Achieving a Fairer Future by Changing the Past

Jiafan He[1] , **Ariel D. Procaccia**[2] , **Alexandros Psomas**[2] and **David Zeng**[2]
[1]Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]Computer Science Department, Carnegie Mellon University
hejf15@mails.tsinghua.edu.cn, {arielpro, cpsomas}@cs.cmu.edu, dzeng@andrew.cmu.edu

Abstract

We study the problem of allocating T indivisible items that arrive online to agents with additive valuations. The allocation must satisfy a prominent fairness notion, envy-freeness up to one item (EF1), at each round. To make this possible, we allow the reallocation of previously allocated items, but aim to minimize these so-called *adjustments*. For the case of two agents, we show that algorithms that are informed about the values of future items can get by without any adjustments, whereas uninformed algorithms require $\Theta(T)$ adjustments. For the general case of three or more agents, we prove that even informed algorithms must use $\Omega(T)$ adjustments, and design an uninformed algorithm that requires only $O(T^{3/2})$.

1 Introduction

One of the most well-studied problems in computational social choice [Brandt *et al.*, 2016] is that of *fairly* allocating *indivisible* items when agents have *additive* valuations [Kurokawa *et al.*, 2018; Bouveret and Lemaître, 2016; Bouveret *et al.*, 2017; Caragiannis *et al.*, 2016; Barman and Krishnamurthy, 2017; Barman *et al.*, 2018; Amanatidis *et al.*, 2016; Cole and Gkatzelis, 2015]. Here, 'indivisible' means that items cannot be split between multiple agents, and 'additive valuations' means that the value of an agent for a bundle of items is the sum of his values for individual items in the bundle; it remains to define 'fairly.'

In contrast to several prominent fair division settings — such as cake cutting [Brams and Taylor, 1996; Procaccia, 2016] and rent division [Su, 1999; Gal *et al.*, 2017] — the foregoing setting clearly does not admit solutions that are *envy free*, in the sense that each agent (weakly) prefers his own bundle to the bundle of any other agent. But a natural relaxation of envy-freeness can always be guaranteed [Lipton *et al.*, 2004]. Specifically, an allocation is *envy free up to one item (EF1)* if, for every two agents i and j, it is sufficient to remove a single item from the bundle of j to eliminate any envy i might have had. This notion of fairness underlies a widely-used algorithm for the allocation of indivisible items, which is deployed on the website `Spliddit.org` [Goldman and Procaccia, 2014; Caragiannis *et al.*, 2016].

The picture becomes muddier, however, when the items arrive *online* instead of being available upfront, and must be allocated as they arrive. This setting was most recently explored by Benadè et al. [2018]. Assuming values are normalized to be in $[0, 1]$, they show that the maximum envy — the maximum difference between an agent's value for another bundle and his own bundle — must be (roughly) as large as $\Omega(\sqrt{T})$ after T rounds in the worst case. That is, not only is it impossible to achieve EF1, but agents must become increasingly envious over time.

We propose to circumvent this obstacle by slightly relaxing the requirements. Specifically, we allow *adjustments* to the allocation — each adjustment is a reallocation of a previously allocated item. This is relevant in any setting where allocations are provisional, and can change as new items become available. A common example is the (re)distribution of expensive scientific equipment (such as confocal laser scanning microscopes) among different departments within a college or a medical school.

Needless to say, adjustments are undesirable and should be avoided as much as possible, subject to maintaining fairness. More formally, our research question is:

> *What is the minimum number of adjustments needed to guarantee EF1 in each round, as a function of the number of items (equivalently, rounds) and the number of agents?*

A simple baseline is to reallocate all items in each round, which would require $\Theta(T^2)$ adjustments for T rounds. Of course, we aim to do better.

1.1 Our Results

We study the foregoing question in two related models. In the first model the allocation algorithm is *uninformed*, in that it has no information about items that will arrive in the future. In the second model, the algorithm is *informed*, that is, it knows the future; even in this model adjustments are inevitable due to the requirement of maintaining EF1 in each round. The former model is pertinent when the arrival of items is outside the organization's control, as is the case when items are donated. The latter model is relevant when the arrival of items follows a budgetary or production schedule. Real-world settings are likely to include both predictable and unpredictable item arrivals, but, as we shall see, it is actually

Number of agents	Uninformed algorithm		Informed algorithm
$n = 2$ (Sec. 3)	$\Theta(T)$ (Thms. 3.2 and 3.3)		0 (Thm. 3.4)
$n > 2$ (Sec. 4)	$O(T^{3/2})$ (Thm. 4.1)	$\Longrightarrow$	$O(T^{3/2})$
	$\Omega(T)$	$\Longleftarrow$	$\Omega(T)$ (Thm. 3.3)

Table 1: Minimum number of adjustments required to achieve EF1 in each of T rounds.

quite difficult to leverage (even complete) information about future arrivals.

We first look at the problem when there are only two agents (Section 3), and demonstrate a separation between the informed and uninformed settings. In the uninformed setting, we give an allocation algorithm that uses at most T adjustments to maintain EF1 at each round. We then show that this is tight up to constant factors by constructing an instance that ensures that any allocation algorithm would need $\Omega(T)$ adjustments to maintain an EF1 allocation. By contrast, when our algorithm knows the values for all the items that will arrive, it becomes possible to maintain an EF1 allocation without using any adjustments.

For the general case of three or more agents (Section 4), we show that even an informed algorithm requires $\Omega(T)$ adjustments to maintain EF1. In addition, we present an uninformed algorithm that uses $O(T^{3/2})$ adjustments — an improvement over the baseline of $O(T^2)$ adjustments. Note that here we cannot separate the informed and uniformed settings, and leave open the challenging problem of obtaining tighter bounds, as we discuss in Section 5.

Our results are summarized in Table 1.

1.2 Related Work

On a high level, our paper is related to the literature on *online* or *dynamic* fair division [Benadè *et al.*, 2018; Friedman *et al.*, 2015; Friedman *et al.*, 2017; Aleksandrov *et al.*, 2015; Aleksandrov and Walsh, 2017; Walsh, 2011; Kash *et al.*, 2014]. We elaborate on the two most closely related papers.

Benadè et al. [2018] consider indivisible items that arrive online, additive valuations, and uninformed algorithms. Assuming values are in $[0, 1]$, they design a deterministic algorithm that achieves maximum envy of $\tilde{O}(\sqrt{T/n})$, where T is the number of items (rounds) and n is the number of agents. They also show that this bound is tight up to polylogarithmic factors (and extend these results to a setting where items arrive in batches that are allocated simultaneously). By contrast, we are able to achieve EF1, and circumvent the lower bound of Benadè et al., by allowing adjustments.

Apropos adjustments, they are inspired by the notion of *disruptions*, first suggested by Friedman et al. [2015] as a way to achieve fairness in an online setting, albeit a fundamentally different one. In their setting, it is the agents — not the items — that arrive dynamically, and, in fact, there is only a single *divisible* good. The utility of each agent only depends on the fraction of the good he is allocated. A disruption here means taking a fraction of the good that has been allocated to an agent and redistributing it. The goal is to optimize certain measures of fairness (*fairness ratio* and *envy ratio*) while minimizing disruptions. Friedman et al. give optimal bounds that relate the allowed number of disruptions per round to their measures of fairness. In a subsequent paper [Friedman *et al.*, 2017], they extend the results to the case where less than one disruption per arrival, on average, is allowed.

2 Preliminaries

For each natural number $s \in \mathbb{N}$, we let $[s] = \{1, \ldots, s\}$. In our setting, there is a set $A = \{a_1, a_2, \ldots, a_n\}$ of n agents and a set $G = \{g_1, g_2, \ldots, g_T\}$ of T items (also known as goods, hence the notation).

Each agent a_i has a valuation function v_i; $v_i(S)$ is the value a_i has for a subset of items S. We simplify notation by using $v_i(g_j)$ to denote $v_i(\{g_j\})$. A valuation function v_i is additive if $v_i(S) = \sum_{g \in S} v_i(g)$. We assume throughout this paper that agents have additive valuation functions.

An *allocation* of the goods is a partition $A = (A_1, \ldots, A_n)$, where A_i is the bundle of goods allocated to agent a_i. We are interested in fair allocations that are, specifically, *envy-free up to one item (EF1)*. Formally, an allocation $(A_1, A_2, \ldots A_n)$ is EF1 if for any i, j such that $v_i(A_i) < v_i(A_j)$, there exists an item $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$.

Despite our focus on this 'qualitative' notion of fairness, we often find it useful to refer to a numerical value for envy. Given an allocation A, we define

$$\mathsf{ENVY}_{i,j}(A) = v_i(A_j) - v_i(A_i).$$

2.1 The Online Setting

We assume that the items arrive in order, one per round, over T total rounds. Let $G^t = \{g_1, g_2, \ldots g_t\}$ be the set of items that have arrived by round t. We denote an allocation of G^t by $A^t = (A_1^t, \ldots, A_n^t)$. An online allocation algorithm outputs an allocation A^t of the items G^t for each round $t \in [T]$. When we say that an online algorithm is EF1, we mean that A^t is EF1 for all $t \in [T]$. We also use $\mathsf{ENVY}_{i,j}^t$ as a shorthand for $\mathsf{ENVY}_{i,j}(A^t)$. Throughout the paper we use superscripts in the allocations to denote the round and subscripts to denote the agent.

An *adjustment* is a reallocation of a previously allocated item to a different agent. The number of adjustments needed by an allocation algorithm at round t is the number of adjustments needed to go from A^{t-1} to A^t. Formally, this can be expressed as $\sum_{i \in [n]} |A_i^t \setminus (A_i^{t-1} \cup \{g_t\})|$. The sum of the number of adjustments needed across the T rounds is the number of adjustments needed by an online allocation algorithm.

2.2 Informed and Uninformed Algorithms

Intuitively, an informed algorithm 'knows the future', whereas an uninformed algorithms does not. To formalize this, we can think of the setting as a zero-sum two-player game between an algorithm and an adversary that chooses the values of agents for items. The algorithm must maintain EF1 at each round, and may need to use adjustments to that end. The payoffs in the game are the number of adjustments used (which the algorithm tries to minimize, and the adversary tries to maximize).

An *informed* algorithm moves second in this game, that is, it first observes the sequence of values chosen by the adversary, and then decides how to allocate. By contrast, an *uninformed* algorithm moves first, i.e. the algorithm is announced, and then the instance is constructed by the adversary.

Consequently, to establish a lower bound against an informed algorithm, we must construct a sequence of values that is independent of the algorithm. To establish a lower bound against an uninformed algorithm, we may design an adversary that chooses values for the items that depend on the history.

3 Two Agents

We start with the case of two agents, i.e., $n = 2$, which often plays a central role in computational fair division. We establish tight bounds on the required number of adjustments, which, in particular, show that informed algorithms are strictly more powerful than their uninformed counterparts.

3.1 Uninformed Algorithms

In order to present our uninformed algorithm for the two-agent case, we first introduce the concept of fractional allocations. Rather than each item being assigned to an agent, a fractional allocation is an n by m matrix X, where n is the number of agents and m the number of items. For each i, j, X_{ij} represents the proportion of g_j allocated to agent i. X is constrained so that $\sum_{i\in[n]} X_{ij} = 1$ for all j and $X_{ij} \in [0, 1]$ for all i, j. For a fractional allocation X, we define the number of fractional items as the number of items g_j for which there exists an i such that $X_{ij} \in (0, 1)$. All other items are whole items. In addition, let X_i denote the i^{th} row of X. X_i can be thought of as the allocation to agent i.

Given an additive valuation function for each agent, we can extend each agent's valuation function to work with fractional items by treating items as if they were divisible, that is, by setting $v_i(X_i) = \sum_{j\in[m]} X_{ij} v_i(g_j)$. Finally, an allocation is *proportional* if for all i, $v_i(X_i) \geq \frac{1}{n} \sum_{j\in[m]} v_i(g_j)$.

We now formulate a lemma that will play a central role in the design of our algorithm. While we require only the $n = 2$ case of the lemma, we establish a more general version, which may be of independent interest. It is related to a result of Bogomolnaia et al. [2017], but our proof is different.

Lemma 3.1. *For any proportional fractional allocation X, there exists a proportional allocation Y where Y has at most $n-1$ fractional items and every whole item in X is allocated to the same agent in Y. Moreover, Y can be computed in polynomial time.*

Proof. Consider the bipartite graph B induced by X, where edges are between two vertex sets A, G representing the agents and items. Let there be an edge between (a_i, g_j) when $X_{ij} > 0$.

Suppose B has a cycle C of length $2k$. Since the graph is bipartite, it contains k agents and k items. Note that whole items cannot be part of a cycle since they have degree 1. For ease of notation, in the following section, take all indices modulo k. Without loss of generality, assume that for each $i \in [k]$, $(a_i, g_i) \in C$ and $(a_i, g_{i+1}) \in C$.

Now suppose that for each item g_i, we were to transfer δ_i of the item from a_i to a_{i-1}. Consider the vector Δ, where Δ_i is the change in each agents' value for their own items. We can express Δ using the following linear equations, where we denote $v_{ij} = v_i(g_j)$:

$$\begin{bmatrix} -v_{11} & v_{12} & 0 & 0 & \cdots & 0 \\ 0 & -v_{22} & v_{23} & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ v_{k1} & 0 & 0 & 0 & \cdots & -v_{kk} \end{bmatrix} \cdot \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_k \end{bmatrix} = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_k \end{bmatrix}$$

Let V be the value matrix defined above. We case on whether V is invertible. If it is invertible, then there exists a vector δ such that $V \cdot \delta = \Delta$, where $\Delta = \vec{1}$. If V is not invertible, then there exists a non-zero vector δ such that $V\delta = \vec{0}$. In either case, we can find a non-zero δ such that $V\delta = \Delta$ where for each i, $\Delta_i \geq 0$.

We conclude that, if we choose c appropriately, $c \cdot \delta$ will describe a way to transfer items that is implementable (after the transfer, no agent will have a negative fraction of any item) while ensuring that there is at least one X_{ij} that is now equal to 0. After implementing the transfers $c \cdot \delta$, we know that $v_i(\hat{X}_i) \geq v_i(X_i)$, where $\hat{X}$ is the new allocation. Moreover, the number of i, j such that $X_{ij} > 0$ decreases by at least 1. Thus, the number of edges in the induced graph also decreases by 1.

We repeat the above until no cycles exist, which is guaranteed to happen because the number of edges is strictly decreasing in each step. Let Y be the resulting allocation. Since for each i, $v_i(X_i)$ cannot decrease, Y must be proportional.

Any graph with no cycles has at most $|V| - 1$ edges. So if there are n agents and m items, there are at most $n + m - 1$ edges. Since each item must have an edge to at least one agent, this implies that the number of fractional items in Y is at most $n - 1$. Because cycles do not contain any integral items in X, the allocation of each integral item stays the same.

The proof of existence clearly induces a computationally efficient algorithm, which we will refer to as fractional item elimination. □

We next present Algorithm 1. We compute a proportional fractional allocation X^t for every round t. To obtain the fractional allocation for items G^t we start with X^{t-1}, give a $\frac{1}{2}$ fraction of the newly arrived item to each agent and then apply fractional item elimination, as described in the proof of Lemma 3.1. We obtain the actual allocations by rounding fractional items to the agent with the larger portion of the

item. Since the (integral) allocation for round t, A^t, depends only on the value of the first t items, this is an uninformed algorithm.

Algorithm 1 Fractional Item Rounding

input: v_1, v_2

1: **for** $t = 1, \ldots, T$ **do**
2: Let $P \in \mathbb{R}^{2\times t}$ such that

$$P_{a,b} \leftarrow \begin{cases} X^{t-1}_{a,b} & \text{if } b < t \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

3: Let X^t be the allocation found by applying fractional item elimination to P under v_1, v_2
4: $A^t_1 \leftarrow \{g_k \in G^t : X^t_{1,k} \geq \frac{1}{2}\}$
5: $A^t_2 \leftarrow \{g_k \in G^t : X^t_{2,k} > \frac{1}{2}\}$
6: **end for**
7: **return** $[A^1, A^2, \cdots, A^T]$

Theorem 3.2. *Algorithm 1 is an uninformed EF1 algorithm for two agents that uses at most T adjustments.*

Proof. We first show that the algorithm maintains an EF1 allocation. From Lemma 3.1, in every round t the underlying fractional allocation X^t satisfies proportionality. When $n = 2$, this implies that X^t is also an envy-free allocation. Let A^t be the corresponding rounded allocation. Without loss of generality, assume that a_1 is assigned the only fractional item g_j. In this case, $v_1(A^t_1) \geq v_1(X^t_1) \geq v_1(X^t_2) \geq v_1(A^t_2)$ so a_1 does not envy a_2. In addition, because a_1 received g_j, we know that in the fractional allocation, a_1 received at least half of it and a_2 received at most half of it. Therefore,

$$\begin{aligned} v_2(A^t_2) &\geq v_2(X^t_2) - \frac{1}{2}v_2(g_j) \\ &\geq v_2(X^t_1) - \frac{1}{2}v_2(g_j) \geq v_2(A^t_1 \setminus \{g_j\}). \end{aligned}$$

Thus, the allocation satisfies EF1.

To show that at most T adjustments are needed, note that, on any round t, the only item that could be reallocated is the fractional item in X^{t-1}. Over T rounds, this leads to at most T adjustments. □

We complement Theorem 3.2 with an asymptotically tight lower bound.

Theorem 3.3. *Any uninformed EF1 algorithm must use at least $\lfloor T/6 \rfloor$ adjustments in the worst case, even when there are two agents.*

Proof. We construct an adversary strategy that is built around a repeated 6-step sequence. Sequence s_j will describe the values for items $6j+1$ to $6j+6$. For $t = 6j+1, 6j+3, 6j+5$, the item values are fixed at $v_1(g_t) = v_2(g_t) = 14^j$. For steps $t = 6j+2, 6j+4, 6j+6$, the adversary chooses item values that depend on the allocation of item g_{t-1}. If a_1 received g_{t-1} in A^{t-1}, then for g_t, $v_1(g_t) = 14^j$ and $v_2(g_t) = \frac{1}{4}14^j$. The case where agent a_2 receives the item is analogous, with the values swapped. The adversary strategy ensures each pair of items is allocated in a way that increases envy. After each sequence, the values of the items increase exponentially to minimize the effect of the allocation of previous items.

To establish the lower bound, we show that any allocation algorithm must use at least one adjustment for each 6-step sequence to maintain an EF1 allocation at each round. Assume for the sake of contradiction that for some sequence s_j, no adjustments are used during that sequence but the allocation at each round in the sequence is EF1.

We first show that for each item pair $(6j+1, 6j+2)$, $(6j+3, 6j+4)$, $(6j+5, 6j+6)$, each agent must receive exactly one item from each pair. Let $S^{6j} = \mathsf{ENVY}^{6j}_{1,2} + \mathsf{ENVY}^{6j}_{2,1}$. It holds that

$$S^{6j} \geq -\sum_{t=1}^{6j} |v_1(g_t) - v_2(g_t)| = -3\sum_{i=1}^{j-1} \frac{3}{4}14^i > -\frac{9}{52}14^j.$$

Note that if v is the largest value of any item in A^t, for A^t to be EF1, we must have $\mathsf{ENVY}^t_{i,j} < v$ for any i, j. In particular, the largest item value in A^{6j} is 14^{j-1} so it must hold that $\mathsf{ENVY}^{6j}_{1,2} \leq 14^{j-1}$. Since $\mathsf{ENVY}^{6j}_{2,1} = S^{6j} - \mathsf{ENVY}^{6j}_{1,2}$, we put the two inequalities together to conclude that

$$\mathsf{ENVY}^{6j}_{2,1} > -\frac{9}{52}14^j - 14^{j-1} > -\frac{1}{4}14^j.$$

We can show the same for $\mathsf{ENVY}^{6j}_{1,2}$. Thus, if both item $6j+1$ and $6j+2$ are given to the same agent, the allocation would not be EF1 since the other agent would have envy greater than 14^j. To show this also holds for the remaining item pairs, note that after the arrival of any item pair, the envy of each agent can only increase.

Finally, we know that one of the agents received at least two of the (less valuable) items $6j+2, 6j+4, 6j+6$. Without loss of generality, suppose it was agent 1. Then we have

$$\mathsf{ENVY}^{6j+6}_{1,2} \geq \mathsf{ENVY}^{6j}_{1,2} + 2 \cdot (14^j - \frac{1}{4}14^j) > 14^j,$$

implying that the allocation is not EF1.

In Table 2, we give an example of a sequence assuming no adjustments. □

3.2 Informed Algorithms

We now turn to the setting where the algorithm is informed of the values of all items upfront. Lipton et al. [2004] introduce the envy cycle elimination algorithm for finding EF1 allocations in the offline setting. We describe a modified version of the algorithm that produces an allocation for the offline setting. This allocation has the property that all prefixes of the allocation are also EF1 allocations. This leads to the surprising result that no adjustments are needed to achieve an EF1 allocation at each round when the algorithm is informed.

Algorithm 2 considers the items one at a time in the order they are to arrive and builds an allocation iteratively, with a candidate allocation C^t for each round t. We maintain a counter s that keeps track of the last round s in which C^s was envy-free. Given a candidate allocation C^y and $x \leq y$, define $C^{x,y}$ to be the candidate allocation C^y when only considering items in $G^y \setminus G^x$.

Round t	$6j$	$6j+1$	$6j+2$	$6j+3$	$6j+4$	$6j+5$	$6j+6$
Value of g_t to Agent 1	$\cdots$	$[14^j]$	14^j	14^j	$[\frac{1}{4}14^j]$	$[14^j]$	14^j
Value of g_t to Agent 2	$\cdots$	14^j	$[\frac{1}{4}14^j]$	$[14^j]$	14^j	14^j	$[\frac{1}{4}14^j]$
Lower bound for $\mathsf{ENVY}^t_{1,2}$	$-\frac{1}{4}14^j$	$-\frac{5}{4}14^j$	$-\frac{1}{4}14^j$	$\frac{3}{4}14^j$	$\frac{1}{2}14^j$	$-\frac{1}{2}14^j$	$\frac{1}{2}14^j$
Lower bound for $\mathsf{ENVY}^t_{2,1}$	$-\frac{1}{4}14^j$	$\frac{3}{4}14^j$	$\frac{1}{2}14^j$	$-\frac{1}{2}14^j$	$\frac{1}{2}14^j$	$\frac{3}{2}14^j$	$\frac{5}{4}14^j$

Table 2: An example of the adversarial sequence from the proof of Theorem 3.3, assuming no adjustments are made. The lower bounds are non-inclusive and the brackets denote which agent the item was given to.

We generate C^t by building from C^{t-1}. In Lines 3–7, we assign the item g_t to an arbitrary unenvied agent in $C^{s,t-1}$. If this results in both agents envying each other in $C^{s,t}$, in Line 9, we swap the allocation of items $G^t \setminus G^s$ in C^t. This step guarantees there will always be an unenvied agent when assigning the next item.

The allocation for round t is found by taking the final candidate allocation C^T and only considering the first t items. In contrast to Algorithm 1, therefore, the allocation for round t does depend on the values for all T items, which restricts this algorithm to the informed setting.

Algorithm 2 Envy Balancing

input: v_1, v_2

1: $s \leftarrow 0, C^0 = (\emptyset, \emptyset)$
2: **for** $t = 1, \ldots, T$ **do**
3: **if** a_1 is unenvied in $C^{s,t-1}$ **then**
4: $C^t \leftarrow (C_1^{t-1} \cup \{g_t\}, C_2^{t-1})$
5: **else**
6: $C^t \leftarrow (C_1^{t-1}, C_2^{t-1} \cup \{g_t\})$
7: **end if**
8: **if** a_1 and a_2 envy each other in $C^{s,t}$ **then**
9: $C^t \leftarrow ((C_1^t \setminus C_1^{s,t}) \cup C_2^{s,t}, (C_2^t \setminus C_2^{s,t}) \cup C_1^{s,t})$
10: **end if**
11: **if** $C^{s,t}$ is envy-free **then**
12: $s \leftarrow t$
13: **end if**
14: **end for**
15: **for** $t = 1, \ldots, T$ **do**
16: $A^t = (C_1^T \cap G^t, C_2^T \cap G^t)$
17: **end for**
18: **return** $[A^1, A^2, \cdots, A^T]$

To analyze the above algorithm, we introduce the following notation. Let A be the allocation given by Algorithm 2. Then define $A^{x,y}$ to be the allocation A but only considering items in $G^y \setminus G^x$, and let $\mathsf{ENVY}^{x,y}_{i,j} = \mathsf{ENVY}_{i,j}(A^{x,y})$.

Theorem 3.4. *Algorithm 2 is an informed EF1 algorithm for two agents that requires no adjustments.*

Proof. The algorithm requires no adjustments by design, so we focus on establishing EF1. Consider an arbitrary $t \in [T]$ and let s be the last round before t in which candidate allocation C^s is envy-free. Let $M_1 = \max\{v_1(g_j) : g_j \in A_2^{s,t}\}$. Let M_2 be defined analogously.

We claim that it is sufficient to show that

$$\mathsf{ENVY}^{s,t}_{1,2} \leq M_1 \text{ and } \mathsf{ENVY}^{s,t}_{2,1} \leq M_2. \quad (1)$$

Indeed, for any t,

$$\mathsf{ENVY}^t_{1,2} = \mathsf{ENVY}^s_{1,2} + \mathsf{ENVY}^{s,t}_{1,2}.$$

By definition of s, $\mathsf{ENVY}^s_{1,2} \leq 0$ and from Equation (1), $\mathsf{ENVY}^{s,t}_{1,2} \leq M_1$. Therefore $\mathsf{ENVY}^t_{1,2} \leq M_1$, and the analogous inequality holds for agent a_2.

We therefore focus on proving (1). We will show that $\mathsf{ENVY}^{s,t}_{1,2} \leq M_1$; the proof of $\mathsf{ENVY}^{s,t}_{2,1} \leq M_2$ is analogous.

We claim that either $A^{s,t} = (C_1^{s,t}, C_2^{s,t})$ or $A^{s,t} = (C_2^{s,t}, C_1^{s,t})$. First, no swaps occur between s and t. To see why, observe that if a swap occurred when generating C^x for $s < x \leq t$, the candidate allocation would be envy-free since C^x consists of C^s and $C^{s,x}$ and both portions are now envy free, so s would have been updated to x. Now consider the next candidate allocation C^y, for $y > t$, which is envy free (if any). If a swap occurred on round y, then $A^{s,t} = (C_2^{s,t}, C_1^{s,t})$ and otherwise $A^{s,t} = (C_1^{s,t}, C_2^{s,t})$. As a result, it is sufficient to show that $|\mathsf{ENVY}_{1,2}(C^{s,t})| \leq M_1$.

We can show this through induction on t. Suppose that $s = t-1$. In this case, the allocation $C^{s,t}$ consists of one item so $|\mathsf{ENVY}_{1,2}(C^{s,t})| \leq M_1$. Otherwise, we know that $s < t-1$. In this case, we can apply the induction hypothesis which tells us that for the same s and some $M_1' \leq M_1$, $|\mathsf{ENVY}_{1,2}(C^{s,t-1})| \leq M_1'$.

Now suppose that $\mathsf{ENVY}_{1,2}(C^{s,t-1}) > 0$. a_2 is envied so this implies the next item would be given to a_1. Since the value of the next item is at most M_1, we know that $\mathsf{ENVY}_{1,2}(C^{s,t}) \geq -M_1$. In addition, the envy can only decrease which means $\mathsf{ENVY}_{1,2}(C^{s,t}) \leq M_1' \leq M_1$.

Otherwise, $\mathsf{ENVY}_{1,2}(C^{s,t-1}) \leq 0$. Since $s < t-1$, the allocation $C^{s,t-1}$ is not envy-free, and so it must be that a_1 is envied, and the next item is given to a_2. Recalling that $\mathsf{ENVY}_{1,2}(C^{s,t-1}) \geq -M_1$, it follows that $|\mathsf{ENVY}_{1,2}(C^{s,t})| \leq M_1$. □

4 More Than Two Agents

In this section we explore the general case of $n > 2$, which — we will show — is qualitatively different from the case of two

agents.

For our upper bound, consider the round-robin protocol for allocating items. In this protocol, we start with an arbitrary ordering of agents. Following this ordering, each agent takes turns selecting their most preferred item out of the remaining available items, continuing until no items remain.

This protocol is used in the following algorithm for the uninformed setting for more than two agents. For ease of exposition, we will assume here that T is a perfect square.

For each round t, we take the items G^t and divide them into a main pile and side pile. Let $q = \lfloor t/\sqrt{T} \rfloor$. The first $q\sqrt{T}$ items go into the main pile while the remaining $t - q\sqrt{T}$ go to the side pile. Equivalently, in each round, the new item g_t goes into the side pile and every $\sqrt{T}$ rounds, the side pile is emptied and the items are moved to the main pile.

On each round, to generate A^t, we allocate the main pile using the ordering $a_1 > a_2 > \cdots > a_n$ and the side pile using the reverse ordering $a_n > \cdots > a_1$, and then merge the two allocations together. We use adjustments to maintain the structure of these allocations at each step and, thus, this algorithm is uninformed.

Algorithm 3 Double Round Robin

input: v_i for each agent a_i

$M \leftarrow \emptyset, S \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $S \leftarrow S \cup \{g_t\}$
  if t is a multiple of $\sqrt{T}$ **then**
    $M \leftarrow M \cup S$
    $S \leftarrow \emptyset$
  end if
  $A_M \leftarrow$ ROUND-ROBIN$(M, a_1 > \cdots > a_n)$
  $A_S \leftarrow$ ROUND-ROBIN$(S, a_n > \cdots > a_1)$
  Let A^t be the combination of allocations A_S and A_M
end for
return $[A^1, A^2, \cdots, A^T]$

Theorem 4.1. *Algorithm 3 is an uninformed EF1 algorithm that requires $O(T^{3/2})$ adjustments.*

Proof. To see why the allocation on each round is EF1, we make a few observations about the round-robin protocol. Let B be an ordering and let $a_i >_B a_j$ denote that a_i chooses before a_j in that ordering. One can easily see that if we use round-robin to allocate some set of items G, then for any pair of agents a_i, a_j, a_i will envy a_j by at most one item. Furthermore, if $a_i >_B a_j$, then a_i will not envy a_j. To show this, note that when $a_i >_B a_j$, we can match each item that a_j receives to a distinct item that a_i receives such that a_i prefers his item to the item it is matched with. Otherwise, when $a_j >_B a_i$, we can remove the first item a_j receives and then proceed with matching.

Since the main and side piles are allocated using reversed orderings, a_i envies a_j in one allocation only if a_i does not envy a_j in the other allocation. In both allocations, we know a_i envies a_j by at most one item, so after combining the allocations, a_i still envies a_j by at most one item.

Now, we show that this algorithm uses at most $O(T^{3/2})$ adjustments. If we consider successive rounds, the main pile will only change every $\sqrt{T}$ rounds. The first change uses at most $\sqrt{T}$ adjustments, the second one uses at most $2\sqrt{T}$, and so on, until the last change that uses T adjustments, for a total of (at most) $T^{3/2}$ adjustments. Meanwhile, the side pile can be completely reallocated every round, but reaches size at most $\sqrt{T}$. Therefore, the total number of adjustments needed to maintain the allocation of the side pile over T rounds can be upper-bounded by $T^{3/2}$. Overall, this leads to at most $2T^{3/2}$ adjustments. □

In contrast to the case of two agents, where informed algorithms can achieve EF1 at each round without adjustments, when $n \geq 3$, a linear number of adjustments is inevitable.

Theorem 4.2. *For $n \geq 3$, any informed EF1 algorithm must use at least $\Omega(T)$ adjustments in the worst case.*

The theorem's proof is relegated to the full version.[1]

5 Discussion

While our results for the case of two agents are tight — and show a stark separation between informed and uninformed algorithms — our bounds for the general case are not. The most interesting open question is whether there is a separation between informed and uninformed algorithms in the general case. Our results allow for the possibility of informed EF1 algorithms that require $O(T)$ adjustments, and a lower bound of $\Omega(T^{3/2})$ on the adjustments required by uninformed algorithms. However, we conjecture that $O(T)$ adjustments are sufficient even for uninformed algorithms. In fact, we can prove this in the special case where n is constant, and each $v_i(j)$ can only take a constant number of values; see Appendix B (and Footnote 1) for details.

In addition, our results focus on achieving EF1 at each round, and this fairness guarantee may not be sufficient to ensure truly satisfying outcomes. For example, a number of recent papers aim to guarantee EF1 and Pareto efficiency simultaneously [Caragiannis *et al.*, 2016; Barman *et al.*, 2018]. However, as noted by Benadè et al. [2018], it is crucial to first understand fairness constraints in isolation; if the requirements were more stringent, our negative results would immediately carry over, while our positive results would potentially serve as building blocks for more elaborate algorithms.

Acknowledgments

This work was partially supported by the National Science Foundation under grants IIS-1350598, IIS-1714140, CCF-1525932, and CCF-1733556; by the Office of Naval Research under grants N00014-16-1-3075 and N00014-17-1-2428; and by a Sloan Research Fellowship and a Guggenheim Fellowship.

[1]The full version of this paper can be found at http://procaccia.info/papers/adjustments.pdf.

References

[Aleksandrov and Walsh, 2017] M. Aleksandrov and T. Walsh. Pure Nash equilibria in online fair division. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 42–48, 2017.

[Aleksandrov *et al.*, 2015] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2540–2546, 2015.

[Amanatidis *et al.*, 2016] G. Amanatidis, G. Birmpas, and E. Markakis. On truthful mechanisms for maximin share allocations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 31–37, 2016.

[Barman and Krishnamurthy, 2017] S. Barman and S. K. Krishnamurthy. Approximation algorithms for maximin fair division. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 647–664, 2017.

[Barman *et al.*, 2018] S. Barman, S. K. Krishnamurthy, and R. Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018.

[Benadè *et al.*, 2018] G. Benadè, A. M. Kazachkov, A. D. Procaccia, and C.-A. Psomas. How to make envy vanish over time. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 593–610, 2018.

[Bogomolnaia *et al.*, 2017] A. Bogomolnaia, H. Moulin, F. Sandomirskiy, and E. Yanovskaia. Dividing goods or bads under additive utilities. arXiv:1608.01540, 2017.

[Bouveret and Lemaître, 2016] S. Bouveret and M. Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.

[Bouveret *et al.*, 2017] S. Bouveret, K. Cechlárová, E. Elkind, A. Igarashi, and D. Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 135–141, 2017.

[Brams and Taylor, 1996] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[Brandt *et al.*, 2016] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Caragiannis *et al.*, 2016] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum Nash welfare. In *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*, pages 305–322, 2016.

[Cole and Gkatzelis, 2015] R. Cole and V. Gkatzelis. Approximating the Nash social welfare with indivisible items. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 371–380, 2015.

[Friedman *et al.*, 2015] E. J. Friedman, C.-A. Psomas, and S. Vardi. Dynamic fair division with minimal disruptions. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, pages 697–713, 2015.

[Friedman *et al.*, 2017] E. J. Friedman, C.-A. Psomas, and S. Vardi. Controlled dynamic fair division. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 461–478, 2017.

[Gal *et al.*, 2017] Y. Gal, M. Mash, A. D. Procaccia, and Y. Zick. Which is the fairest (rent division) of them all? *Journal of the ACM*, 64(6): article 39, 2017.

[Goldman and Procaccia, 2014] J. Goldman and A. D. Procaccia. Spliddit: Unleashing fair division algorithms. *SIGecom Exchanges*, 13(2):41–46, 2014.

[Kash *et al.*, 2014] I. Kash, A. D. Procaccia, and N. Shah. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research*, 51:579–603, 2014.

[Kurokawa *et al.*, 2018] D. Kurokawa, A. D. Procaccia, and J. Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM*, 64(2): article 8, 2018.

[Lipton *et al.*, 2004] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 6th ACM Conference on Economics and Computation (EC)*, pages 125–131, 2004.

[Procaccia, 2016] A. D. Procaccia. Cake cutting algorithms. In F. Brandt, V. Conitzer, U. Endress, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13. Cambridge University Press, 2016.

[Su, 1999] F. E. Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.

[Walsh, 2011] T. Walsh. Online cake cutting. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, pages 292–305, 2011.