How to Make Envy Vanish Over Time

GERDUS BENADE, Carnegie Mellon University, USA
ALEKSANDR M. KAZACHKOV, Carnegie Mellon University, USA
ARIEL D. PROCACCIA, Carnegie Mellon University, USA
CHRISTOS-ALEXANDROS PSOMAS, Carnegie Mellon University, USA

We study the dynamic fair division of indivisible goods. Suppose T items arrive online and must be allocated upon arrival to one of n agents, each of whom has a value in $[0, 1]$ for the current item. Our goal is to design allocation algorithms that minimize the maximum *envy* at time T, $\textsc{Envy}_T$, defined as the maximum difference between any agent's overall value for items allocated to another agent and to herself. We say that an algorithm has *vanishing envy* if the ratio of envy over time, $\textsc{Envy}_T/T$, goes to zero as T goes to infinity. We design a polynomial-time, deterministic algorithm that achieves $\textsc{Envy}_T \in \tilde{O}(\sqrt{T/n})$, and show that this guarantee is asymptotically optimal. We also derive tight (in T) bounds for a more general setting where items arrive in batches.

1 INTRODUCTION

We consider the setting of fairly allocating indivisible goods when agents have additive valuations. It involves a set $[n] = \{1, \ldots, n\}$ of agents, and a set of items. Each agent $i \in [n]$ assigns a normalized[1] value $v_{it} \in [0, 1]$ to each item t; for a bundle of items S, the value of agent i is $v_i(S) = \sum_{t \in S} v_{it}$. An *allocation* is a partition of the items into bundles $A_1, \ldots, A_n$, where A_i is assigned to agent $i \in [n]$.

Ideally, we would like to guarantee a fairness property called *envy-freeness* — arguably, the gold standard of fairness — which requires that each agent is at least as happy with her own allocation as the allocation of any other agent, that is, $v_i(A_i) \geq v_i(A_j)$ for any two agents $i, j \in [n]$. Envy-free solutions indeed always exist in other paradigmatic fair division settings that involve divisible goods or a numéraire, such as cake cutting [5, 14] and rent division [10, 19]. By contrast, in our context, envy is clearly unavoidable — just imagine a single (indivisible) item that is desired by two agents. That is why recent papers [7, 12] focus on the relaxed notion of *envy-freeness up to one good (EF1)*, which, when values are normalized as above, implies that $v_i(A_i) \geq v_i(A_j) - 1$ for all $i, j \in [n]$. Unlike its more stringent cousin, EF1 can always be guaranteed, and, in fact, it is quite easy to do so, e.g., by allocating the items in a round-robin fashion — each agent in her turn picks her favorite item among those that are still available.

Our point of departure is that we allow items to arrive *online*, that is, we must choose how to allocate an item when it arrives, without knowing the values of items that will arrive in the future.

[1]This assumption is made primarily for ease of exposition.

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.

Authors' addresses: Gerdus Benade, Carnegie Mellon University, Pittsburgh, PA, 15213, USA; Aleksandr M. Kazachkov, Carnegie Mellon University, Pittsburgh, PA, 15213, USA; Ariel D. Procaccia, Carnegie Mellon University, Pittsburgh, PA, 15213, USA; Christos-Alexandros Psomas, Carnegie Mellon University, Pittsburgh, PA, 15213, USA.

XXXX-XXXX/2018/5-ART $15.00
https://doi.org/10.1145/3219166.3219179

This setup mirrors common decision-making scenarios in humanitarian logistics. A paradigmatic example is that of food banks [1], which receive food donations, and deliver them to nonprofit organizations such as food pantries and soup kitchens. Indeed, items are often perishable, which is why allocation decisions must be made quickly, and donated items are typically leftovers, leading to lack of information about items that will arrive in the future.

Suppose, then, that at each time step an item t arrives, where $t = 1, \ldots, T$. The allocation decisions made by an algorithm at each step induce an allocation $A_1, \ldots, A_n$ at the end of step T. For $i, j \in [n]$, let

$$\textsc{Envy}_T^{ij} = \max\left\{v_i(A_j) - v_i(A_i), 0\right\}$$

be the envy of i for j. Note that this measure of cumulative envy increases by v_{it} if item t is allocated to j (which places it in A_j), and decreases by v_{it} if item t is allocated to i (which places it in A_i). Moreover, let

$$\textsc{Envy}_T = \max_{i,j \in [n]} \textsc{Envy}_T^{ij}$$

be the maximum envy. As noted above, in the static setting it is possible to maintain $\textsc{Envy}_T \leq 1$ for any number of items T, but, of course, even the round-robin allocation requires complete information about values upfront. By contrast, in the online setting, one would expect $\textsc{Envy}_T$ to inevitably grow with T.

Nevertheless, we can hope to control the rate at which envy grows over time. Specifically, we aim to design algorithms with *vanishing envy* — algorithms that lead to $\textsc{Envy}_T \in o(T)$, or, equivalently, we want the *average per-round envy* $\textsc{Envy}_T/T$ to go to zero as T goes to infinity. Our primary research question is:

> *Are there online allocation algorithms with vanishing envy, and, if so, what is the optimal average per-round envy?*

1.1 Our Results

We first study randomized algorithms. The most natural candidate is the "random allocation" algorithm: allocate item t to an agent chosen uniformly at random. We analyze this algorithm against an adaptive adversary that chooses the values v_{it} for item t after seeing the (random) allocations for items 1 through $t - 1$. Our first step is showing that the optimal strategy for an adaptive adversary (against the random allocation algorithm) is in fact nonadaptive: the adversary always picks values $v_{it} = 1$. This makes the random variables for the envy between any agents i and j at steps t and t' independent, which enables us to use standard concentration inequalities for bounding the overall envy. Our result for this setting is that the random allocation algorithm has vanishing envy. Formally, Theorem 2.1 asserts that this algorithm achieves $\mathbb{E}\left[\textsc{Envy}_T\right] \in \tilde{O}(\sqrt{T/n})$.

One may hope that it would be possible to do better than allocating blindly. Surprisingly, though, we show that the random allocation algorithm is asymptotically optimal (up to logarithmic factors). Indeed, there exists an adaptive strategy for the adversary such that any allocation algorithm for T items accumulates envy in $\Omega((T/n)^{r/2})$, for any $r < 1$ (Theorem 2.13). However, despite its theoretical optimality, the random allocation algorithm is intuitively unappealing. We therefore turn our attention to deterministic algorithms, only to discover that simple, greedy schemes seem to fail miserably in this setting. For example, one can construct counterexamples for natural candidates, like allocating item t to minimize the maximum envy, or allocating it to the most envious agent.

Nevertheless, we prove, in Theorem 2.6, that there exists a deterministic polynomial-time algorithm with the same envy bound as the random allocation algorithm (up to logarithmic factors). The former algorithm is the result of derandomizing the latter with the method of *pessimistic estimators* [15]. Specifically, we define a potential function $\phi(t)$ that depends on the values of the first t items and the allocations of the first $t-1$ items. Out algorithm allocates item t in a way that $\phi(t)$ is minimized. Our proof has three main ingredients. First, the potential function is nonincreasing, i.e., $\phi(t+1) \leq \phi(t)$. Second, we prove that the potential at time t is an overestimation of the probability of having large envy if in steps $t+1$ through T we allocate items uniformly at random. Formally, $\phi(t) \geq \sum_{i,j \in [n]: i \neq j} \Pr[\textsc{Envy}_T^{ij} > \lambda]$, where λ is the (target) performance of our derandomized algorithm. Lastly, we show that $\phi(0) < 1$. Note that characterizing the adversary's optimal strategy is not a hurdle here: since our algorithm is deterministic, adaptivity does not help. Combined, the three ingredients imply that greedily minimizing the potential function ϕ achieves $\textsc{Envy}_T \in \tilde{O}(\sqrt{T/n})$.

Having completed the picture for the setting where one item arrives at a time, in Section 3 we proceed to study a more general model. Suppose, as before, that T items with adversarially chosen values arrive over time. Instead of arriving one by one, as assumed in Section 2, items arrive in *batches* of size m. In other words, at each time step a batch of m items arrives, for a total of T/m batches. To motivate this, note that in the food bank setting it is reasonable to wait until the end of the day before allocating all food donations that arrived that day. When batch t arrives, the algorithm learns the values of all m items for all agents, and must allocate these items immediately and irrevocably, before the next batch arrives. We use $\textsc{Envy}_{T,m}$ to denote the envy after the allocation of T items, arriving in batches of size m, with the convention that $\textsc{Envy}_{T,1} = \textsc{Envy}_T$. Since the allocation algorithm is less myopic in this setting, it is natural to expect stronger performance guarantees than in the $m = 1$ case. For example, in the extreme case where $m = T$, there exist algorithms that are envy free up to one good, so $\textsc{Envy}_{T,T} \leq 1$.

The upper bound of Theorem 2.1 when $m = 1$ may be interpreted as the expected distance from the origin of a random walk that remains stationary with probability $1 - 2/n$, and increases or decreases by 1, each with probability $1/n$. The "step size" of 1 is the maximum change in the envy between any pair of agents after the allocation of a single item; the number of nonstationary steps is expected to be $2T/n$. This interpretation informs our approach when items arrive in batches. It is easy to find an EF1 allocation for every batch of items (round-robin suffices). Under such an allocation the maximum change in any pairwise envy due to a single batch remains 1; however, the value of an agent's bundle is likely to change with every batch. Since there are T/m batches ("steps" in the random walk), we may expect a bound of the form $\textsc{Envy}_{T,m} \in \tilde{O}(\sqrt{T/m})$. Indeed, our main result for this setting, given in Theorem 3.3, is a deterministic algorithm that achieves this bound.

To realize this intuition, we first need to overcome a technical obstacle. Even though it is easy to find an allocation with small pairwise envy for a given batch, it is not obvious how to find allocations with low pairwise envy such that randomly outputting one of them results in an (*ex ante*) envy-free allocation. In the random walk interpretation, we need to keep the envy between agents i and j stationary in expectation, while at the same time maintaining a small step size. This is a crucial argument in the analysis of the one-by-one setting, in which it is trivially satisfied by allocating each item uniformly at random.

We overcome this obstacle by leveraging a result from the literature on continuous cake cutting, which allows us to show, for each batch, the existence of a fractional allocation that is entirely envy free and can be written as a convex combination of integral allocations with constant pairwise envy. We then use ideas from the derandomization of the random allocation algorithm of Section 2.2 to give a deterministic algorithm for the batch setting.[2] Finally, we prove that the lower bound for the $m = 1$ setting may be extended to show that $\textsc{Envy}_{T,m} \in \Omega(\sqrt{T/(mn)})$.

[2]Interestingly, we do not explicitly derandomize a specific randomized allocation algorithm as in Section 2.

1.2 Related Work

Conceptually our paper is related to the growing literature on *online* or *dynamic* fair division [1, 2, 9, 11, 13, 20]. In particular, motivated by applications to the food bank domain, Aleksandrov et al. [1] introduce and analyze a closely related setting where indivisible items arrive online. However, they generally assume that all values are in $\{0, 1\}$, i.e., each agent "likes" or "dislikes" every item. They introduce two simple mechanisms, Like and Balanced Like; the former allocates the current item uniformly at random among agents who like it, whereas the latter allocates the current item uniformly at random among agents who like it and have so far received the fewest items. The analysis of these mechanisms focuses on properties such as strategyproofness, envy-freeness, and impact on welfare. Most relevant to us is the observation that Balanced Like is EF1. This also highlights the technical differences between our setting and theirs, because, as noted above, with general values EF1 is impossible.

Our paper is also related to the vast body of work on *online learning* [6]. In the quintessential setting, *experts learning* (with full-information feedback), there are T days, and on each day the algorithm chooses to follow the advice of one of n experts. Then, the value of each expert is revealed, and the algorithm gains the value of the expert whose advice it chose to follow. The algorithm's regret is the difference between the total value accumulated by the best expert in hindsight and the value it itself has accumulated; a *no-regret learning algorithm* has the property that the ratio between regret and time goes to zero (*vanishing regret* may have been a more accurate term). Similarly, we are also interested in the difference in value accumulated over time. However, to the best of our knowledge the two problems are technically unrelated. To appreciate the difference, note that in our setting the values of the current item to all agents are known to the algorithm. But if the values of the different experts were known in the expert learning setting, the problem would be trivial — the algorithm would simply choose the expert with maximum value. Nevertheless, some of our notations were chosen to be consistent with those used in the online learning literature.

Finally, we can make a technical connection to the literature on *vector balancing games* [17]. At each time step $t \in [T]$, the adversary picks an n-dimensional vector with values in $[-1, 1]$, and the algorithm chooses to multiply this vector by either -1 or $+1$ and add it to a running partial sum vector. In one version of this game, the goal of the algorithm is to minimize the maximum entry of the partial sum vector, while the adversary wishes to maximize that quantity. For the case where $n = 2$ and items arrive one by one, our setting can be reduced to a version of vector balancing games equipped with a weaker adversary. This means that the upper bound of Spencer [17] applies to our setting (and matches our results): there exists a deterministic algorithm that guarantees envy in $O(\sqrt{T})$ when $n = 2$. Conversely, our lower bound for $n = 2$ matches the lower bound from that paper, indicating that the ostensibly weaker adversary that we consider — restricted to picking values from just one orthant — has roughly the same strength as the stronger adversary of Spencer [17]; consequently, our lower bound is significantly more involved. For more than two agents, the two problems appear unrelated, and, moreover, the batch setting has no equivalent in the vector balancing games literature.

2 WHEN ITEMS ARRIVE ONE BY ONE

In this section, we discuss our basic setting, in which, at each time step, exactly one item arrives. Proofs missing from this section can be found in the full version of this paper.

2.1 Random Allocation

A natural randomized algorithm for the case where items arrive one by one is to allocate each item to an agent selected uniformly at random; we refer to this as the *random allocation algorithm*. We analyze the random allocation algorithm by first characterizing the adversary's optimal strategy. We prove that for an adaptive adversary who maximizes $\mathbb{E}\left[\textsc{Envy}_T\right]$, where the expectation is with respect to the randomness of the algorithm, the optimal strategy is integral, that is, all values are in $\{0,1\}$. Using this, we show that the optimal strategy, in fact, assigns $v_{it} = 1$ for all $i \in [n], t \in [T]$. This optimal adversary strategy is nonadaptive, and therefore, since all the randomness is coming from the algorithm, the random variables for the envy between agents i and j at times t and t' are independent. Standard concentration inequalities for the envy between any pair of agents, combined with a union bound over all such pairs, gives an upper bound on the expected envy.

Theorem 2.1. *Suppose that $T \geq n \log T$, where* log *is the natural logarithm. Then the random allocation algorithm guarantees that $\mathbb{E}\left[\textsc{Envy}_T\right] \in O(\sqrt{T \log T/n})$.*

Note that the assumption of $T \geq n \log T$ is innocuous, otherwise we can give each agent at most $\log T$ items to achieve $\textsc{Envy}_T \leq \log T$.

Proof of Theorem 2.1. Consider a game tree[3] with nodes on $T+1$ levels. Every node on level $1, \ldots, T$ has n outgoing arcs labeled $1, \ldots, n$. The leaf nodes on level $T+1$ are labeled by the maximum envy for the corresponding path. Let Ω be the set of all paths from the root to a leaf node, so $|\Omega| = n^T$. Equivalently, Ω is the set of all possible allocations of the T items. For an allocation $\omega \in \Omega$, denote by $\omega_t \in [n]$ the agent to whom item $t \in [T]$ was allocated by ω.

A fully adaptive strategy s for the adversary is defined by labeling every internal node u with a value vector $s(u)$, where $s(u)_i$ is the value of agent i for the item corresponding to node u. The algorithm's strategy consists of selecting an outgoing edge, corresponding to an allocation of the item with valuation $s(u)$, at every node u. The adversary's strategy is allowed to depend on the allocations and valuations so far, i.e., the path from the root to u.

For a given adversary strategy s and an allocation ω, let $\textsc{Envy}^{ij}(s,\omega)$ denote the envy of agent i for agent j. Denote with $\textsc{Envy}(s,\omega) = \max_{i,j\in[n]} \textsc{Envy}^{ij}(s,\omega)$ the maximum envy experienced by any agent. The objective of the adversary is to choose a strategy s that maximizes the expected envy $\mathbb{E}[\textsc{Envy}(s,\omega)]$, where the expectation is taken over allocating every item uniformly at random.

We consider the algorithm that allocates every item uniformly at random. This is equivalent to picking a random outgoing edge at each node u. The following two lemmas show that the adversary labels every internal node of this tree with the vector $\mathbf{1}^n$. These lemmas are inspired by the work of Sanders [16] on load balancing.

Lemma 2.2. *The adversary has an optimal adaptive strategy that labels every internal node of the game tree with a vector in $\{0,1\}^n$.*

In a nutshell, Lemma 2.2 follows from the fact that under any allocation algorithm, for every agent's valuation of any item, it is possible to compute whether that item increases or decreases the maximum envy (in expectation). If it increases (resp. decreases) the maximum envy, the adversary benefits by increasing (resp. decreasing) the corresponding valuation to 1 (resp. to 0).

While the previous result holds for any allocation strategy, the following lemma leverages specific properties of the random allocation algorithm.

Lemma 2.3. *The adversary has an optimal adaptive strategy that labels every internal node of the game tree with the vector $\mathbf{1}^n$.*

[3]Typically we would think of an extensive-form game with nodes associated with the algorithm or the adversary, and arcs corresponding to actions (allocation of the current item in the case of the algorithm, value vector in the case of the adversary). However, because we consider a fixed algorithm here, it is more convenient to imagine an unusual, adversary-oriented game tree.

Proof. By Lemma 2.2, the adversary has an optimal strategy that labels every internal node with a vector in $\{0,1\}^n$. Let s be such an optimal strategy with the smallest number of zeros, and suppose (for the sake of contradiction) that there exist internal nodes that are not labeled $\mathbf{1}^n$. Let u on layer $\ell \in [T]$ be the node closest to a leaf node for which $s(u)$ contains a 0 and $s(u') = \mathbf{1}^n$ for all descendants u' of u. Without loss of generality assume $s(u)_i = 0$, so agent i has value 0 for item ℓ at node u. Define a strategy s' identical to s except that $s'(u)_i = 1$. Let $j(\omega) \in \arg\max_{j \in [n]} \textsc{Envy}^{ij}(s, \omega)$.

For any fixed $\omega \in \Omega$, changing s to s' only changes the envy of agent i and only for paths that go through u. In particular, if $\omega_\ell \neq i$, the envy of agent i toward agent ω_ℓ increases by 1, which only helps the adversary. By contrast, if $\omega_\ell = i$, the envy of agent i decreases by 1, toward every agent j such that $\textsc{Envy}^{ij}(s, \omega) > 0$; the maximum envy, $\textsc{Envy}(s, \omega)$, is only affected if $\textsc{Envy}(s, \omega) = \textsc{Envy}^{i,j(\omega)}(s, \omega)$.

Thus, let $\omega \in \Omega$ be an arbitrary path going through u with $\omega_\ell = i$ and satisfying $\textsc{Envy}(s, \omega) = \textsc{Envy}^{i,j(\omega)}(s, \omega) > 0$. Since agent i may not have been the unique agent having envy equal to $\textsc{Envy}(s, \omega)$, $\textsc{Envy}(s', \omega) \geq \textsc{Envy}(s, \omega) - 1$. Now consider the path ω' that is identical to ω except that $\omega_\ell = j(\omega)$. Observe that $\textsc{Envy}(s', \omega') = \textsc{Envy}(s, \omega') + 1$. Hence, any decrease in envy due to allocating item ℓ to agent i on ω is compensated for (in the calculation of expected envy) along ω'. Since ω was picked arbitrarily and the mapping $\omega \mapsto \omega'$ is injective, it follows that the expected envy under s' is at least the expected envy under s, and s' has fewer zeros than s, contradicting our assumption on s. □

The fact that the adversary is adaptive naturally introduces a dependence in the change in any pairwise envy from one arrival to the next. The value of Lemma 2.3 lies in the fact that it allows us to circumvent this dependence as though we are dealing with a nonadaptive adversary and express any pairwise envy as the sum of independent random variables.

Specifically, given this adversary strategy, define independent random variables

$$X_t^{ij} = \begin{cases} -1, & \text{with probability } 1/n, \\ 0, & \text{with probability } 1 - 2/n, \\ 1, & \text{with probability } 1/n \end{cases}$$

for all $t \in [T]$, $i, j \in [n]$. Clearly, $\textsc{Envy}_T^{ij} = \max_{i,j \in [n]} \{\sum_{t=1}^T X_t^{ij}, 0\}$. For each X_t^{ij}, $\mathbb{E}[X_t^{ij}] = 0$, $\mathbb{E}[(X_t^{ij})^2] = 2/n$ and $|X_t^{ij}| \leq 1$. We use a version of Bernstein's inequality to bound the probability of having large envy between any pair of agents i and j.

Lemma 2.4 (Bernstein 4). *Let $X_1, \ldots, X_T$ be independent variables with $\mathbb{E}[X_t] = 0$ and $|X_t| \leq M$ almost surely for all $t \in [T]$. Then, for all $\lambda > 0$,*

$$\Pr\left[\sum_{t=1}^T X_t > \lambda\right] \leq \exp\left(-\frac{\frac{1}{2}\lambda^2}{\sum_{t=1}^T \mathbb{E}\left[X_t^2\right] + \frac{1}{3}M\lambda}\right).$$

When applying this result to $\textsc{Envy}_T^{ij}$ (which equals $\sum_{t=1}^T X_t^{ij}$ when envy exists), it follows that

$$\Pr\left[\textsc{Envy}_T^{ij} \geq \lambda\right] = \Pr\left[\sum_{t=1}^T X_t^{ij} \geq \lambda\right] \leq \exp\left(-\frac{\frac{1}{2}\lambda^2}{\frac{2T}{n} + \frac{1}{3}\lambda}\right) = \exp\left(-\frac{3n\lambda^2}{12T + 2\lambda n}\right).$$

Let $\lambda = 10\sqrt{T \log T/n}$. Taking a union bound gives

$$\Pr\left[\textsc{Envy}_T \geq \lambda\right] = \Pr\left[\exists i, j \in [n] \text{ such that } \textsc{Envy}_T^{ij} \geq \lambda\right] \leq n^2 \exp\left(-\frac{300T \log T}{12T + 20\sqrt{nT \log T}}\right) \leq \frac{1}{T},$$

where the last inequality uses the assumption that $T \geq n \log T$. Since the maximum possible envy is T, the desired bound on expected envy directly follows. $\square$

2.2 Derandomization with Pessimistic Estimators

The astute reader might have noticed that considering an adaptive adversary induces an extensive-form game of complete information between the algorithm and the adversary. In such games, randomization does not provide any benefit to either agent, as the backward induction solution is optimal [3]. This implies that there exists a deterministic algorithm with the same envy guarantee as the random allocation algorithm, i.e., $\textsc{Envy}_T \in \tilde{O}(\sqrt{T/n})$. However, it is *a priori* unclear whether this can be achieved in polynomial time. In fact, simple deterministic algorithms do not even lead to vanishing envy, as the following example shows.

Example 2.5. Consider the algorithm that at step t allocates the item to the agent with the maximum envy (if she has positive value for the item, and otherwise, say, allocates to the agent with the highest value for the item). We claim that it leads to $\textsc{Envy}_T \in \Omega(T)$.

We construct an example where each agent envies the other by $t = 2$. For $t \geq 3$, whenever agent i has maximum envy, we present an item with value ϵ for her, and value 1 for the other agent. Table 1 summarizes the analysis.

Table 1. Blindly allocating to the agent with the highest envy leads to constant per-round envy.

t	1	2	3	4	5	$\cdots$
Value of agent 1	$\enclose{circle}{1/2}$	1	$\enclose{circle}{\epsilon}$	1	$\enclose{circle}{\epsilon}$	$\cdots$
Value of agent 2	$1/2$	$\enclose{circle}{1/4}$	1	$\enclose{circle}{\epsilon}$	1	$\cdots$
Envy of agent 1	$-1/2$	$1/2$	$1/2 - \epsilon$	$3/2 - \epsilon$	$3/2 - 2\epsilon$	$\cdots$
Envy of agent 2	$1/2$	$1/4$	$5/4$	$5/4 - \epsilon$	$9/4 - \epsilon$	$\cdots$

For $t \geq 2$, the envy of each agent increases by 1 every two steps. Therefore, the maximum envy at step $2t$ is approximately t, and $\textsc{Envy}_T/T$ approaches $1/2$ as T goes to infinity.

In the full version of this paper, we show that the average per-round envy is constant for another natural algorithm, which allocates item t in a way that the maximum envy after allocation is as small as possible. These examples show that even though a simple randomized algorithm is optimal and there exists a deterministic algorithm with the same guarantee, simple deterministic algorithms are not able to provide any useful bounds.

Nevertheless, we are able to match the randomized bound of Theorem 2.1 by derandomizing the random allocation algorithm with the method of *pessimistic estimators* [15]. The outcome is an intuitively pleasing, polynomial-time, deterministic algorithm that at each step minimizes a potential function, which is essentially a penalty function exponential in each of the pairwise envy expressions.

Theorem 2.6. *Suppose that $T \geq n \log n$. Then there exists a polynomial-time, deterministic algorithm that achieves $\textsc{Envy}_T \in O(\sqrt{T \log n / n})$.*

The rest of this section is devoted to the proof of Theorem 2.6.

The Algorithm

We define a potential function $\phi(t)$ that depends on n, T, the values of the first t items, as well as their allocations. When item t arrives, we allocate it to the agent for which the value of $\phi(t)$ is minimized. Call this algorithm $\mathcal{A}^*$. Since our algorithm is deterministic, an adversary that wants to maximize $\textsc{Envy}_T$ does not gain from being adaptive. It therefore suffices to analyze our algorithm for an arbitrary choice of item values.

Theorem 2.6 follows from choosing $\phi(t)$ in a way that satisfies three particular properties, stated in the following three lemmas. Given $t \in [T]$, let $\mathcal{A}^t$ be the algorithm that allocates, for all $\ell \in [t]$, the item ℓ to an agent for which $\phi(\ell)$ is minimized, and the remaining items $t+1, \dots, T$ uniformly at random. Let $\textsc{Envy}_T^{ij}(\mathcal{A}^t)$ be the envy of agent i for agent j at the end of the execution of $\mathcal{A}^t$.

Lemma 2.7. $\phi(t) \geq \sum_{i,j\in[n]} \Pr\left[\textsc{Envy}_T^{ij}(\mathcal{A}^t) > 10\sqrt{T\log n/n}\right]$.

Lemma 2.8. *For all $t \in [T-1]$, $\phi(t+1) \leq \phi(t)$.*

Lemma 2.9. *For $T \geq n \log n$, $\phi(0) < 1$.*

Proof of Theorem 2.6. Notice that $\mathcal{A}^T$ is exactly the same as the algorithm $\mathcal{A}^*$. Lemmas 2.8 and 2.9 imply that $\phi(T) < 1$. Combining with Lemma 2.7, we get that for any choice of item values, and therefore for the optimal adversary strategy,

$$\Pr\left[\exists i,j \in [n] : \textsc{Envy}_T^{ij}(\mathcal{A}^T) > 10\sqrt{\frac{T\log n}{n}}\right] \leq \sum_{i,j\in[n]} \Pr\left[\textsc{Envy}_T^{ij}(\mathcal{A}^T) > 10\sqrt{\frac{T\log n}{n}}\right] \leq \phi(T) < 1.$$

Since $\mathcal{A}^T$ is deterministic — all items have been allocated after time T — the inequality above implies that, for the allocation of items by $\mathcal{A}^T$, there is no $i,j \in [n]$ such that $\textsc{Envy}_T^{ij} > 10\sqrt{T\log n/n}$, and we conclude that

$$\textsc{Envy}_T = \max_{i,j\in[n]} \textsc{Envy}_T^{ij} \leq 10\sqrt{\frac{T\log n}{n}} \in O\left(\sqrt{\frac{T\log n}{n}}\right). \qquad \square$$

Setup

We now define $\phi(t)$ and prove that it satisfies the desired properties from Lemmas 2.7, 2.8, and 2.9. For $i,j,k \in [n]$ and $t \in [T]$, let y_{tk}^{ij} be a helper variable for the effect on envy between agents i and j when item t goes to agent k, i.e.,

$$y_{tk}^{ij} = \begin{cases} -1, & \text{if } k = i, \\ 0, & \text{if } k \neq i,j, \\ 1, & \text{if } k = j, \end{cases}$$

and let y_t^{ij} be the same but with the dependence on k implicit (as k is exactly determined given an allocation). Denote with $f_{ij}(t) = \sum_{\ell=1}^{t} y_\ell^{ij} v_{i\ell}$ the net value agent i has for agent j's allocation with respect to her own at time t. Notice that $\textsc{Envy}_t^{ij} = \max\{f_{ij}(t), 0\}$. Let $C = (1 + (e^s + e^{-s} - 2)/n)$, where s is a damping parameter that depends only on T and n. We use

$$s = \sqrt{2\log\left(1 + \frac{n\log n}{T}\right)},$$

and let $\lambda = 10\sqrt{T\log n/n}$ be the target maximum envy that the algorithm allows.

Define the potential function at time t as $\phi(t) = \sum_{i,j\in[n]:i\neq j} \phi_{ij}(t)$, where for $i,j \in [n]$,

$$\phi_{ij}(t) = C^{T-t} \cdot \exp\left(s\left(f_{ij}(t) - \lambda\right)\right).$$

The Proofs

Proof of Lemma 2.7. For all $\ell \in [t]$, item ℓ has been allocated in order to minimize $\phi(\ell)$. It suffices to show that at any time $t \leq T$, for any pair of agents i, j, with $\lambda = 10\sqrt{T \log n / n}$,

$$\Pr\left[f_{ij}(t) + \sum_{\ell=t+1}^{T} X_\ell^{ij} v_{i\ell} > \lambda\right] \leq \phi_{ij}(t), \tag{1}$$

where X_ℓ^{ij} is a random variable that takes values -1 and 1 with probability $1/n$ each, and it takes value 0 with probability $1 - 2/n$. Notice that $\textsc{Envy}_T^{ij}(\mathcal{A}^t) = \max\{f_{ij}(t) + \sum_{\ell=t+1}^{T} X_\ell^{ij} v_{i\ell}, 0\}$; summing up over all pairs i, j proves the lemma. Equation (1) follows from

$$\begin{aligned}
\Pr\left[f_{ij}(t) + \sum_{\ell=t+1}^{T} X_\ell^{ij} v_{i\ell} > \lambda\right] &= \Pr\left[e^{s\left(f_{ij}(t) + \sum_{\ell=t+1}^{T} X_\ell^{ij} v_{i\ell}\right)} > e^{s\lambda}\right] \\
&\leq e^{-s\lambda} \cdot \mathbb{E}\left[e^{s\left(f_{ij}(t) + \sum_{\ell=t+1}^{T} X_\ell^{ij} v_{i\ell}\right)}\right] && \text{(Markov's ineq.)} \\
&= e^{s(f_{ij}(t) - \lambda)} \cdot \mathbb{E}\left[\prod_{\ell=t+1}^{T} e^{s X_\ell^{ij} v_{i\ell}}\right] \\
&= e^{s(f_{ij}(t) - \lambda)} \prod_{\ell=t+1}^{T} \mathbb{E}\left[e^{s X_\ell^{ij} v_{i\ell}}\right] && \text{(independence)} \\
&= e^{s(f_{ij}(t) - \lambda)} \prod_{\ell=t+1}^{T} \left(1 - \frac{2}{n} + \frac{e^{s v_{i\ell}}}{n} + \frac{e^{-s v_{i\ell}}}{n}\right) \\
&\leq e^{s(f_{ij}(t) - \lambda)} \prod_{\ell=t+1}^{T} \left(1 - \frac{2}{n} + \frac{e^{s}}{n} + \frac{e^{-s}}{n}\right) \\
&= e^{s(f_{ij}(t) - \lambda)} \left(1 + \frac{e^s + e^{-s} - 2}{n}\right)^{T-t} \\
&= \phi_{ij}(t).
\end{aligned}$$

The second inequality follows from the fact that $e^x + e^{-x}$ is nondecreasing for $x \geq 0$. □

Proof of Lemma 2.8. Denote with $\phi_k(t+1)$ the potential function after giving item $t+1$ to agent k. We show that $(1/n)\sum_{k \in [n]} \phi_k(t+1) \leq \phi(t)$, which implies the desired result, as by definition of $\mathcal{A}^*$ we have $\phi(t+1) = \min_{k \in [n]} \phi_k(t+1)$. Recall that, for distinct $i, j, k \in [n]$, y_{tk}^{ij} takes values -1, 1, and 0 depending on whether item t was allocated to agent i, j, or k. Thus, $f_{ij}(t+1) = f_{ij}(t) + y_{t+1,k}^{ij} v_{i,t+1}$.

$$\begin{aligned}
\frac{1}{n}\sum_{k \in [n]} \phi_k(t+1) &= \frac{1}{n}\sum_{k \in [n]} \sum_{i,j \in [n]: i \neq j} e^{s\left(f_{ij}(t) + y_{t+1,k}^{ij} v_{i,t+1} - \lambda\right)} C^{T-(t+1)} \\
&= \frac{1}{n} C^{T-(t+1)} \sum_{i,j \in [n]: i \neq j} e^{s\left(f_{ij}(t) - \lambda\right)} \sum_{k \in [n]} e^{s y_{t+1,k}^{ij} v_{i,t+1}} \\
&= \frac{1}{n} C^{T-(t+1)} \sum_{i,j \in [n]: i \neq j} e^{s\left(f_{ij}(t) - \lambda\right)} \left(e^{s \cdot (1) \cdot v_{i,t+1}} + e^{s \cdot (-1) \cdot v_{i,t+1}} + \sum_{k \in [n] \setminus \{i,j\}} e^{s \cdot (0) \cdot v_{i,t+1}}\right)
\end{aligned}$$

$$= \frac{1}{n} C^{T-(t+1)} \sum_{i,j\in[n]:i\neq j} e^{s\left(f_{ij}(t)-\lambda\right)} \left(e^{s v_{i,t+1}} + e^{-s v_{i,t+1}} + n - 2\right)$$
$$\leq \frac{1}{n} C^{T-(t+1)} \sum_{i,j\in[n]:i\neq j} e^{s\left(f_{ij}(t)-\lambda\right)} \left(e^{s} + e^{-s} + n - 2\right)$$
$$= C^{T-(t+1)} \sum_{i,j\in[n]:i\neq j} e^{s\left(f_{ij}(t)-\lambda\right)} \cdot C \;=\; \phi(t). \qquad \square$$

Proof of Lemma 2.9.
$$\phi(0) = \sum_{i,j\in[n]:i\neq j} \phi_{ij}(0) = \sum_{i,j\in[n]:i\neq j} C^T e^{s f_{ij}(0) - s\lambda} < n^2 C^T e^{-s\lambda} = e^{-s\lambda + 2\log n + T\log C}.$$
We want the above expression to be less than one, or equivalently $s\lambda - 2\log n - T\log C > 0$. Using $1 + x \leq e^x$ implies that
$$C = 1 + \frac{e^s + e^{-s} - 2}{n} \leq e^{(e^s + e^{-s} - 2)/n} = e^{2(\cosh(s)-1)/n}.$$
Furthermore, $\cosh(x) \leq \exp\left(x^2/2\right)$, so that $C \leq \exp\left(2(\exp\left(s^2/2\right) - 1)/n\right)$. Therefore,
$$\begin{aligned} s\lambda - 2\log n - T\log C &\geq s\lambda - 2\log n - \frac{2T}{n}\left(e^{s^2/2} - 1\right) \\ &= 10\sqrt{2\log\left(1 + \frac{n\log n}{T}\right)\frac{T\log n}{n}} - 2\log n - \frac{2T}{n}\left(\frac{n\log n}{T}\right) \\ &= \left(\frac{5}{\sqrt{2}}\sqrt{\frac{T}{n\log n}\log\left(1 + \frac{n\log n}{T}\right)} - 1\right) 4\log n. \end{aligned}$$
We factored out $4\log n$ for convenience; it remains to show the parenthetical expression is positive. The function $\sqrt{x\log(1 + 1/x)}$ is increasing for all $x \geq 0$. Set $x = T/(n\log n)$, and note that the assumption $T \geq n\log n$ implies $x \geq 1$. Observing that $5\sqrt{\log(2)/2} > 1$ completes the proof. $\square$

2.3 Lower Bound

In this section, we show that an adversary can guarantee $\text{Envy}_T \in \Omega((T/n)^{r/2})$ for any $r < 1$. It follows that the deterministic algorithm we presented in Section 2.1 is optimal (up to a logarithmic factor). We first prove the bound for $n = 2$, followed by the case of an arbitrary number of agents.

Lower Bound for Two Agents

Lemma 2.10. *For $n = 2$ and any $r < 1$, there exists an adversary strategy for setting item values such that any algorithm must have $\text{Envy}_T \in \Omega(T^{r/2})$.*

Proof. Label the agents L and R, and let $\{v_0 = 1, v_1, v_2, \ldots\}$ be a decreasing sequence of values that we specify later, satisfying $v_d - v_{d+1} < v_{d'} - v_{d'+1}$ for all $d' < d$. The adversary keeps track of the *state* of the game, and the current state defines its strategy for choosing the agents' valuations. The adversary strategy that implies the lower bound is illustrated in Figure 1. Start in state 0, which we will also refer to as L_0 and R_0, for which the adversary sets the value of the arriving item as $(1, 1)$. To the left of state 0 are states labeled $L_1, L_2, \ldots$; in state L_d, the item that arrives has value $(1, v_d)$. To the right of state 0 are states labeled $R_1, R_2, \ldots$; in state R_d, an item will arrive with value $(v_d, 1)$. Whenever the algorithm allocates an item to agent L (resp. R), which we will refer to as making an L (resp. R) step, the adversary moves one state to the left (resp. right) to determine the value of the next item.

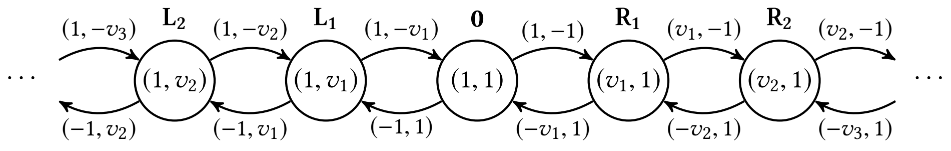

Fig. 1. Adversary strategy for two-agent lower bound. In state L_d, an item valued $(1, v_d)$ arrives, while in state R_d, an item valued $(v_d, 1)$ arrives. The arrows indicate whether agent L or agent R is given the item in each state. The arrows are labeled by the amount envy changes after that item is allocated.

We construct the optimal allocation algorithm against this adversary, and show that for this algorithm the envy at some time step $t \in [T]$ will be at least $\Omega(T^{r/2})$ for the given $r < 1$. This immediately implies Lemma 2.10: if the envy is sufficiently large at some time step t the adversary can guarantee the same envy at time T by making all future items valued at zero by both agents.

The intuition for the adversary strategy we have defined is that it forces the algorithm to avoid entering state L_d or R_d for high d, as otherwise the envy of some agent will grow to $v_0 + v_1 + \cdots + v_d$, which will be large by our choice of $\{v_d\}$. At the same time, if an L step is taken at state L_d, followed by a later return to state L_d, the envy of R increases by at least $v_d - v_{d+1}$; we choose $\{v_d\}$ so that this increase in envy is large enough to ensure that any algorithm which spends too many time steps close to state 0 incurs a large cost.

By the pigeonhole principle, either the states to the left or to the right of state 0 are visited for at least half the time. For the rest of this section, we assume, without loss of generality, that our optimal algorithm spends time $T' = \lceil T/2 \rceil$ in the "left" states $(L_0, L_1, \ldots)$, and that T' is an even number. We prove that the envy of agent R grows large at some time step t. We ignore any time the algorithm spends in the states R_d, $d \geq 1$. To see why this is without loss of generality, consider first a cycle spent in the right states that starts at R_0 with an item allocated to R and eventually returns to R_0. In such a cycle, an equal number of items are allocated to both agents. All of these items have value 1 to agent R, yielding a net effect of 0 on agent R's envy.[4] The other case is when the algorithm starts at R_0 but does not return to R_0. This scenario can only occur once, which means that the algorithm has already taken T' steps on the left side; the allocation of these items does not affect our proof.

Let K be an integer such that $K \leq \sqrt{T'/2}$, which we will show is without loss of generality. Denote by $\mathrm{OPT}(K)$ the set of envy-minimizing allocation algorithms that spend the T' steps in states $L_0, \ldots, L_K$ (and reach L_K). Note that the algorithm aims to minimize the maximum envy at any point in its execution. Let $\mathcal{A}^*(K)$ be the following algorithm, starting at L_0: Allocate the first K items to agent L, thus arriving at state L_K. For the next $T' - 2K$ items, alternate between allocating to agents R and L, thereby alternating between states L_{K-1} and L_K. Allocate the remaining K items to agent R. We show $\mathcal{A}^*(K)$ belongs to $\mathrm{OPT}(K)$.

Lemma 2.11. $\mathcal{A}^*(K) \in \mathrm{OPT}(K)$.

Proof. An algorithm that starts at state 0 and spends T' steps in the left states can be described as a sequence of choices $s_t \in \{L, R\}$ for $t \in [T']$ such that $s_1 = L$, and at every $t \in [T']$, agent L has received at least as many of the first t items as agent R (to avoid entering the right states). We refer to the state at time t as the state *after* the algorithm choice s_t.

[4]We ignore agent L completely, as our analysis is of the envy of agent R.

Consider any $\mathcal{A}(K) \in \mathsf{OPT}(K)$. We show that the corresponding sequence of allocations satisfy: (1) at time T' the state is L_0, so agent L receives the same number of items as agent R; and (2) there is exactly one R move at states $L_1, \ldots, L_{K-1}$. This proves the lemma, since $\mathcal{A}^*(K)$ is the only algorithm that satisfies these two conditions. We utilize the fact that the envy of an allocation sequence can be calculated from the number of L and R moves in every state: at state L_d, an L move increases the envy of agent R by v_d while an R move decreases it by v_d.

We start with the first property: suppose that the state at time T' is not 0. Let t be the last index such that $s_t = L$. Allocating $s_t = R$ instead (and $s_\ell = R$ for the remaining steps $\ell > t$) reduces the envy of agent R without entering state R_1, a contradiction.

For the second property, it suffices to show that if $s_t = L$ and $s_{t+1} = R$, then it must be that at step t the state is L_{K-1} (and therefore at step $t+1$ the state is L_K). Assume this is not the case, and we have such a t where the algorithm is in state $L_{\widehat{K}-1}$, $\widehat{K} < K$. Let ℓ be a step in which the algorithm is in state L_{K-1}, which exists by the definition of $\mathcal{A}(K)$. Assume that $\ell > t+1$ (an analogous argument can be applied to the case that $\ell < t$). We divide T' into three phases: (1) the first $t-1$ items, (2) the next $\ell - (t+1)$ items, and (3) the last $T' - \ell + 2$ items and consider $s' = s_1, \ldots, s_{t-1}, s_{t+2}, \ldots, s_\ell, s_t, s_{t+1}, s_{\ell+1}, \ldots, s_{T'}$. Notice that s' is s, except the alternating allocations L then R are now made at state L_{K-1} instead of at $L_{\widehat{K}-1}$. By construction, sequence s' never goes past state L_K. We now prove that, using s', the envy decreases with respect to s at each time step after $t-1$, contradicting the assumption $\mathcal{A}(K) \in \mathsf{OPT}(K)$.

In phase (1), the envy is unchanged. For phase (2), when using $\mathcal{A}(K)$, the pair of moves s_t and s_{t+1} increases envy by $v_{\widehat{K}} - v_{\widehat{K}-1}$. Hence, in comparison, s' has that much less envy during each time step of phase (2). At the start of phase (3) in s', the alternating allocations are performed at state L_{K-1}, increasing envy (in s') by $v_{K-1} - v_K < v_{\widehat{K}} - v_{\widehat{K}+1}$. At all remaining steps in (3), the envy is smaller in s' (compared to s) by $(v_{\widehat{K}} - v_{\widehat{K}+1}) - (v_{K-1} - v_K)$. This completes the proof that $\mathcal{A}(K)$ must satisfy both properties; the lemma follows. □

We analyze the envy of $\mathcal{A}^*(K)$ as a function of K before optimizing K. Agent R's maximum envy is realized at step $T' - K$, right before the sequence of R moves. $\textsc{Envy}_{T'-K}$ has two terms: the envy accumulated to reach state L_K, and the envy from alternating R and L moves between states L_K and L_{K-1}, so

$$\textsc{Envy}_{T'-K} = \sum_{d=0}^{K-1} v_d + \frac{T' - 2K}{2} \cdot (v_{K-1} - v_K)\,.$$

Given $r < 1$, define $v_d = (d+1)^r - d^r$. Notice that $\sum_{d=0}^{K-1} v_d = K^r$. This validates the initial assumption that $K \le \sqrt{T'/2}$, as otherwise $\sum_{d=0}^{K-1} v_d \ge (T'/2)^{r/2} \in \Omega(T^{r/2})$.

Lemma 2.12. *$v_{K-1} - v_K \ge r(1-r)K^{r-2}$.*

Applying Lemma 2.12 and distributing terms yields

$$\textsc{Envy}_{T'-K} \ge K^r - r(1-r)K^{r-1} + \frac{T'}{2} r(1-r)K^{r-2} \ge \frac{1}{2}\left(K^r + T'r(1-r)K^{r-2}\right),$$

where the second inequality uses the fact that $r(1-r) \le 1/4 < 1/2$ and assumes $K > 1$ (otherwise the envy would be linear in T'). To optimize K, noting that the second derivative of the above bound is positive for $K \le \sqrt{T'/2}$, we find the critical point:

$$\frac{\partial}{\partial K}\left(K^r + T'r(1-r)K^{r-2}\right) = rK^{r-1} - T'r(1-r)(2-r)K^{r-3} = 0 \implies K = \sqrt{T'(1-r)(2-r)}.$$

Defining $C_1 = \sqrt{(1-r)(2-r)}$ and substituting into the bound on $\textsc{Envy}_{T'-K}$, we obtain

$$\textsc{Envy}_{T'-K} \geq \frac{1}{2}\left(C_1^r (T')^{r/2} + T' r(1-r) C_1^{r-2} (T')^{r/2-1}\right) \in \Omega(T^{r/2}).$$

This completes the proof of Lemma 2.10. □

Lower Bound for Any Number of Agents

THEOREM 2.13. *For any $n \geq 2$ and $r < 1$, there exists an adversary strategy for setting item values such that any algorithm must have $\textsc{Envy}_T \in \Omega((T/n)^{r/2})$.*

PROOF. We augment the instance of Figure 1 in the following way. In addition to the first two agents, L and R, we have $n-2$ other agents. Each of these other agents will not value *any* of the items that arrive; hence, the nonzero values remain the same as before. State transitions work as follows. If the algorithm allocates an item to agent L or agent R, the transitions are the same as when $n = 2$. Otherwise, the adversary will remain in the same state.

Let T_0 be the number of items allocated to either agent L or R. We break the analysis into two cases. First, if $T_0 \in \Omega(T/n)$, then, $\textsc{Envy}_T \in \Omega((T/n)^{r/2})$ by the analysis of Lemma 2.10. Otherwise, $T_0 \in o(T/n)$ and therefore $T - T_0 \in \Theta(T)$, i.e., agents 3 through n receive many items. This implies that there exists an agent $i \in [3, n]$ that is allocated $\Omega(T/n)$ items. Without loss of generality, at least half these items were allocated in the left states, in which agent L values each item at 1, so that agent L has $\Omega(T/n)$ value for the items received by agent i. The value of agent L for her own allocation is at most $O(T_0)$, i.e., $o(T/n)$. Therefore, the envy of agent L for agent i is at least $\Theta(T/n) - o(T/n) \in \Theta(T/n)$. □

3 WHEN ITEMS ARRIVE IN BATCHES

In this section, we study the more general setting where items arrive in batches of size m, and the values of all items in a batch are revealed simultaneously. We assume m divides T for convenience.

3.1 Upper Bound

We use the following result from the literature on the division of *divisible* goods.

LEMMA 3.1 (STROMQUIST 18). *Suppose n agents have valuation functions over the interval $[0, 1]$, such that an agent's value for a subinterval is the integral of her value density function. Then there exists an envy-free division of the interval in which every agent receives a single contiguous interval.*

It will be convenient to think of the n contiguous allocations as created by $n-1$ cuts on the interval $[0, 1]$. In the context of m indivisible goods with additive valuations, this result implies that, if the items are placed on a line (in any order), there exists a fractional envy-free allocation in which no agent receives more than 2 fractional items. Every item corresponds to an interval of size $1/m$, and every agent's valuation in that interval is constant and proportional to her valuation for that item. Given the solution guaranteed to exist by Lemma 3.1, every agent's allocation is between at most two cuts and therefore contains no more than 2 fractional items.[5] Such a near-integral envy-free allocation is useful, since any integral allocation found by randomized rounding is guaranteed to have small envy *ex post*, as the following lemma shows.

LEMMA 3.2. *Given m items, there exists an envy-free fractional allocation $A = A_1, \ldots, A_n$, such that every agent receives at most 2 fractional items. Furthermore, if $x_{i\ell} \in [0, 1]$ is the fraction of item ℓ allocated to agent i, then randomly giving each item ℓ to each agent i with probability $x_{i\ell}$ results in an integral allocation A' where for all $i, j \in [n]$, $v_i(A'_i) \geq v_i(A'_j) - 4$.*

[5]In fact only the agents who get the first and the last contiguous interval are adjacent to one cut; all other agents' allocations are between two cuts.

Proof. The first part of the statement, that there exists an allocation A that is envy-free and in which each agent receives at most 2 fractional items, follows from the previous discussion. For the second part, notice that the worst-case scenario for an agent i is to not get either of the fractional items allocated to her in A. Furthermore, some other agent j might get both of her fractional items from A. In this scenario, the envy of agent i for agent j is maximized and is at most 4 (since the value for every item is at most 1). □

In Section 2, before giving a deterministic algorithm, we first analyzed the performance of the random allocation algorithm. Crucially, our analysis characterized the optimal strategy for an adaptive adversary against the random allocation algorithm, and we showed that this strategy is in fact nonadaptive. This allowed us to use standard concentration inequalities. Proving such a characterization is much trickier in the batch case. Fortunately, we can bypass this step and directly "derandomize" the algorithm that at each step outputs the (randomly-rounded) allocation of Lemma 3.2. Our main result for this setting is the following:

Theorem 3.3. *Suppose that $T \geq m \log n$. Then there exists a deterministic algorithm that achieves $\textsc{Envy}_{T,m} \in O(\sqrt{T \log n/m})$.*

Again, the assumption of $T \geq m \log n$ is very weak, otherwise there are at most $T/m \leq \log n$ batches, and we can use an EF1 algorithm in each to achieve $\textsc{Envy}_{T,m} \leq \log n$.

The Algorithm

We define a potential function $\phi(t)$ that depends on n, T, the values of the items in the first t batches, as well as their allocations. When batch $t+1$ arrives, we first find the near-integral envy-free allocation A^{t+1} (of the items in batch $t+1$) guaranteed to exist by Lemma 3.2 (we address computation below). This fractional allocation is then rounded to an integral allocation in a way that $\phi(t+1)$ is minimized. Call this algorithm $\mathcal{A}^*$. Since our algorithm is deterministic, an adversary that wants to maximize $\textsc{Envy}_{T,m}$ does not gain from being adaptive. Therefore, there exists some optimal (for the adversary) choice of values for items 1 through T. We analyze our algorithm for an arbitrary choice of item values.

Similarly to our algorithm from Section 2.2, we rely on three properties of ϕ. Given $t \in [T/m]$, let $\mathcal{A}^t$ be the algorithm that rounds A^ℓ (the allocation in batch ℓ) in a way that $\phi(\ell)$ is minimized, for all $\ell \in [1, t]$, and rounds the remaining A^ℓ for $\ell = t+1, \dots, T/m$ randomly. Let $\textsc{Envy}^{ij}_{T,m}(\mathcal{A}^t)$ be the envy of agent i for agent j at the end of the execution of $\mathcal{A}^t$.

Lemma 3.4. $\phi(t) \geq \sum_{i,j\in[n]} \Pr\left[\textsc{Envy}^{ij}_T(\mathcal{A}^t) > 100\sqrt{T \log n/m}\right]$.

Lemma 3.5. *For all $t \in [T/m - 1]$, $\phi(t+1) \leq \phi(t)$.*

Lemma 3.6. *For $T \geq m \log n$, $\phi(0) < 1$.*

Proof of Theorem 3.3. Notice that $\mathcal{A}^{T/m}$ is exactly the same as the algorithm $\mathcal{A}^*$. Lemmas 3.5 and 3.6 imply that $\phi(T) < 1$. Combining with Lemma 3.4, we get that for any item valuations,

$$\Pr\left[\exists i,j \in [n] : \textsc{Envy}^{ij}_T(\mathcal{A}^*) > 100\sqrt{\frac{T \log n}{m}}\right] \leq \sum_{i,j\in[n]} \Pr\left[\textsc{Envy}^{ij}_T(\mathcal{A}^*) > 100\sqrt{\frac{T \log n}{m}}\right] \leq \phi(T) < 1,$$

Since $\mathcal{A}^*$ is deterministic, the inequality above implies that there is no $i,j \in [n]$ such that $\textsc{Envy}^{ij}_T > 100\sqrt{T \log n/m}$, and we conclude that $\textsc{Envy}_{T,m} \leq 100\sqrt{T \log n/m} \in O(\sqrt{T \log n/m})$. □

Setup

In batch t, define $A^t = A_1^t, \ldots, A_n^t$ as the envy-free fractional solution of Lemma 3.2, in which no agent receives more than 2 fractional items. Let $\hat{A}^t = \hat{A}_1^t, \ldots, \hat{A}_n^t$ be an integral rounding of A^t; $\hat{A}^t$ is the actual allocation used in batch t. Define

$$\Delta_{ij}^t(\hat{A}^t) = \left(v_i(\hat{A}_j^t) - v_i(\hat{A}_i^t)\right) + \left(v_i(A_i^t) - v_i(A_j^t)\right)$$

and let $f_{ij}(t, \hat{A}^1, \ldots, \hat{A}^t) = \sum_{\ell=1}^t \Delta_{ij}^t(\hat{A}^t)$. To simplify notation, we write $f_{ij}(t)$ when the allocation is clear from context. Notice that $\text{Envy}_{T,m}^{ij} \leq f_{ij}(T)$; this is an inequality because Δ_{ij}^t is centered to have zero mean, while $v_i(\hat{A}_j^t) - v_i(\hat{A}_i^t)$ may have mean less than zero. Also, observe that $\hat{A}^t$ is not random. However, if we were to *randomly* round A^t to an integral allocation $\hat{B}^t$, then the resulting random variable $\Delta_{ij}^t(\hat{B}^t)$ has zero mean and satisfies $|\Delta_{ij}^t(\hat{B}^t)| \leq 4$, by Lemma 3.2.

Let $\lambda = 100\sqrt{T \log n/m}$ and $s = \frac{1}{4}\log(1 + \frac{\lambda m}{4T})$. For $i, j \in [n]$, define the potential function at time t for i with respect to j as

$$\phi_{ij}(t) = \exp\left(s f_{ij}(t) - s\lambda + \left(\frac{T}{m} - t\right)\left(e^{4s} - 4s - 1\right)\right),$$

and define the overall potential function as $\phi(t) = \sum_{i,j \in [n]: i \neq j} \phi_{ij}(t)$.

The Proofs

The proof of Lemma 3.4 is relegated to the full version of this paper. In its proof, as in the proof of Lemma 3.5, we use the following property of bounded, centered random variables (the proof of which may also be found in the full version of this paper).

Lemma 3.7. *Let X be a random variable with $\mathbb{E}[X] = 0$ and $|X| \leq 4$. Then for all $v \in [0,1]$ it holds that $\mathbb{E}\left[e^{sXv}\right] \leq \exp\left(e^{4s} - 4s - 1\right)$.*

Proof of Lemma 3.5. We prove that there exists a rounding $\hat{A}^*$ of the fractional allocation A^{t+1} of batch $t+1$ so that allocating according to $\hat{A}^*$ results in $\phi(t+1) \leq \phi(t)$. Let $x_{i\ell}^{t+1}$ be the fraction of item ℓ in batch $t+1$ allocated to agent i in A^{t+1}. We show that allocating every item ℓ to agent i with probability $x_{i\ell}^{t+1}$ makes the expected value of $\phi(t+1)$ at most $\phi(t)$. We can immediately conclude that there exists an integral allocation for which $\phi(t+1) \leq \phi(t)$.

Let $\hat{B}^{t+1}$ be a possible (rounded) integral allocation, with corresponding probability $p(\hat{B}^{t+1})$, and let D be the distribution where allocation $\hat{B}^{t+1}$ appears with probability $p(\hat{B}^{t+1})$. Finally, let $\phi_{\hat{B}^{t+1}}(t+1)$ be the value of the potential function after allocating batch $t+1$ according to $\hat{B}^{t+1}$. Note that $f_{ij}(t+1, \hat{A}^1, \ldots, \hat{A}^t, \hat{B}^{t+1}) = f_{ij}(t) + \Delta_{ij}^{t+1}(\hat{B}^{t+1})$.

$$\begin{aligned}
\mathbb{E}_{\hat{B}^{t+1} \sim D}\left[\phi_{\hat{B}^{t+1}}(t+1)\right] &= \sum_{\hat{B}^{t+1}} p(\hat{B}^{t+1}) \cdot \left(e^{-s\lambda} e^{(\frac{T}{m} - t - 1)(e^{4s} - 4s - 1)} \sum_{i,j \in [n]: i \neq j} e^{s f_{ij}(t+1, \hat{A}^1, \ldots, \hat{A}^t, \hat{B}^{t+1})}\right) \\
&= \sum_{\hat{B}^{t+1}} p(\hat{B}^{t+1}) \cdot \left(e^{-s\lambda} e^{(\frac{T}{m} - t - 1)(e^{4s} - 4s - 1)} \sum_{i,j \in [n]: i \neq j} e^{s f_{ij}(t) + s\Delta_{ij}^{t+1}(\hat{B}^{t+1})}\right) \\
&= e^{-s\lambda} e^{(\frac{T}{m} - t - 1)(e^{4s} - 4s - 1)} \sum_{i,j \in [n]: i \neq j} \left(e^{s f_{ij}(t)} \sum_{\hat{B}^{t+1}} p(\hat{B}^{t+1})\, e^{s\Delta_{ij}^{t+1}(\hat{B}^{t+1})}\right) \\
&\leq e^{-s\lambda} e^{(\frac{T}{m} - t - 1)(e^{4s} - 4s - 1)} \sum_{i,j \in [n]: i \neq j} \left(e^{s f_{ij}(t)}\, \mathbb{E}_{\hat{B}^{t+1} \sim D}\left[e^{s\Delta_{ij}^{t+1}(\hat{B}^{t+1})}\right]\right).
\end{aligned}$$

We notice that $\Delta_{ij}^{t+1}(\hat{B}^{t+1})$ is a random variable (since $\hat{B}^{t+1}$ is random) that satisfies the conditions of Lemma 3.7, so we get

$$\begin{aligned}\mathbb{E}_{\hat{B}^{t+1}\sim D}\left[\phi_{\hat{B}^{t+1}}(t+1)\right] &\leq e^{-s\lambda}e^{(\frac{T}{m}-t-1)(e^{4s}-4s-1)}\sum_{i,j\in[n]:i\neq j} e^{sf_{ij}(t)}e^{e^{4s}-4s-1}\\ &= e^{-s\lambda}e^{(\frac{T}{m}-t)(e^{4s}-4s-1)}\sum_{i,j\in[n]:i\neq j} e^{sf_{ij}(t)} = \phi(t).\end{aligned}$$ □

Proof of Lemma 3.6.

$$\begin{aligned}\phi(0) &= \sum_{i,j\in[n]:i\neq j}\phi_{ij}(0)\\ &= \sum_{i,j\in[n]:i\neq j}\exp\left(sf_{ij}(0)-s\lambda+\frac{T}{m}\left(e^{4s}-4s-1\right)\right)\\ &< n^2\exp\left(-s\lambda+\frac{T}{m}\left(e^{4s}-4s-1\right)\right)\\ &= n^2\exp\left(-\frac{T}{m}\left(1+4s+s\frac{\lambda m}{T}-e^{4s}\right)\right)\\ &= n^2\exp\left(-\frac{T}{m}\left(1+4s\left(1+\frac{\lambda m}{4T}\right)-e^{4s}\right)\right)\\ &= n^2\exp\left(-\frac{T}{m}\left(1+\left(1+\frac{\lambda m}{4T}\right)\log\left(1+\frac{\lambda m}{4T}\right)-\left(1+\frac{\lambda m}{4T}\right)\right)\right)\\ &= n^2\exp\left(-\frac{T}{m}\left((1+x)\log(1+x)-x\right)\right),\end{aligned}$$

where $x=\frac{\lambda m}{4T}$. The function $h(x)=(1+x)\log(1+x)-x$ satisfies $h(x)\geq x^2/(2+2x/3)$. Therefore,

$$\phi(0) < n^2\exp\left(-\frac{T}{m}\left(\frac{(\frac{\lambda m}{4T})^2}{2+\frac{2(\frac{\lambda m}{4T})}{3}}\right)\right) = n^2\exp\left(-\frac{3m\lambda^2}{96T+8\lambda m}\right) = \exp\left(2\log n-\frac{3m\lambda^2}{96T+8\lambda m}\right).$$

Substituting in $\lambda=100\sqrt{T\log n/m}$ gives: $\phi(0)\leq\exp\left(2\log n-\frac{30000T\log n}{96T+800\sqrt{Tm\log n}}\right)$, which is strictly less than 1 for $T\geq m\log n$. □

Computational Issues

Lemma 3.2, just like Lemma 3.1, is existential and leaves unanswered the question of how to find the nearly-integral envy-free allocation for every batch. We partially address this, at least from a practical point of view, by formulating a *mixed integer program (MIP)* to compute such an allocation.

Let $x_{i\ell}$ be the fraction of item ℓ given to agent i. Binary variables $x_{i\ell}^0$ and $x_{i\ell}^1$ will sum to 0 when $x_{i\ell}$ is fractional, and sum to 1 otherwise. Lemma 3.1 implies that the following MIP is feasible:

$$\sum_{\ell=1}^{m} v_{i\ell}(x_{i\ell}-x_{j\ell})\geq 0, \quad \text{for all } i,j\in[n] \tag{2}$$

$$\sum_{i=1}^{n} x_{i\ell}=1, \quad \text{for all } \ell\in[m] \tag{3}$$

$$x^0_{i\ell} \leq x_{i\ell} \leq 1 - x^1_{i\ell}, \qquad \text{for all } i \in [n], \ell \in [m] \tag{4}$$

$$\sum_{\ell=1}^{m} (x^0_{i\ell} + x^1_{i\ell}) \geq m - 2, \qquad \text{for all } i \in [n] \tag{5}$$

$$x_{i\ell} \in [0,1], \qquad \text{for all } i \in [n], \ell \in [m] \tag{6}$$

$$x^0_{i\ell}, x^1_{i\ell} \in \{0,1\}, \qquad \text{for all } i \in [n], \ell \in [m] \tag{7}$$

Constraint (2) ensures that the allocation is envy free, while Constraint (3) ensures every item is fully allocated. Constraint (4) ensures that $x^0_{i\ell}$ and $x^1_{i\ell}$ sum to 0 when $x_{i\ell}$ is fractional, and sum to 1 otherwise (using the fact that these variables are binary, by Constraint (7)). Constraint (5) guarantees at most 2 fractional items per agent. These constraints may be coupled with any objective function to find a near-integral fractional solution.

Unfortunately, solving a MIP is unlikely to be computationally efficient in general. Furthermore, known hardness results for related problems [8] suggest that producing an envy-free (or approximately envy-free) and contiguous fractional allocation in our setting might be a PPAD-hard problem. Such a hardness result does not rule out a polynomial time algorithm for finding an allocation with the properties of Lemma 3.2, i.e., a fractional envy-free allocation where each agent gets at most 2 fractional items.[6] In fact, we present some tractable special cases in full version of this paper. The general problem is left open.

Another step that may seem problematic (from a computational viewpoint) is rounding the fractional allocation in a way that minimizes the potential function. However, since the potential function is convex in the allocation, this can be done efficiently.

3.2 Lower Bound

Our last result is a lower bound for the batch setting, which is asymptotically tight in T/m, but, unlike the one-by-one setting, does leave a gap in terms of the dependence on the number of agents.

Theorem 3.8. *For any $n \geq 2$ and $r < 1$, there exists an adversary strategy for setting item values such that any algorithm must have* $\text{Envy}_T \in \Omega((\frac{T}{mn})^{r/2})$.

Proof. The theorem follows almost directly from Theorem 2.13. Indeed, assume that in each batch there are $m-1$ items that are worthless to all agents. In this case the batch setting reduces to the one-by-one setting, and we obtain the lower bound given by Theorem 2.13, with a total number of items equal to the number of batches, i.e., $T' = T/m$. □

4 DISCUSSION

We finish with a discussion of several pertinent issues that have not been addressed so far.

Additivity assumption. We have assumed that agents have additive valuations for bundles of items. This common assumption is typically considered strong. But for the purpose of defining envy in our online setting we consider it to be very natural. Indeed, in an online setting, the allocated items would typically be used independently of each other. Consequently, we can interpret the envy of i for j, $\sum_{t\in A_i} v_{it} - \sum_{t\in A_j} v_{it}$, as $\sum_{t=1}^{T} v_{it}(\mathbb{I}_{t\in A_i} - \mathbb{I}_{t\in A_j})$. Notice that this is a sum over per-round envy. In other words, the additivity assumption actually amounts to envy being additive over time.

The partial information model. In our model the values of agents for the current item (or batch of items) are revealed before the item is allocated, and inform that decision. One can imagine a natural variant — the *partial information model* — where the values are only revealed *after* the item

[6] For our upper bound to work, even a constant number of fractional items suffices.

has been allocated. Notice that the randomized upper bound of Theorem 2.1 carries over, because the algorithm ignores the values shown to it. So does the lower bound of Theorem 2.13, because it holds *even* against more powerful algorithms. However, the game between the algorithm and the adversary is now an extensive-form game of *incomplete* information, where randomization can potentially help. This turns out to be the case, and, in fact, deterministic algorithms cannot have vanishing envy in this setting. For more details, see the full version of this paper.

Is low envy fair enough? We have focused with single-minded determination on a single goal — that of minimizing envy. A possible concern is that low envy, in and of itself, is not sufficient to lead to intuitively fair outcomes, as has been observed in various contexts [7, 10]. Be that as it may, even if one is interested in a combination of low envy and other properties (Pareto efficiency comes to mind), our results establish a baseline for what one could hope for, and are therefore a crucial first step in any such investigation.

ACKNOWLEDGMENTS

We are grateful to Noga Alon for crucial pointers to the literature on derandomization and vector balancing games, and to Hervé Moulin for his help with Lemma 3.2.

REFERENCES

[1] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. 2015. Online Fair Division: Analysing a Food Bank Problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2540–2546.

[2] M. Aleksandrov and T. Walsh. 2017. Pure Nash Equilibria in Online Fair Division. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 42–48.

[3] S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Wigderson. 1994. On the Power of Randomization in On-line Algorithms. *Algorithmica* 11, 1 (1994), 2–14.

[4] S. Bernstein. 1946. *The Theory of Probabilities.* Gastehizdat Publishing House.

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

[6] S. Bubeck and N. Cesa-Bianchi. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning* 5, 1 (2012), 1–122.

[7] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. 2016. The Unreasonable Fairness of Maximum Nash Product. In *Proceedings of the 17th ACM Conference on Economics and Computation (EC)*. 305–322.

[8] X. Deng, Q. Qi, and A. Saberi. 2012. Algorithmic Solutions for Envy-Free Cake Cutting. *Operations Research* 60, 6 (2012), 1461–1476.

[9] E. Friedman, C.-A. Psomas, and S. Vardi. 2017. Controlled Dynamic Fair Division. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*. 461–478.

[10] Y. Gal, M. Mash, A. D. Procaccia, and Y. Zick. 2017. Which Is the Fairest (Rent Division) of Them All? *Journal of the ACM* 64, 6 (2017), article 39.

[11] I. Kash, A. D. Procaccia, and N. Shah. 2014. No Agent Left Behind: Dynamic Fair Division of Multiple Resources. *Journal of Artificial Intelligence Research* 51 (2014), 579–603.

[12] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. 2004. On Approximately Fair Allocations of Indivisible Goods. In *Proceedings of the 6th ACM Conference on Economics and Computation (EC)*. 125–131.

[13] D. C. Parkes, A. D. Procaccia, and N. Shah. 2015. Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities. *ACM Transactions on Economics and Computation* 3, 1 (2015), article 3.

[14] A. D. Procaccia. 2016. Cake Cutting Algorithms. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endress, J. Lang, and A. D. Procaccia (Eds.). Cambridge University Press, Chapter 13.

[15] P. Raghavan. 1988. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *J. Comput. System Sci.* 37, 2 (1988), 130–143.

[16] P. Sanders. 1996. On the Competitive Analysis of Randomized Static Load Balancing. In *Proceedings of the 1st Workshop on Randomized Parallel Algorithms (RANDOM).*

[17] J. Spencer. 1977. Balancing Games. *Journal of Combinatorial Theory Series B* 23, 1 (1977), 68–74.

[18] W. Stromquist. 1980. How to Cut a Cake Fairly. *Amer. Math. Monthly* 87, 8 (1980), 640–644.

[19] F. E. Su. 1999. Rental Harmony: Sperner's Lemma in Fair Division. *Amer. Math. Monthly* 106, 10 (1999), 930–942.

[20] T. Walsh. 2011. Online Cake Cutting. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*. 292–305.