments

UB and RV acknowledge support from project no. RTI4001 of the Department of Atomic Energy, Government of India. RV also acknowledges support from Prof. R Narasimhan postdoctoral award.

References

[ACIW18] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair Allocation of Combinations of Indivisible Goods and Chores. *arXiv preprint arXiv:1807.10684 (version v3)*, 2018. (Cited on pages 2, 3, 5, and 9)

[ACIW19] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair Allocation of Indivisible Goods and Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 53–59, 2019. (Cited on pages 5, 6, and 9)

[ACL19] Haris Aziz, Hau Chan, and Bo Li. Weighted Maxmin Fair Share Allocation of Indivisible Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 46–52, 2019. (Cited on page 5)

[ADG91] Ahmet Alkan, Gabrielle Demange, and David Gale. Fair Allocation of Indivisible Goods and Criteria of Justice. *Econometrica: Journal of the Econometric Society*, pages 1023–1039, 1991. (Cited on pages 3 and 5)

[AGMW15] Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. Fair Assignment of Indivisible Objects under Ordinal Preferences. *Artificial Intelligence*, 227:71–92, 2015. (Cited on page 4)

[AK19] Sergey Avvakumov and Roman Karasev. Envy-Free Division Using Mapping Degree. *arXiv preprint arXiv:1907.11183*, 2019. (Cited on page 4)

[AK20] Sergey Avvakumov and Roman Karasev. Equipartition of a Segment. *arXiv preprint arXiv:2009.09862*, 2020. (Cited on page 4)

[Ale18] Martin Aleksandrov. Almost Envy Freeness and Welfare Efficiency in Fair Division with Goods or Bads. *arXiv preprint arXiv:1808.00422*, 2018. (Cited on page 5)

[Ale20a] Martin Aleksandrov. Envy-Freeness Up to One Item: Shall we Add or Remove Resources? *arXiv preprint arXiv:2006.11312*, 2020. (Cited on pages 5 and 6)

[Ale20b] Martin Aleksandrov. Jealousy-Freeness and Other Common Properties in Fair Division of Mixed Manna. *arXiv preprint arXiv:2004.11469*, 2020. (Cited on page 5)

[Alo87] Noga Alon. Splitting Necklaces. *Advances in Mathematics*, 63(3):247–253, 1987. (Cited on pages 1 and 4)

[ALW19] Haris Aziz, Bo Li, and Xiaowei Wu. Strategyproof and Approximately Maxmin Fair Share Allocation of Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 60–66, 2019. (Cited on page 5)

[AM16] Haris Aziz and Simon Mackenzie. A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 416–427. IEEE, 2016. (Cited on pages 1 and 4)

[AMN20] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. Multiple Birds with One Stone: Beating 1/2 for EFX and GMMS via Envy Cycle Elimination. *Theoretical Computer Science*, 841:94–109, 2020. (Cited on page 4)

[AMS20] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A Polynomial-Time Algorithm for Computing a Pareto Optimal and Almost Proportional Allocation. *Operations Research Letters*, 48(5):573–578, 2020. (Cited on page 5)

[AR20] Haris Aziz and Simon Rey. Almost Group Envy-Free Allocation of Indivisible Goods and Chores. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (forthcoming)*, 2020. (Cited on page 5)

[Ara95] Enriqueta Aragones. A Derivation of the Money Rawlsian Solution. *Social Choice and Welfare*, 12(3):267–276, 1995. (Cited on pages 3 and 5)

[ARSW17] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for Max-Min Share Fair Allocation of Indivisible Chores. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 335–341, 2017. (Cited on page 5)

[AW20] Martin Aleksandrov and Toby Walsh. Two Algorithms for Additive and Fair Division of Mixed Manna. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 3–17. Springer, 2020. (Cited on page 5)

[AY14] Haris Aziz and Chun Ye. Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations. In *International Conference on Web and Internet Economics*, pages 1–14. Springer, 2014. (Cited on pages 1 and 4)

[Azi20] Haris Aziz. Achieving Envy-freeness and Equitability with Monetary Transfers. *arXiv preprint arXiv:2003.08125*, 2020. (Cited on pages 3 and 5)

[BBKB+20] Kristóf Bérczi, Erika R Bérczi-Kovács, Endre Boros, Fekadu Tolessa Gedefa, Naoyuki Kamiyama, Telikepalli Kavitha, Yusuke Kobayashi, and Kazuhisa Makino. Envy-Free Relaxations for Goods, Chores, and Mixed Items. *arXiv preprint arXiv:2006.04428*, 2020. (Cited on pages 5 and 6)

[BCH+20] Nawal Benabbou, Mithun Chakraborty, Xuan-Vinh Ho, Jakub Sliwinski, and Yair Zick. The Price of Quota-based Diversity in Assignment Problems. *ACM Transactions on Economics and Computation*, 8(3):1–32, 2020. (Cited on page 1)

[BCIZ20] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. Finding Fair and Efficient Allocations When Valuations Don't Add Up. In *Proceedings of the Thirteenth International Symposium on Algorithmic Game Theory*, pages 32–46, 2020. (Cited on page 4)

[BCKO17] Eric Budish, Gérard P Cachon, Judd B Kessler, and Abraham Othman. Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation. *Operations Research*, 65(2):314–336, 2017. (Cited on page 1)

[BDN+20] Johannes Brustle, Jack Dippel, Vishnu V Narayan, Mashbat Suzuki, and Adrian Vetta. One Dollar Each Eliminates Envy. In *Proceedings of the Twenty-First ACM Conference on Economics and Computation*, pages 23–39, 2020. (Cited on pages 3 and 5)

[BKV18a] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding Fair and Efficient Allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 557–574, 2018. (Cited on page 4)

[BKV18b] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 2018 International Conference on Autonomous Agents and Multiagent Systems*, pages 7–13, 2018. (Cited on page 4)

[BLL+20] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair Division of Mixed Divisible and Indivisible Goods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1814–1821, 2020. (Cited on pages 3, 5, 7, 13, 15, and 17)

[BLLW20] Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. Maximin Fairness with Mixed Divisible and Indivisible Goods. *arXiv preprint arXiv:2002.05245*, 2020. (Cited on page 5)

[BMSY17] Anna Bogomolnaia, Herve Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. Competitive Division of a Mixed Manna. *Econometrica*, 85(6):1847–1871, 2017. (Cited on page 4)

[BMSY19] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaia. Dividing Bads under Additive Utilities. *Social Choice and Welfare*, 52(3):395–417, 2019. (Cited on page 4)

[BR20] Siddharth Barman and Nidhi Rathi. Fair Cake Division Under Monotone Likelihood Ratios. In *Proceedings of the Twenty-First ACM Conference on Economics and Computation*, pages 401–437, 2020. (Cited on pages 1 and 4)

[BS19] Simina Brânzei and Fedor Sandomirskiy. Algorithms for Competitive Division of Chores. *arXiv preprint arXiv:1907.01766*, 2019. (Cited on pages 4 and 5)

[BT96] Steven J Brams and Alan D Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996. (Cited on page 1)

[Bud11] Eric Budish. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011. (Cited on pages 1 and 6)

[CFSV19] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Wortman Vaughan. Group Fairness for the Allocation of Indivisible Goods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1853–1860, 2019. (Cited on page 5)

[CGM20] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. Fair and Efficient Allocations under Subadditive Valuations. *arXiv preprint arXiv:2005.06511*, 2020. (Cited on page 4)

[CGMM20] Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. Dividing Bads is Harder than Dividing Goods: On the Complexity of Fair and Efficient Division of Chores. *arXiv preprint arXiv:2008.00285*, 2020. (Cited on page 4)

[CGMM21] Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. Competitive Allocation of a Mixed Manna. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (forthcoming)*, 2021. (Cited on page 4)

[CI20] Ioannis Caragiannis and Stavros Ioannidis. Computing Envy-Freeable Allocations with Limited Subsidies. *arXiv preprint arXiv:2002.02789*, 2020. (Cited on pages 3 and 5)

[CKM+19] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation*, 7(3):12, 2019. (Cited on pages 4 and 6)

[CL20] Xingyu Chen and Zijie Liu. The Fairness of Leximin in Allocation of Indivisible Chores. *arXiv preprint arXiv:2005.04864*, 2020. (Cited on pages 3 and 5)

[CLPP11] Yuga J Cohler, John K Lai, David C Parkes, and Ariel D Procaccia. Optimal Envy-Free Cake Cutting. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 626–631, 2011. (Cited on pages 1 and 4)

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. (Cited on page 13)

[DFHY18] Sina Dehghani, Alireza Farhadi, MohammadTaghi HajiAghayi, and Hadi Yami. Envy-Free Chore Division For an Arbitrary Number of Agents. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2564–2583. SIAM, 2018. (Cited on page 4)

[EG59] Edmund Eisenberg and David Gale. Consensus of Subjective Probabilities: The Pari-Mutuel Method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959. (Cited on page 4)

[Fol67] Duncan Foley. Resource Allocation and the Public Sector. *Yale Economic Essays*, pages 45–98, 1967. (Cited on pages 1 and 6)

[FSV20] Rupert Freeman, Nisarg Shah, and Rohit Vaish. Best of Both Worlds: Ex-Ante and Ex-Post Fairness in Resource Allocation. In *Proceedings of the Twenty-First ACM Conference on Economics and Computation*, pages 21–22, 2020. (Cited on pages 4 and 5)

[FSVX19] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable Allocations of Indivisible Goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 280–286, 2019. (Cited on page 4)

[FSVX20] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable Allocations of Indivisible Chores. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 384–392, 2020. (Cited on page 5)

[Gar78] Martin Gardner. *aha! Insight*. W.H.Freeman and Company, 1978. (Cited on page 2)

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979. (Cited on page 8)

[GM20] Jugal Garg and Peter McGlaughlin. Computing Competitive Equilibria with Mixed Manna. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 420–428, 2020. (Cited on page 4)

[GP15] Jonathan Goldman and Ariel D Procaccia. Spliddit: Unleashing Fair Division Algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2015. (Cited on page 1)

[HL19] Xin Huang and Pinyan Lu. An Algorithmic Framework for Approximating Maximin Share Allocation of Chores. *arXiv preprint arXiv:1907.04505*, 2019. (Cited on page 5)

[HRS02] Claus-Jochen Haake, Matthias G Raith, and Francis Edward Su. Bidding for Envy-Freeness: A Procedural Approach to n-Player Fair-Division Problems. *Social Choice and Welfare*, 19(4):723–749, 2002. (Cited on pages 3 and 5)

[HS19] Daniel Halpern and Nisarg Shah. Fair Division with Subsidy. In *International Symposium on Algorithmic Game Theory*, pages 374–389. Springer, 2019. (Cited on pages 3 and 5)

[HSV+20] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, Hejun Wang, and Lirong Xia. Fair Division through Information Withholding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2014–2021, 2020. (Cited on page 4)

[Kli00] Flip Klijn. An Algorithm for Envy-Free Allocations in an Economy with Indivisible Objects and Money. *Social Choice and Welfare*, 17(2):201–215, 2000. (Cited on pages 3 and 5)

[KLP13] David Kurokawa, John K Lai, and Ariel D Procaccia. How to Cut a Cake Before the Party Ends. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 555–561, 2013. (Cited on pages 1 and 4)

[KMT20] Rucha Kulkarni, Ruta Mehta, and Setareh Taki. Approximating Maximin Shares with Mixed Manna. *arXiv preprint arXiv:2007.09133*, 2020. (Cited on page 5)

[LMMS04] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On Approximately Fair Allocations of Indivisible Goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131, 2004. (Cited on pages 1, 2, 3, 4, 5, 6, 9, 10, and 11)

[Mas87] Eric S Maskin. On the Fair Allocation of Indivisible Goods. In *Arrow and the Foundations of the Theory of Economic Policy*, pages 341–349. Springer, 1987. (Cited on pages 3 and 5)

[MPR02] Marc Meertens, Jos Potters, and Hans Reijnierse. Envy-Free and Pareto Efficient Allocations in Economies with Indivisible Goods and Money. *Mathematical Social Sciences*, 44(3):223–233, 2002. (Cited on pages 3 and 5)

[MZ19] Frédéric Meunier and Shira Zerbib. Envy-Free Cake Division Without Assuming the Players Prefer Nonempty Pieces. *Israel Journal of Mathematics*, 234(2):907–925, 2019. (Cited on page 4)

[OSB10] Abraham Othman, Tuomas Sandholm, and Eric Budish. Finding Approximate Competitive Equilibria: Efficient and Fair Course Allocation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 873–880, 2010. (Cited on page 1)

[Pro15] Ariel D Procaccia. Cake Cutting Algorithms. In *Handbook of Computational Social Choice, Chapter 13*. Citeseer, 2015. (Cited on pages 1 and 4)

[PS09] Elisha Peterson and Francis Edward Su. N-Person Envy-Free Chore Division. *arXiv preprint arXiv:0909.0303*, 2009. (Cited on page 4)

[PSÜY20] Parag A Pathak, Tayfun Sönmez, M Utku Ünver, and M Bumin Yenmez. Fair Allocation of Vaccines, Ventilators and Antiviral Treatments: Leaving No Ethical Value Behind in Health Care Rationing. *arXiv preprint arXiv:2008.00374*, 2020. (Cited on page 1)

[RW98] Jack Robertson and William Webb. *Cake-Cutting Algorithms: Be Fair If You Can*. CRC Press, 1998. (Cited on pages 1 and 8)

[SH18] Erel Segal-Halevi. Fairly Dividing a Cake after Some Parts were Burnt in the Oven. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1276–1284, 2018. (Cited on page 4)

[SH20] Erel Segal-Halevi. Competitive Equilibrium for Almost All Incomes: Existence and Fairness. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–50, 2020. (Cited on page 5)

[SS74] Lloyd Shapley and Herbert Scarf. On Cores and Indivisibility. *Journal of mathematical economics*, 1(1):23–37, 1974. (Cited on page 10)

[Str80] Walter Stromquist. How to Cut a Cake Fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980. (Cited on pages 1 and 4)

[Su99] Francis Edward Su. Rental Harmony: Sperner's Lemma in Fair Division. *The American Mathematical Monthly*, 106(10):930–942, 1999. (Cited on pages 1, 3, 4, 5, and 16)

[Tra02] Martino Traxler. Fair Chore Division for Climate Change. *Social Theory and Practice*, 28(1):101–134, 2002. (Cited on page 2)