# Online MMS with Worst-Case Guarantees: Randomization Beats the Adversary

ZHIYI HUANG[*], University of Texas, Austin, USA
POOJA KULKARNI[*], University of Illinois at Urbana-Champaign, USA
PARNIAN SHAHKAR[*], University of California, Irvine, USA
SHUCHI CHAWLA, University of Texas, Austin, USA
RUTA MEHTA, University of Illinois Urbana-Champaign, USA

We study the problem of fairly allocating $m$ indivisible goods among $n$ agents who arrive online, under the notion of maximin share (MMS) fairness. Fair allocation with online agent arrivals is known to be extremely challenging: prior work shows that even with two agents, no deterministic online algorithm can simultaneously guarantee for both agents any non-trivial approximation to MMS, or to any standard fairness notion, without knowledge of future arrivals.

In this work, we show that randomization can circumvent these strong impossibility results. We first present a randomized allocation algorithm parameterized by $c \in \Omega(\log n)$, that allocates goods independently in each round, and with probability at least $1 - \frac{(O(\log c))^3}{c^2}$ guarantees *every* agent an $O(1/c)$-approximation to MMS. We extend this guarantee all the way to submodular valuation functions, providing a strong counter to known hardness results.

Our second main result targets the stronger notion of constant-MMS guarantees. We propose an allocation scheme based on *dependent rounding* that achieves a $1/256$-approximation to MMS with constant probability under binary additive valuations. The analysis proceeds by carefully interleaving our algorithm, round by round, with an auxiliary algorithm. This coupling enables us to apply concentration bounds for exchangeable random variables, which in turn allows us to control the allocation probabilities.

We complement these results with almost matching limitations: for any $c \geq 1$, no online algorithm can achieve a $1/c$-approximation to MMS with probability greater than $1 - 1/c^2$, even for the case of binary additive valuations.

## Contents

[*]Equal Contribution.

*Manuscript submitted for review to the 27th ACM Conference on Economics & Computation (EC'26).*

## 1 Introduction

Consider a disaster relief supply center that distributes essential goods to members of an affected community. Recipients have diverse requirements. They arrive over time, communicate their needs, and receive allocations. The center must manage its limited supply carefully: allocating too generously to early arrivals risks shortages for those who come later, while overly conservative allocations can lead to inefficiency and unmet needs. Compounding this challenge, demand is difficult to forecast, as it depends sensitively on the severity and local impact of the disaster. In such uncertain, dynamic environments, is it nevertheless possible to design allocation policies that achieve meaningful guarantees of fairness and efficiency?

Motivated by this setting, we study an online allocation problem with $m$ heterogeneous, indivisible goods and $n$ heterogeneous agents with combinatorial valuation functions. Agents arrive sequentially over time; upon arrival, an agent receives an allocation and then departs, and all allocation decisions are irrevocable. The allocator knows the total number of agents $n$ in advance, but has no information about the valuation functions of future arrivals. The goal is to design an online allocation policy that guarantees a fair share to *every* agent, despite the uncertainty inherent in future demand.

Given the online nature of the problem, we focus on a notion of fairness that depends only on an agent's own valuation and allocation, rather than on allocations made to agents who arrive later. For indivisible goods, the most natural, as well as sought-after, such notion is the *maximin share (MMS)* [Budish, 2011]. Informally, the MMS of an agent is the maximum value she can guarantee for herself by partitioning all goods into $n$ bundles and receiving the least desirable bundle. Formally, let $v_i : 2^{[m]} \to \mathbb{R}_+$ denote the monotone valuation function of agent $i$ for $i \in [n]$. The MMS value of agent $i$, denoted $\mathsf{MMS}_i$, is defined as

$$\mathsf{MMS}_i = \max_{(P_1,\dots,P_n)\in\Pi([m])} \min_{\ell\in[n]} v_i(P_\ell),$$

where $\Pi([m])$ denotes the set of all partitions of the set of goods $[m]$ into $n$ bundles. Notably, an agent's MMS value depends only on her own valuation and the total number of agents $n$, making it particularly well suited to online settings with sequential agent arrivals.

Fair division, the problem of allocating scarce resources "fairly" among heterogeneous agents, has long been a central topic across computer science, economics, social choice theory, and operations research. Over the past several decades, it has been extensively studied in offline settings, where all agents and their valuations are known in advance (e.g., see latest surveys [Amanatidis et al., 2023, Moulin, 2019] and books [Brams and Taylor, 1996, Moulin, 2004]). More recently, motivated by practical applications such as job and task scheduling or disaster relief and food banks, attention has shifted to online settings in which goods or agents arrive over time (see Section 1.2). Despite this growing body of work, the notion of maximin share (MMS) has received relatively little attention in online models, in part due to inherent technical challenges [Amanatidis et al., 2022, Kulkarni et al., 2025]. In this paper, we overcome these barriers using carefully designed randomized algorithms and obtain nontrivial worst-case MMS guarantees for *all* agents, with high probability, in an online setting with adversarial agent arrivals.

It is well known that an exact MMS allocation–one in which every agent receives at least her full MMS value–need not exist even in offline settings [Feige et al., 2021, Procaccia and Wang, 2014]. This has led to the study of *approximate* MMS guarantees: an allocation $(A_1, \dots, A_n) \in \Pi(M)$ is said to be $\alpha$-MMS, for $\alpha \in [0, 1]$, if $v_i(A_i) \geq \alpha\mathsf{MMS}_i,\ \forall i \in [n]$. An extensive literature in the past decade has established constant-factor MMS approximations in offline settings for various classes of valuation functions, including additive [Akrami and Garg, 2024, Aziz et al., 2017, Barman and Krishnamurthy, 2020], submodular [Ghodsi et al., 2018, Uziahu and Feige, 2023], and their special cases (e.g., [Bouveret and Lemaître, 2016, Feige, 2022, Kulkarni et al., 2024, Shahkar and Garg, 2025]). In contrast, little is known in online settings. No MMS guarantees are known for online good arrivals; the main barrier is that MMS value is undefined until all the goods are seen. For online agent arrivals, prior work [Kulkarni et al., 2025] considers a stochastic model with additive valuations, achieving constant-factor MMS guarantees with high probability only when agents are drawn from a distribution over a fixed set of $k$ types known a priori. In the adversarial arrival model, that work shows a strong inapproximability beyond $\Omega\left(\frac{1}{\sqrt{k}}\right)$ when the adversary is adaptive.[1] This negative result naturally raises the following question.

*Is it possible to achieve any nontrivial approximate-MMS guarantee against an oblivious[2] adversary, when the allocator has no prior knowledge of the agents' valuation functions?*

In this paper, we make significant progress on the above question. First, we design a randomized algorithm based on independent good allocation, that ensures $\frac{1}{c}$-approximate MMS allocation to *all* agents with probability $\left(1 - O\left(\frac{(\log c)^3}{c^2}\right)\right)$, for $c \in \Omega(\log n)$. We extend this result all the way to general submodular valuations. Second, we complement this result with an almost matching lower bound: for any $c \geq 1$, no algorithm, randomized or not, can achieve $\frac{1}{c}$-approximate MMS for all agents simultaneously with probability better than $\left(1 - \frac{1}{c^2}\right)$. We note that our lower bound example holds even for the case of binary additive valuations.

In light of the counterexample and the importance of the binary additive case (e.g., [Aleksandrov et al., 2015, Barman et al., 2018, Brams and Fishburn, 1978, Halpern et al., 2020]), we next study the possibility of constant-MMS for the case of binary additive valuations. This problem indeed turns out to be quite tricky. The independent good allocation approach does not concentrate sufficiently to ensure constant-approximate MMS to all $n$ agents, even with constant probability. Instead, we carefully design a *dependent allocation* scheme that can be analyzed by coupling the algorithm's random variables with exchangeable random variables of a different algorithm. Together these allow us to apply concentration, and thereby get our final result. Namely, a randomized algorithm that ensures $\frac{1}{256}$-approximate MMS to all agents with constant probability. We note that all our algorithms run in polynomial-time, and are the first to achieve online MMS guarantees in adversarial setting.

In the next section, we give an overview of the main challenges, our high-level approach to overcome these, and highlight the nuances of our randomized rounding schemes and the proof techniques.

## 1.1 Technical Overview

We begin by recalling a fundamental challenge in fully-adversarial online fair division, illustrated by the following example from Kulkarni et al. [2025].

**Example 1.1.** Let $m$ be even. Suppose $m$ goods must be allocated among *two* agents who arrive sequentially. The first agent arrives reports an additive valuation with each individual good valued at 1. Her maximin share (MMS) is therefore $\frac{m}{2}$. To guarantee an $\alpha$-MMS allocation ex post for some $\alpha > 0$, the algorithm must allocate her at least $\lceil \alpha \cdot \frac{m}{2} \rceil$ goods, after which she departs with her bundle.

The second agent then arrives and reports that she values exactly two of the $m$ goods at 1, with zero value for all others; hence, her MMS value is 1. However, unless $\alpha \leq \frac{2}{m}$, it is possible that both of the goods she values are contained in the bundle allocated to the first agent. In this case, the

[1]An adaptive adversary can observe the random coin tosses of the algorithm.

[2]An oblivious adversary can select worst-case agents' valuation functions after knowing the algorithm, but *not* knowing the outcomes of the random coin tosses (decisions) made by the algorithm.

second agent derives zero value from any feasible allocation, and only a zero-MMS guarantee is achievable for her. □

This example demonstrates that in a fully adversarial setting, even with only two agents, the lack of knowledge of the second agent's value function can prevent any algorithm from guaranteeing a positive MMS approximation to both. To circumvent this impossibility, Kulkarni et al. [2025] adopt a learning-augmented approach, assuming that the algorithm has access to partial information about the preferences of future agents.

In contrast, we pursue a different route to overcoming the adversary: we leverage randomization. To illustrate the idea, reconsider the example above. Suppose the algorithm allocates a uniformly random subset of size $\frac{m}{2}$ to the first agent. If the adversary does not observe the realized allocation, then the probability that both goods valued by the second agent lie in the first agent's bundle is at most $\frac{1}{4}$. Consequently, with non-trivial probability, the algorithm can guarantee each agent her full MMS value in this instance.

*Randomized "rounding" for additive agents.* In order to extend the above idea to multiple agents, we imagine the allocation process as an "online rounding" of a trivially-fair fractional solution. Intuitively, for additive value agents, if fractional allocation was allowed, each agent would be happy receiving a $1/n$ fraction of each good. We therefore attempt to randomly allocate goods in such a manner that for each good $j \in [m]$ and agent $i \in [n]$, the marginal probability of allocating $j$ to $i$ is $\alpha/n$ for large enough $\alpha$.

One way to do so is to allocate each good *independently* with probability $\approx \alpha$ to each agent $i$. In particular, consider allocating to each arriving agent $i$ each remaining good with probability $1/2n$. Then, the total probability with which an good is allocated is at most $1/2$. So, in expectation, when agent $i$ arrives, the as-yet unallocated set of goods retains at least half her value. Then, in expectation, the agent receives at least a quarter of her MMS value from this set.

While this process allocates a constant-fraction of her MMS to every agent in expectation, we are interested in providing an *ex post* guarantee of approximate MMS with high probability to every agent simultaneously. Unfortunately, a roadblock arises: we can only tolerate a $1/\text{poly}(n)$ probability of failure for each agent, in order to be able to use a union bound over the $n$ agents. Consider, in particular, an agent who values $n$ of the $m$ goods at 1 each, and the remaining goods at 0. Then, even ignoring any previous allocations, if we allocate each good with probability $1/2n$ to this agent, with constant probability she receives no valuable good at all!

To handle this issue, we separately handle agents that place at least one good at high value (comparable to their MMS), and those that place every good at low value. For the former type, that we call *special* agents, we allocate exactly one high value good to them. For the latter type, we allocate each good independently with appropriate probability.

Our eventual goal is to argue that when an agent arrives, the set of goods remaining retains some constant fraction of her value with high probability, and of these she obtains an $\alpha/n$ fraction with high probability. The presence of special agents complicates this argument in two ways. First, we need to account for the reduction in future agents' value due to the special allocations, and accordingly increase their probability of receiving each individual remaining good. Second, we need to argue that special agents themselves are able to receive a high-value good by ensuring that with high probability at least one such good remains. We carefully balance allocation probabilities to resolve both issues.

Our first main result, Theorem 3.2, asserts that any fraction $c \in O(1/\log n)$ of MMS can be guaranteed simultaneously for all agents with probability at least $1 - O(c^2 \operatorname{poly}\log(1/c))$. We complement this result with a nearly matching lower bound, Theorem 3.1: for any $c > 0$, no algorithm can simultaneously guarantee $c$-MMS to all agents with probability more than $1 - c^2/2$.

*Extension to Submodular Valuations.* Our analysis of independent rounding for additive valuations extends to submodular valuations with only minor modifications. Under independent rounding, the agent's expected value is exactly the value of the corresponding fractional allocation evaluated under the multilinear extension. To relate this quantity to the agent's MMS value, we use the fact that the concave extension of an allocation in which each good is assigned to the agent with probability $\alpha$ is at least $\alpha \cdot \mathsf{MMS}$. For submodular valuations, it is well known that the multilinear extension is within a factor of $\left(1 - \frac{1}{e}\right)$ of the concave extension, i.e., the correlation gap is bounded by this constant. Combining these observations shows that the core of the argument from the additive case continues to apply. The only loss is a multiplicative factor of $\left(1 - \frac{1}{e}\right)$ in the constant $c$; consequently, the asymptotic guarantees of our result remain unchanged.

*Constant MMS for Binary Additive Valuations.* We next ask whether it is possible to guarantee a constant fraction of MMS ex post for every agent, at the expense of a slightly larger probability of failure. This requires moving away from independent good allocation, while still ensuring that every good is equally likely to remain unallocated.

Henceforth we will focus on binary additive valuations. In particular, we assume that every agent $i$ has a desired set $S_i \subseteq [m]$ of goods; The agent assigns equal value to every good in $S_i$ and zero to goods not in $S_i$. In order to achieve an $\alpha$-MMS for some $\alpha \in (0, 1)$, it suffices to allocate any subset of size $\lceil \frac{\alpha}{n} |S_i| \rceil$ to the agent. Our algorithm picks a uniformly random subset of this size to allocate to the agent.

Our goal is once again to prove that when an agent arrives, enough elements in their desired set $S_i$ remain unallocated. The challenge is that the allocation events across different goods are no longer independent, so standard concentration inequalities no longer apply.

To resolve this, we carry out a two-step argument. We first consider the setting of nested agent valuations, that is, if agent $i$ arrives after agent $i'$, then $S_i \subseteq S_{i'}$. In this case, we show that the goods in each agent's desired set at the time of her arrival are exchangeable and therefore negatively associated. This allows us to apply appropriate concentration inequalities.

Second, we provide a novel and clever charging argument relating the sizes of allocations in the general binary case to a suitably defined "partially nested" setting where the last agent's desired set is contained in all other agents' desired sets. Therefore, while the original algorithm's goods satisfy neither independence nor exchangeability we can couple them with a set of exchangeable goods. By carefully interleaving the concentration and charging arguments round by round, we bound the overall probability of over-allocation in the algorithm.

Our final result, Theorem 4.5 achieves a 256 factor approximation of MMS with a constant probability.

## 1.2 Other Related Work

In this section, we highlight previous work that is most closely related to our setting. We refer the reader to excellent surveys [Aleksandrov and Walsh, 2020, Amanatidis et al., 2023] for further references.

*Online fair division with agent arrivals.* To the best of our knowledge, [Kulkarni et al., 2025] is the only paper that considers the agent arrival setting with indivisible goods prior to our work. As discussed in Section 1, the paper considers the stochastic case where the valuation function of the arriving agent is drawn from a finite distribution over apriori known additive $k$ agent types. No results are known when the allocator has no knowledge of future agents' valuation functions that are chosen by an (oblivious) adversary.

Other works on online fair division with agent arrivals focus on the setting where goods are divisible. Walsh [2011] consider the setting of dividing a single divisible good to $n$ agents arriving online. Kash et al. [2014] studied a setting called dynamic fair division to fairly allocate a set of divisible goods to agents who arrive but never depart from the system. Works by Sinclair et al. [2022] and Banerjee et al. [2023] studied the setting of allocating a set of goods to agents which are classified by their valuation functions. Friedman et al. [2015, 2017] consider algorithms simultaneously achieve fairness notions and minimize disruptions where disruption is defined when a reallocation of a previously allocated resource happens. In a set of related works, Bogomolnaia et al. [2019] consider the setting where agent's valuation functions are drawn from a distribution, while Donahue and Kleinberg [2020] study the setting where uncertain number of agents from different "groups" arrive. They both define the notion of fairness as agents from different groups receive resource with equal probability.

*Online fair division with indivisible items.* While most of the current literature in online fair division focuses on divisible items, there are a few papers dealing with indivisible items. These mostly focus on the item-arrival setting. Benade et al. [2018] study the fairness notion of envy, minimizing the total envy over a certain period of time. He et al. [2019] minimize the total number of re-allocations needed to achieve EF1. Zhou et al. [2023] study MMS in the item-arrival setting for indivisible goods and chores. Zeng and Psomas [2020] focus on the setting where agents' values are drawn from a random distribution, and study the fairness and efficiency trade-offs over time. Procaccia et al. [2024] and Yamada et al. [2024] study how to allocate indivisible online items through the lens of bandit learning. A recent work Halpern et al. [2025] connects envy minimization in online indivisible item arrival setting to discrepancy theory.

## 2 Notation and Preliminaries

*Notations.* For any positive integer $n$, let $[n] = \{1, 2, \dots, n\}$, and for two positive integers $i, j$ where $i < j$, let $[i, j] = \{i, i + 1, \cdots, j\}$. For a set or list $N$, let $|N|$ denote the number of elements in $N$.

### 2.1 Problem Setting

**Instance.** We consider the problem of fairly allocating $m$ indivisible goods/goods among $n$ heterogeneous agents. Throughout, we use index $i$ to denote an agent and index $j$ to denote a good. Without loss of generality, we assume agent $i$ arrives at time $i$, and her preferences over the subsets of goods are represented by a non-negative non-decreasing valuation function $v_i : 2^{[m]} \to \mathbb{R}_+$. We denote an instance of the fair division problem as $\mathcal{I} = \{[n], [m], (v_i)_{i\in[n]}\}$.

**Valuation functions.** In this paper we study three well-known valuation functions, namely additive, binary additive, and submodular.

Valuation function $v_i$ is said to be additive if for any subset $S \subseteq [m]$ of goods, $v_i(S) = \sum_{j\in S} v_i(j)$ where $v_i(j)$ is the value that agent $i$ derives from good $j$. Additive is well known to strike a good balance between the expressivity of preferences and complexity of reporting/eliciting and cognition, and hence has been extensively studied not only within fair division (see survey [Amanatidis et al., 2023], but also in optimization (e.g., [Bansal and Sviridenko, 2006]).

An additive function $v_i$ is called *binary additive*, if $v_i(j) \in \{0, 1\}$ for all good $j \in [m]$. That is, the agent needs to only report which goods she likes without any comparison between them. Such valuations are clearly the easiest to report/elicit and therefore are preferred in practice, attracting extensive work within social choice theory, e.g., see [Aleksandrov et al., 2015, Barman et al., 2018, Brams and Fishburn, 1978, Halpern et al., 2020].[3]

*Submodular* functions are much more general, and manages to capture the *diminishing marginal utility* of agents' preferences [Kauder, 2015]. Formally, A valuation function $v_i$ is said to be

[3]Approval voting is essentially based on binary preferences.

submodular iff it satisfies the following:

$$\forall S \subset T \subseteq [m] \text{ and } \forall j \in [m] \setminus T, \quad v_i(T \cup \{j\}) - v_i(T) \geq v_i(S \cup \{j\}) - v_i(S)$$

Note that, reporting such a valuation function will require the agent to report $2^m$ values! To keep things *polynomial-time*, it is typical to assume that the agent will report on a need-to-know basis through oracle access.

**Fairness Notion: Maximin Share (MMS).** For any set $S$ of goods and a positive integer $d$, let $\Pi_d(S)$ denote the collection of all partitions of $S$ into $d$ bundles.

**Definition 2.1** (Maximin Share (MMS)). Given an instance $\mathcal{I} = ([n], [m], \{v_i\}_{i\in[k]})$ of the fairdivision problem, the *maximin share* (MMS) value of agent $i$ is defined as

$$\mathsf{MMS}_i = \max_{P\in\Pi_n([m])} \min_{j\in[n]} v_i(P_j).$$

An allocation $(A_1, \ldots, A_n) \in \Pi_n([m])$ is said to be an MMS allocation if $v_i(A_i) \geq \mathsf{MMS}_i$ for all agents $i$. It is said to be $\alpha$-approximate MMS or just $\alpha$-MMS, for an $\alpha \in [0, 1]$ if,

$$v_i(A_i) \geq \alpha * \mathsf{MMS}_i, \quad \forall i \in [n].$$

**OnlineMMS problem with adversarial agent arrival.** We study the problem of computing an (approximate) MMS allocation in the online setting where agents arrive online over time. When agent $i$ arrives at time $i$, her valuation function is revealed to the algorithm, and it must irrevocably allocate a subset of goods to the agent from the available set of goods.

We consider an adversarial arrival model where the adversary has access to the algorithm we apply to allocate the goods for choosing the agents' valuations. However, the adversary does not have access to the random coin tosses of the algorithm. In fact, it is easy to see that against an adaptive adversary, who has access to the latter as well, no non-trivial guarantees are possible [Kulkarni et al., 2025]. Therefore, our adversary model is the strongest possible for which one can hope to get non-trivial guarantees, as far as we understand.

**Definition 2.2** ($(\alpha, \beta)$-competitive algorithm). For $\alpha \in [0, 1]$ and $\beta \in [0, 1]$, we say that an online algorithm is $(\alpha, \beta)$-MMS competitive if it ensures $\alpha$-approximate MMS allocation to *all* agents ex-post, with probability at leat $\beta$.

**Definition 2.3** (High-valued goods). In an OnlineMMS problem where the goal is to obtain an $(\alpha, \beta)$-competitive guarantee given a normalized input instance, an good $j$ is considered *high-valued* by agent $i$ if $v_i(j) \geq \alpha$.

**Normalizations.** To aide our analysis we will assume that the given instance is normalized, without loss of generality (wlog), as defined below (see Appendix A for why it is wlog).

**Definition 2.4** (Normalized Instance). An input instance $\mathcal{I} = ([n], [m], \{v_i\}_{i\in[n]})$ of an OnlineMMS problem with (non-binary) additive valuations is *normalized* if for every agent $i \in [n]$, $\mathsf{MMS}_i = 1$ and the total value $v_i([m]) = n$. This in turn implies $v_i(j) \in [0, 1]$, for all agent $i$ and goods $j$.

We call instance $\mathcal{I}$ with submodular valuations normalizes if for every agent $i \in [n]$, $\mathsf{MMS}_i \leq 1$.

## 2.2 Probabilistic/Fractional Allocations: Function Extensions

Our work considers randomized allocation algorithms in which goods are assigned to agents with certain probabilities. To analyze the expected value received by an agent under such randomized outcomes, we must extend valuation functions to probabilistic or fractional allocations. For additive valuations, this extension is straightforward: if an agent $i$ receives good $j \in [m]$ with probability $p_j$, then their expected value is $\sum_{j \in [m]} v_i(j) p_j$.

For non-additive valuations, defining the value of a fractional allocation vector $\vec{p} \in [0,1]^m$ is more subtle. In particular, we need to understand the expected value an agent receives when each good $j \in [m]$ is allocated with marginal probability $p_j$. There are multiple natural ways to interpret such fractional allocations. Two important notions that have been extensively studied in the literature are the multilinear extension and the concave extension of submodular valuation functions. We give the details of these extensions in Appendix B.

## 2.3 Probability Preliminaries

In this section, we introduce two notions of correlations that a collection of random variables may satisfy, along with the corresponding concentration guarantees. These notions arise naturally from different rounding procedures used in our allocation algorithms and are crucial for analyzing the value obtained by each agent.

*2.3.1 Independent Random Variables.* We first consider the case where goods are rounded independently for each agent. In this setting, to show that the resulting allocation provides sufficient value to each agent with high probability, we rely on a dimension-free concentration bound for submodular functions given by [Vondrak, 2010].

**Lemma 2.5** (Corollary 3.4 of [Vondrak, 2010]). *Let $f : 2^N \to \mathbb{R}_+$ be monotone submodular function with marginals in $[0, \ell]$. For $\mathbf{x} \in [0,1]^N$, let $R$ be a random set where each element $i$ is drawn independently using the marginals induced by $\mathbf{x}$. Then for $\delta \geq 0$,*

$$Pr[f(R) \leq (1-\delta)F(\mathbf{x})] \leq e^{-\frac{\delta^2 F(\mathbf{x})}{2\ell}}.$$

*2.3.2 Exchangeable Random Variables.* A common and analytically tractable form of dependence is *exchangeability*, which captures symmetric correlations among random variables. Formally, a collection of random variables $(X_1, \ldots, X_m)$ is said to be *exchangeable* if, for every permutation $\sigma$ of $\{1, \ldots, m\}$, $(X_1, \ldots, X_m) \overset{d}{=} (X_{\sigma(1)}, \ldots, X_{\sigma(m)})$, that is, the joint distribution of $(X_1, \ldots, X_m)$ is invariant under any permutation of the indices. We use the following concentration inequality for exchangeable random variables, which provides Chernoff-type tail bounds for weighted sums. Let $\epsilon_N = O\left(\frac{\log N}{N}\right)$.

Theorem 2.6 ([Barber, 2024]). *Let $w \in \mathbb{R}^n$ be a fixed weight vector and let $X_1, \ldots, X_N \in [-1,1]$ be exchangeable random variables, where $N \geq n \geq 2$. Then, for any $\delta \in (0,1)$,*

$$\mathbb{P}\left\{\sum_{i=1}^{n} w_i (X_i - \bar{X}) \geq \|w\|_2 \sqrt{2(1+\epsilon_N)\,\log(1/\delta)}\right\} \leq \delta,$$

*where $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$.*

## 3 Independent Rounding: A Logarithmic Approximation to OnlineMMS

In this section, we will focus on designing an $(\alpha, \beta)$-competitive algorithm for any $\alpha = O(\frac{1}{\log n})$ and $\beta = 1 - O(\alpha^2 \log \frac{1}{\alpha})$. Our algorithm relies on item-independent rounding with carefully designed probabilities of allocation that depend on the stage of the algorithm.

We begin the section by establishing a lower bound on the tradeoff between the MMS approximation achieved and the probability with which it is guaranteed by the algorithm. We then introduce our algorithm and analysis for additive valuations in Section 3.2. Finally, in Section 3.3 we prove that the same algorithm will work even when agents have submodular valuation functions.

### 3.1 A Lower Bound

We first show that for any $c > 0$, no online allocation algorithm can guarantee $c$-MMS to all agents simultaneously with probability at least $1 - \frac{c^2}{2}$.

Theorem 3.1. *For any $c \in (0, 1]$, there exists an* OnlineMMS *instance $\mathcal{I}$ with binary additive valuations, where no randomized algorithm is $(c, \beta)$-competitive for any $\beta \geq 1 - \frac{c^2}{2}$.*

Proof. We'll construct our instance $\mathcal{I}$ as below. Let $n$ be an sufficiently large integer. The instance contains $n$ agents and $\lceil \frac{1}{c} \rceil n$ goods. The first agent values every good at 1, while the remaining $n-1$ agents value each good in a uniformly random set of $n$ out of $\lceil \frac{1}{c} \rceil n$ goods at 1 while all other goods at 0.

For a randomized algorithm to be $(c, \beta)$-competitive, it must allocate at least $c \cdot \lceil \frac{1}{c} \rceil > 1$ goods to the first agent since her MMS value is $\lceil \frac{1}{c} \rceil$. This means we must allocate at least 2 goods to the first agent. Denote the set of $n$ random goods which are valued 1 for the remaining $n-1$ agents as $S$. To ensure the remaining agents each gets a $c$-MMS bundle, we can at most allocate 1 good from $S$ to the first agent, otherwise at least one of the remaining agents will get no good in $S$, which yields a value 0. Thus $\beta < 1 - \mathbb{P}\left[2 \text{ goods in } S \text{ are allocated to the first agent}\right] = 1 - \binom{n}{2} / \binom{\lceil \frac{1}{c} \rceil n}{2} = 1 - \frac{n(n-1)}{\lceil \frac{1}{c} \rceil n (\lceil \frac{1}{c} \rceil n - 1)} < 1 - \frac{c^2}{2}$. The last inequality holds true if we pick $n$ large enough. □

### 3.2 Additive Valuations

Let us consider the instances in which all agents have additive valuation functions. Fix an instance $\mathcal{I}$, we can assume without loss of generality that it is already normalized. Note that our goal is to guarantee each agent an allocation with value at least $\alpha$ for $\alpha \in O(\frac{1}{\log n})$. Our formal algorithm appears in Algorithm 1.

**Description of Algorithm 1.** When agent $i$ arrives, if there's a good $j$ considered high-valued by agent $i$ ($v_i(j) \geq \alpha$), we directly allocate $j$ to agent $i$. This will guarantee agent $i$ receives value at least $\alpha$. We call this round of allocation as a special round. In a non-special round, no leftover goods are considered high-valued by agent $i$. In this case, if we have had at most $n - \frac{4}{\alpha}$ special rounds previously, we allocate every good with probability $\frac{1}{2(n-t)}$ where $t$ is the number of previous special rounds On the other hand, if the number of special rounds before the current one is more than $n - \frac{4}{\alpha}$, we increase the probability of allocation of each good to $\frac{64}{n-t} \cdot \alpha \log \frac{1}{\alpha}$. The intuition here is that since many special rounds have happened, the agent could have lost a lot of value in these rounds. Therefore, to ensure she gets enough value in this round from the leftover goods, we suitably increase the allocation probability.

We then argue using appropriate concentration bound that no matter when agent $i$ arrives, she receives an $\alpha$-fraction of the MMS value. The threshold on when to increase the probability of allocation balances the trade-off between the present agent receiving large enough value while keeping enough value available to be allocated to future agents.

In the algorithm, $A_i$ is the bundle we allocate to agent $i$. We state the following theorem.

Theorem 3.2. *Given an instance $\mathcal{I} = ([n], [m], (v_i)_{i \in [n]})$ and parameter $\alpha \in O(1/\log n)$, Algorithm 1, is a $(\alpha, 1 - O(\alpha^2 (\log \frac{1}{\alpha})^3))$-competitive algorithm. This means that for* all *agents $i$ we have $v_i(A_i) \geq \alpha$ with probability at least* $1 - O(\alpha^2 (\log \frac{1}{\alpha})^3)$.

Our proof utilizes the following lemmas, which compute the probability that $v_i(A_i) \geq \alpha$ for different possible values of $i$, then by applying union bound we obtain the final proof. Recall that we use special rounds to refer rounds where we are allocating a high-valued good to an agent, and

**ALGORITHM 1:** Algorithm for Additive Valuations with parameter $\alpha \in (0,1)$

```
$t \leftarrow 0$ ;                                   /* counter for the number of special rounds */
$R_0 = [m]$ ;                                                 /* Remaining set of goods */
for each arriving agent $i \in [n]$ do
    $A_i \leftarrow \emptyset$;
    if $\exists j \in R_{i-1}$ such that $v_i(j) \geq \alpha$ then
        $A_i = \{j\}$;
        $t \leftarrow t+1$
    end
    else
        for each good $j \in R_{i-1}$ do
            if $t \leq n - \frac{4}{\alpha}$ then
                $A_i \leftarrow A_i \cup \{j\}$ with probability $\frac{1}{2(n-t)}$
            end
            else
                $A_i \leftarrow A_i \cup \{j\}$ with probability $\frac{64}{n-t} \cdot \alpha \log \frac{1}{\alpha}$
            end
        end
    end
    $R_i = R_{i-1} \setminus A_i$;
end
```

non-special rounds to refer rounds that we are independently allocating all remaining goods with a certain probability.

**Lemma 3.3.** *For agent i where the number of special rounds before agent i is at most* $n - \frac{4}{\alpha}$, $v_i(A_i) \geq \alpha$ *with probability* $p_i \geq 1 - \frac{1}{n^2}$

Proof Sketch. Since the number of special rounds for these agents is bounded by $n - \frac{4}{\alpha}$, the allocation probability of the goods is $\frac{1}{2(n-t)}$. For these agents, we bound the allocation probability in two cases – either the agent has many ($> \frac{2}{\alpha}$) high valued goods or few high valued goods. If the number of high valued goods is large, we argue that at least one of these must be available with a high probability otherwise we argue that there is sufficient value left in the low valued goods of the agent when she arrives so she can receive value from these goods. Full proof is in Appendix C.1.1. □

**Lemma 3.4.** *For agent i where the number of special rounds before agent i is at least* $n - \frac{4}{\alpha}$ *and* $i \neq n$, $v_i(A_i) \geq \alpha$ *with probability* $p_i \geq 1 - O((\alpha \log \frac{1}{\alpha})^3)$

Proof. We'll use $t_i$ to denote the number of special rounds that happens before agent $i$, $S_i$ to denote the set of goods that is not allocated in those $t_i$ special rounds (note this is not $R_{i-1}$). We'll use $H_i$ to denote the set of goods that is considered high-valued by agent $i$ in $S_i$ and $h_i = |H_i|$. We denote $\mathcal{I}_1$ as the indicator of that there is no good in $R_{i-1}$ which is considered high-valued by agent $i$ and $\mathcal{I}_2$ as the indicator of that if we allocate with appropriate probability all the low-valued goods in $R_{i-1}$, the value of this random set is below $\alpha$. If $\mathcal{I}_1 \neq 1$, then we have at least one high-valued good left, thus we'll allocate a high-valued good to $i$. If $\mathcal{I}_2 \neq 1$, then if we are allocating low-valued goods to agent $i$, the random set will have value more than $\alpha$. Thus $p_i \geq 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1])$

Note that, under the case where $t_i > n - \frac{4}{\alpha}$, we observe the transition of potential allocation probability from $\frac{1}{2(n-t)}$ to $\frac{1}{n-t} \cdot \alpha \log \frac{1}{\alpha}$. Again we need to argue about the probability of any good

$j \in S_i$ that is remained in $R_{i-1}$. Denote the number of non-special rounds before the transition as $r$. Then the number of non-special rounds after the transition can be computed as $i-1-t_i-r$. Before the transition, we allocate every good with probability $\frac{1}{2(n-t)} \le \frac{\alpha}{8}$ in each non-special round (this is because $n-t \ge \frac{4}{\alpha}$ in this phase). After the transition, we allocate every good with probability $\frac{1}{n-t} \cdot \alpha \log \frac{1}{\alpha} \le \frac{1}{n-t_i} \cdot \alpha \log \frac{1}{\alpha}$ in each non-special round. Using union bound we have that for $j \in S_i$, $\mathbb{P}[j \in R_{i-1}] \ge 1 - (\frac{\alpha r}{8} + \frac{i-1-t_i-r}{n-t_i} \cdot \alpha \log \frac{1}{\alpha})$.

We'll need to discuss different cases of $r$ and $h_i$. Note that $r \le \frac{4}{\alpha}$ since $r + t_i < i-1$ and $t_i > n - \frac{4}{\alpha}$.
**Case 1:** $r \ge 4 \log \frac{1}{\alpha}$**.** Here, $v_i(S_i) \ge r \ge 4 \log \frac{1}{\alpha}$. Note that in this case for $j \in S_i$, $\mathbb{P}[j \in R_{i-1}] \ge 1 - (\frac{\alpha r}{8} + \frac{(i-1-t_i-r)64}{n-t_i} \cdot \alpha \log \frac{1}{\alpha}) > \frac{1}{2}$. The last inequality follows from the fact that $t_i > n - \frac{4}{\alpha}$.

**Subcase 1:** $|H_i| = h_i \ge 3 \log \frac{1}{\alpha}$**.** Since each good in $H_i$ will remain in $R_{i-1}$ with probability at least $\frac{1}{2}$, $\mathbb{P}[I_1 = 1] < (\frac{1}{2})^{h_i} \le O(\alpha^3)$. Thus $p_i \ge 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \ge 1 - \mathbb{P}[\mathcal{I}_1 = 1] > 1 - O(\alpha^3)$.

**Subcase 2:** $|H_i| = h_i < 3 \log \frac{1}{\alpha}$**.** Then the total value of small-valued goods in $S_i$ is at least $v_i(S_i) - h_i > n - t_i - h_i$. Using the fact that each good from $S_i$ remains with probability at least $\frac{1}{2}$ and since we allocate every good with probability $\frac{1}{n-t_i} \cdot \alpha \log \frac{1}{\alpha}$, we can compute $\mathbb{E}[v_i(A_i)] = \frac{1}{2}(n-t_i-h_i)\frac{64}{n-t_i} \cdot \alpha \log \frac{1}{\alpha} \ge 8\alpha \log \frac{1}{\alpha}$. The last inequality is because $n - t_i \ge 4 \log \frac{1}{\alpha}$ while $h_i < 3 \log \frac{1}{\alpha}$. Applying Lemma 2.5 we have

$$\begin{aligned}
\mathbb{P}[\mathcal{I}_2 = 1] &= \mathbb{P}[v_i(A_i) < \alpha] \\
&= \mathbb{P}\left[v_i(A_i) < 8\frac{1}{\log \frac{1}{\alpha}} \cdot \mathbb{E}[v_i(A_i)]\right] \\
&\le \exp\left(-\frac{(1 - 8 \log {}^1\!/\!_\alpha)^2 \mathbb{E}[v_i(A_i)]}{2\alpha}\right) \\
&= \exp\left(-((1 - 8 \log {}^1\!/\!_\alpha)^2) 4 \log {}^1\!/\!_\alpha\right) \le O(\alpha^4).
\end{aligned}$$

Thus $p_i \ge 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \ge 1 - \mathbb{P}[\mathcal{I}_2 = 1] > 1 - O(\alpha^4)$.
**Case 2:** $r < 4 \log \frac{1}{\alpha}$**.** Note that in this case for $j \in S_i$, $\mathbb{P}[j \in R_{i-1}] \ge 1 - (\frac{\alpha r}{8} + \frac{64(i-1-t_i-r)}{n-t_i} \cdot \alpha \log \frac{1}{\alpha}) > 1 - O(\alpha \log \frac{1}{\alpha})$.

**Subcase 1:** $|H_i| = h_i \ge 3$**.** Since each good in $H_i$ will be not in $R_{i-1}$ with probability at most $O(\alpha \log \frac{1}{\alpha})$, $\mathbb{P}[I_1 = 1] < O(\alpha \log \frac{1}{\alpha})^{h_i} \le O(\alpha \log \frac{1}{\alpha})^3$. Thus $p_i \ge 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \ge 1 - \mathbb{P}[\mathcal{I}_1 = 1] > 1 - O(\alpha \log \frac{1}{\alpha})^3$.

**Subcase 2:** $|H_i| = h_i \le 2$**.** Then the total value of small-valued goods in $S_i$ is at least $v_i(S_i) - h_i > n - t_i - 2$. Since we allocate every good with probability $\frac{c}{n-t_i} \cdot \alpha \log \frac{1}{\alpha}$, we can compute $\mathbb{E}[v_i(A_i)] = (1 - O(\alpha \log \frac{1}{\alpha}))(n - t_i - 2)\frac{64}{n-t_i} \cdot \alpha \log \frac{1}{\alpha} \ge 8\alpha \log \frac{1}{\alpha}$. The last inequality is because $n - t_i \ge n - (i-1) \ge 3$ ($i$ is at most $n-1$) while $h_i \le 2$. Applying Lemma 2.5 we have $\mathbb{P}[\mathcal{I}_2 = 1] = \mathbb{P}[v_i(A_i) < \alpha] = \mathbb{P}\left[v_i(A_i) < 8\frac{1}{\log \frac{1}{\alpha}} \cdot \mathbb{E}[v_i(A_i)]\right] \le O(\alpha^4)$ following the same calculations as those in Case 1, Subcase 2. Thus $p_i \ge 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \ge 1 - \mathbb{P}[\mathcal{I}_2 = 1] > 1 - O(\alpha^4)$.

In all of the cases we have $p_i \ge 1 - O((\alpha \log \frac{1}{\alpha})^3)$. □

**Lemma 3.5.** *For agent $n$, $v_n(A_n) \ge \alpha$ with probability $p_n \ge 1 - O((\alpha \log \frac{1}{\alpha})^2)$.*

Proof. When $t_n \le n - \frac{4}{\alpha}$, all proofs from Lemma 3.3 work and we can obtain $p_n \ge 1 - \frac{1}{n^2}$.

When $t_n > n - \frac{4}{\alpha}$, almost all proofs from Lemma 3.4 work except for the last part when $r < 4 \log \frac{1}{\alpha}$, we need to divide the subcases into $h_i \ge 2$ and $h_i \le 1$. Then applying almost the same argument we can obtain that $p_n \ge 1 - O((\alpha \log \frac{1}{\alpha})^2)$. □

Proof of Theorem 3.2. Denote $p$ as the probability that in Algorithm 1 all agents $i$ will have $v_i(A_i) \geq \alpha$. Combining Lemma 3.3, Lemma 3.4, Lemma 3.5 and union bound, we can compute that $p \geq 1 - (\sum_{i=1}^{n}(1-p_i)) = 1 - (\sum_{t_i \leq n-\frac{4}{\alpha}} \frac{1}{n^2} + (\sum_{t_i > n-\frac{4}{\alpha}} O((\alpha \log \frac{1}{\alpha})^3)) + O((\alpha \log \frac{1}{\alpha})^2)) \geq 1 - O(\alpha^2(\log \frac{1}{\alpha})^3)$. The last inequality is because there are at most $\frac{4}{\alpha}$ agents $i$ which can satisfy $t_i > n - \frac{4}{\alpha}$. □

## 3.3 Submodular Valuations

In this section, we show how to extend the results in the previous section to submodular valuations. The following is the theorem we prove here.

Theorem 3.6. *Given an instance $\mathcal{I} = ([n], [m], (v_i)_{i\in[n]})$ of the OnlineMMS problem with submodular valuations $v_i$ and parameter $\alpha \in O(\frac{1}{\log n})$, Algorithm 1 is $((1-\frac{1}{e})\,\alpha, 1 - O(\alpha^2(\log \frac{1}{\alpha})^3))$-competitive.*

Proof Sketch. First note that we are keeping the algorithm same. Therefore, the probability with which a good is allocated to a particular agent $i \in [n]$ and the probability with which it is remaining for the agent is unchanged since it is independent of the valuation functions. Consequently, for agents who get value with high probability by being allocated a high-valued good, the analysis remains unchanged from the additive case. In particular, Case 1 of Lemma 3.3 and Subcases 1 of both cases in Lemma 3.4 go through as they are. We only need to argue the value obtained by the agents who receive their allocation via independent rounding. For such agents, note that the analysis of the additive case proceeded in the following steps: (1) Compute the expected value the agent receives, and (2) Show that the actual value that the agent receives concentrates around the expected value with a sufficiently high probability that was dependent on the particular case.

The second part of this, i.e., showing an appropriately high concentration follows in each subcase based on having an appropriate expected value. In particular, note that in the analysis, if the expected value of the agent changes by some constant factor, then the analysis of each of the remaining cases goes through unchanged except for a constant factor change in the $\alpha$ we can achieve. Therefore, asymptotically our result will remain unchanged as long as the expected value is only a constant factor away. We therefore, show here that for all agents who receive a bundle from independent rounding of the low-valued goods that are remaining, we get in expectation a value only $(1-\frac{1}{e})$ away from the expected value of the additive case.

Fix an agent $i \in [n]$. As in the previous section, we define $t_i$ to be the number of special rounds that have occurred before agent $i$ came in; $S_i$ to be the set of goods *not* allocated in these special rounds; $H_i$ to be the high valued goods of agent $i$ in $S_i$; and $R_{i-1}$ the random set of goods that are left when agent $i$ arrives. Let $p_1$ be the probability that a good $j$ remains in $R_{i-1}$ and $p_2$ be the probability that a good $j$ from $R_{i-1}$ is allocated to agent $i$. Let $h_i = |H_i|$ be the number of high valued goods of agent $i$. Note that the three cases differ according to the number $h_i$, $t_i$, $p_1$, and $p_2$. The expected value that such agents receive is the expected value they receive when a good from $S_i \setminus H_i$ is allocated to them independently with probability $p_1p_2$ – i.e., the probability that the good is remained multiplied by the probability that the agent obtains the good. This is precisely the value of multilinear extension of the valuation function $v_i$ when we restrict it to the set $S_i \setminus H_i$[4]. We denote the corresponding multilinear extension and concave extension of these restrictions as $F_{\downarrow S_i \setminus H_i}$ and $f^+_{\downarrow S_i \setminus H_i}$. Finally, let us denote the MMS partition of agent $i \in [n]$ as $(A^*_k)_{k\in[n]}$. Note that in these last notations, we have dropped the subscript $i$ corresponding to the agent for clarity and because we have fixed a particular agent. Therefore, recalling the definitions of multilinear extension and

[4]Note that the restriction of a submodular function to a subset is still submodular.

concave extension from Section 2, we can calculate the expected value of the allocated set as

$$\begin{aligned}
\mathbb{E}[v_i(A_i)] &= F_{\downarrow S_i \setminus H_i}(\mathbf{p_1 p_2}) && \text{(By definition of multilinear extension)}\\
&\geq \left(1-\frac{1}{e}\right) f^{+}_{\downarrow S_i \setminus H_i}(\mathbf{p_1 p_2}) && \text{(By correlation gap)}\\
&\geq \left(1-\frac{1}{e}\right) \cdot \sum_{k\in[n]} p_1 p_2 \cdot v_i((A^*_k \cap S_i) \setminus H_i) && \text{(Using definition of concave extension)}\\
&\geq \left(1-\frac{1}{e}\right) \cdot p_1 p_2 \cdot (n - t_i - h_i) MMS_i\\
&= \left(1-\frac{1}{e}\right) \cdot p_1 p_2 \cdot (n - t_i - h_i)
\end{aligned}$$

where the last inequality follows because we assume scaling of MMS to 1. Consequently, it is easy to see that using appropriate values of $p_1$, $p_2$ and $h_i$ for all the cases of the Lemma 3.3, Lemma 3.4, and Lemma 3.5, we can recover all the proofs for an $\alpha$ that scales down by $\left(1-\frac{1}{e}\right)$ to ensure the correct exponent in the probability analysis.

$\square$

## 4 A Constant-Competitive Algorithm for Binary Additive

In this section, we present a correlated rounding algorithm for agents with binary additive valuations. Throughout this section, we assume that agents have additive valuations and that, for all $i \in [n]$ and $j \in [m]$, the value $v_i(j) \in \{0, 1\}$. Consequently, we do not assume that valuation instances are normalized.

Since valuations are binary, for each agent $i \in [n]$ we may equivalently specify a set $S_i \subseteq [m]$ of goods that the agent values, where $v_i(j) = 1$ if and only if $j \in S_i$.

We proceed in two stages. We first analyze an algorithmically simple special case in which agents' valuation sets are *nested*, that is, $S_1 \supseteq S_2 \supseteq \cdots \supseteq S_n$. For this case, we show that concentration bounds for exchangeable random variables allow us to establish strong high-probability guarantees. In particular we obtain a simple $\left(\frac{1}{16}, 1-\frac{1}{n}\right)$ -competitive algorithm for this setting.

We then turn to the general case, where valuation sets need not be nested. Our analysis builds on the intuition that the nested case, while analytically simpler, represents the most challenging instance under binary valuations: for each successive agent, all goods they value are also valued by every earlier agent, leading to maximal competition. Instances in which valuation sets are not nested should therefore be better from the perspective of later agents.

We formalize this intuition by introducing, for each agent $i \in [n]$, an auxiliary algorithm that is used only for the purposes of analysis. We compare the primary algorithm with this auxiliary algorithm by coupling the random variables that describe the number of goods lost by agent $i$ in each process. Establishing such a coupling directly is nontrivial; instead, we construct an *approximate coupling*. This approximate coupling allows us to prove approximate first-order stochastic dominance between the random variables describing the number of goods remaining for agent $i$ under the two algorithms.

Crucially, the analysis algorithm preserves exchangeability for the agent under consideration. As a result, we can invoke concentration bounds for exchangeable random variables, together with approximate first-order stochastic dominance, to derive $(\frac{1}{256}, \Theta(1))$-competitive algorithm for the general case.

### 4.1 Warm Up: Binary *Nested* Valuations

Let us consider the instances in which all agents have 0/1 valuations, agent $i$ has an MMS value of $\mathsf{MMS}_i = k_i$. We can assume without loss of generality that the agent likes exactly $nk_i$ goods. Denote by $S_i$ be the subset of goods that she likes. We assume that the valuations are chained so $S_1 \supseteq S_2 \supseteq \cdots \supseteq S_n$, implying $k_1 \geq k_2 \geq \cdots \geq k_n$. We use $L_i$ to be the random variable denoting the set of goods leftover when agent $i$ arrives. Consider Algorithm 2.

**ALGORITHM 2:** Algorithm for Nested Binary Valuations

```
Draw a random permutation $\sigma$ of the goods $[m]$ uniformly from all permutations.
for each arriving agent $i \in [n]$ do
    Allocate the smallest-indexed $\max\{\lfloor \frac{1}{4n}|L_i| \rfloor, 1\}$ goods from $L_i$.
end
```

We first observe the following claim.

**Claim 4.1.** *Fix any agent $i \in [n]$. Let $L_{ij}$ be the indicator random variable for the event that good $j$ is available when agent $i$ arrives. Then $(L_{ij})_{j \in S_i}$ are exchangeable.*

Proof. The algorithm begins by sampling a permutation $\sigma$ of the goods $[m]$ uniformly at random. Restricting $\sigma$ to the subset $S_i$, the induced permutation on $S_i$ is therefore uniform over all permutations of $S_i$. By the nested valuation assumption, all agents arriving before agent $i$ value every good in $S_i$. Moreover, the allocation rule for each agent depends only on the relative order of the remaining goods under $\sigma$, and not on their identities. Consequently, the allocation decisions of agents $1, \ldots, i-1$ treat all goods in $S_i$ symmetrically. Formally, for any permutation $\pi$ of $S_i$, the joint distribution of the availability indicators $(L_{ij})_{j \in S_i}$ is identical to that of $(L_{i,\pi(j)})_{j \in S_i}$. This invariance under permutations of $S_i$ implies that the random variables $(L_{ij})_{j \in S_i}$ are exchangeable. □

Now, let $X_{ij}$ be the indicator random variable denoting that good $j$ is selected in round $i$.

**Claim 4.2.** *In any iteration where $\lfloor \frac{|L_i|}{4n} \rfloor > 1$, we have $\mathbb{P}\left[X_{ij} = 1\right] \leq \frac{1}{4n}$ for any good $j \in [m]$.*

Proof. In any round $i \in [n]$, if the good $j$ has been allocated previously, or if $j \notin S_i$, it is not allocated and $X_{ij} = 0$ deterministically. Therefore, for these rounds $\mathbb{P}\left[X_{ij} = 1\right] = 0$. On the other hand, if $j$ is available in round $i$ and $j \in S_i$ then, given that $\lfloor \frac{|L_i|}{4n} \rfloor > 1$; we have that we allocate at most the top $\frac{1}{4n}$ fraction of the leftover indices to this agent. Since the initial permutation is drawn uniformly at random, the probability that the good is in top $\frac{1}{4n}$ indices is $\frac{1}{4n}$. Therefore, $\mathbb{P}\left[X_{ij} = 1\right] = \frac{1}{4n}$. Together, the claim is proved. □

Finally, we note the following observation which follows from the fact that the valuations are nested.

**Observation 4.3.** *If at any round $i$ in the algorithm $\max\left\{\lfloor \frac{|L_i|}{4n} \rfloor, 1\right\} = 1$ then for all future rounds $i' > i$, $\max\left\{\lfloor \frac{|L_{i'}|}{4n} \rfloor, 1\right\} = 1$.*

Using these claims and observations, we can prove our main theorem.

Theorem 4.4. *Algorithm 2 is a $\left(\frac{1}{16}, 1 - \frac{1}{n}\right)$- competitive algorithm for binary instances with nested valuations for $n \geq 68$. In particular, with high probability all agents receive $\frac{1}{16}$ of their MMS value.*

Proof. If there is no round in which some agent $i$ is allocated more than one good, then this already holds in the first round, when all goods remain. In that round we have $|L_1| = nk_1 < 8n$, and hence $k_1 < 8$. Since MMS values are non-increasing across agents, for every agent $i$, $\mathsf{MMS}_i \leq 7$, and by the algorithm, it follows that every agent takes only one good in each remaining round. Therefore, each agent deterministically receives at least a 1/7-fraction of their MMS value.

We now consider the complementary case: there exists an index $i^*$ such that agents $1, \ldots, i^*$ are allocated more than one goods, and thereafter each agent is allocated one good. We split the analysis into two cases.

**Case 1:** $i \leq i^*$**.** Recall that $L_{ij}$ is the indicator random variable that good $j$ is available when agent $i$ arrives. In this case, using Claim 4.2 and applying union bound, the total probability that a good is allocated before round $i$ is $\frac{(i-1)}{4n}$. Therefore, $\mathbb{E}[L_{ij}] = 1 - \frac{i-1}{4n}$. Therefore, the total expected value left when agent $i$ arrives is $\mathbb{E}[v_i(L_i)] \geq \sum_{j \in S_i} \left(1 - \frac{i-1}{4n}\right) = \frac{k_i(4n-i+1)}{4}$. Now, using Claim 4.1, since $L_{ij}$ are exchangeable for a given $i$, we can use concentration of exchangeable random variables from 2.6. Recall the concentration bound states that for any *fixed* $w_1, \ldots, w_m$, $\epsilon_m \in O\left(\frac{\log m}{m}\right)$ and any exchangeable random variables $X_1, \ldots, X_m$,

$$\mathbb{P}\left[\sum_{j=1}^{m} w_j(x_j - \bar{x}) \geq \|w\|_2 \sqrt{2(1+\epsilon_m)\ \log(1/\delta)}\right] \leq \delta$$

We use this with $X_j \coloneqq L_{ij}$ and $w_j \coloneqq -1$ for all $j \in S_i$ and 0 otherwise. Further, we use $\delta = \frac{1}{n^2}$ and $\epsilon_m = 1$. We therefore get,

$$\mathbb{P}\left[\sum_{j \in [m]} w_j(L_{ij} - \mathbb{E}[L_{ij}]) \geq \|w\|_2 \sqrt{2(1+\epsilon_m)\log(1/\delta)}\right] \leq \delta$$

$$\implies \mathbb{P}\left[v_i(L_i) \leq \mathbb{E}[v_i(L_i)] - \|w\|_2 \sqrt{4 \log n^2}\right] \leq \frac{1}{n^2}$$

$$\implies \mathbb{P}\left[v_i(L_i) \leq \frac{(4n-i+1)k_i}{4} - \sqrt{8nk_i \log n}\right] \leq \frac{1}{n^2}.$$

Therefore, with probability at least $1 - \frac{1}{n^2}$, agent $i$ is left with a value of $\frac{(4n-i+1)k_i}{4} - \sqrt{8nk_i \log n} \geq \frac{3nk_i}{4} - \sqrt{8nk_i \log n}$ where the last inequality follows because $i \leq n$. Therefore, obtaining a value of $\frac{1}{4n}$ of the remaining value, the agent receives a value of $\lfloor \frac{3k_i}{16} - \sqrt{\frac{k_i \log n}{2n}} \rfloor$. By assumption, when these agents arrive, the set of leftover goods is at least $8n$ (so that they receive more than 1 good), so the MMS value of these agents is at least 8, hence each of these agents will receive a value of at least $\frac{1}{16}\mathsf{MMS}_i$, for all $n \geq 68$. Therefore, by union bound, the probability that *all* agents $\leq i^*$ get a value of $\frac{1}{16} - \mathsf{MMS}_i$ is at least $1 - \frac{i^*}{n^2}$.

**Case 2:** $i > i^*$**.** In this case, following the same calculations as above, after round $i^*$ for each agent $i \geq i^* + 1$, with probability at least $1 - \frac{1}{n^2}$, the number of goods left is at least $\frac{3nk_{i^*}}{4} - \sqrt{8nk_{i^*} \log n}$. Since this is the last round where we assign $\frac{1}{4n}$ of the leftover goods, $|L_{i^*}| \geq 4n$ implying $k_{i^*} \geq 4$, hence we have at least $3n - 4\sqrt{2n \log n}$ goods remained for the next iterations. Hence, as long as $n \geq 27$, $3n - 4\sqrt{2n \log n} \geq n$, and therefore the number of remaining goods is at least $n$, which is sufficient for the subsequent iterations as at most $n - 1$ iterations remain, and in each iteration every agent receives exactly one good. Therefore, putting both cases together, we get that with probability at least $\left(1 - \frac{1}{n}\right)$, all agents receive at least a value at least $\frac{1}{16} - \mathsf{MMS}_i$. □

## 4.2 General Binary Valuations

We keep the notation same as the previous section. Let $\mathrm{MMS}_i = k_i$ denote the MMS value of agent $i \in [n]$. Let $S_i$ be the set of goods she likes at a value of 1. Let $L_i$ be the set of goods that agent $i$ likes at a value of 1 that are left when $i$ arrives. Now, $S_i$ are no longer nested. We first state our very simple algorithm for this case. We note that the algorithm is almost similar to our algorithm

**ALGORITHM 3:** Algorithm for General Binary Valuations

```
Draw a random permutation $\sigma$ of the goods $[m]$ uniformly from all permutations.
for each arriving agent $i \in [n]$ do
    Allocate the smallest-indexed $\max\left\{\left\lfloor \frac{\mathrm{MMS}_i}{256} \right\rfloor, 1\right\}$ goods from $L_i$.
end
```

for nested binary. Instead of allocating a $\frac{1}{4n}$-fraction of remaining goods, we explicitly allocate an $\frac{\mathrm{MMS}_i}{256}$ goods. The reason for this will be clear in the analysis.

Our main guarantee for the algorithm is as follows.

Theorem 4.5. *Given an instance $\mathcal{I} = [n], [m], (v_i)_{i \in [n]}$ of* OnlineMMS *with binary valuations, Algorithm 3 is a $(\frac{1}{256}, \Theta(1))$-competitive algorithm.*

To prove this theorem, we follow the same high-level approach as in the previous section: we aim to show that, with high probability, when an agent $t \in [n]$ arrives, there remains sufficient value among the unallocated goods to guarantee her a good allocation.

As before, let $L_{tj}$ denote the indicator random variable for the event that good $j$ is available when agent $t$ arrives. Unlike in the nested-valuation setting, however, the random variables $(L_{tj})_{j \in [m]}$ are no longer exchangeable for a fixed agent $t$: different goods may be valued by different subsets of agents arriving before $i$, and hence are subject to different levels of competition. Consequently, such goods have different probabilities of remaining unallocated by the time agent $t$ arrives.

Formally, this asymmetry implies that the joint distribution of $(L_{tj})_{j \in [m]}$ is not invariant under permutations of the goods, and therefore the exchangeability property fails in the general case.

To circumvent this, we introduce a new algorithm in the analysis. This algorithm is specific to a certain agent and therefore, we have a different algorithm for different agents. This algorithm ensures that the indicator random variables corresponding to the leftover set of goods when the agent arrives are exchangeable and therefore, we can apply the concentration bound for exchangeable random variables. At the same time, we give a coupling of the number of goods that agent $t$ likes and are allocated when $t$ arrives in Algorithm 3 with the number of goods that agent $t$ likes and are allocated in the analysis algorithm. This allows us to give a bound on the value of the leftover items for agent $t$, when she arrives in Algorithm 3. The remaining parts of the analysis follow in a way similar to the previous section.

For the rest of the algorithm, fix an agent $t \in [n]$. We calculate the probability that agent $t$ receives a value of at least $\frac{\mathrm{MMS}_i}{256}$. Since we are fixing an agent $t \in [n]$, and the algorithm is only used in the analysis, we can assume that we know the number of agents for whom the MMS value was at most 256 and arrived before $t$. Let this number be $n_t$. Now, consider Algorithm 4. We introduce some more notation here.

**Notation for Algorithm 4.** As defined in the algorithm, $H$ is an arbitrary set of $n_t$ goods from $S_t$. For any agent $i \in [t]$, let $L'_i$ be the set of goods that agent $i$ likes from $S_i \setminus H$ and are left when agent $i$ arrives in Algorithm 4. Let $L'_{t,i}$ be the set of goods from $S_t \setminus \{S_i \cup H\}$ that are left when agent $i$ arrives in Algorithm 4. Note that these are leftover goods that are not liked by $i$, but are liked by $t$. Also note that this is different from the random variables $L_{tj}$ of Algorithm 3. Let $X'_{ij}$ be the indicator random variable denoting that good $j$ is available when agent $i$ arrives in Algorithm 4.

**ALGORITHM 4:** Algorithm for analyzing value of agent $t$

```
A random permutation $\sigma$ of the goods $R$ uniformly from all permutations.
Let $R \leftarrow [m] \setminus H$.
Draw a random permutation $\sigma$ of the goods $R$ uniformly from all permutations.
for each arriving agent $i \in [n]$ do
    if $\mathsf{MMS}_i < 256$ then
        Allocate a single good from $H$.
    end
    else
        Allocate the smallest-indexed $\max\left\{\left\lfloor \frac{MMS_i}{256} \right\rfloor, 1\right\}$ goods from $(L'_i \cup L'_{t,i})$.
    end
end
```

For any permutation $\sigma_1$ of the $m$ goods and any subset $H \subseteq S_t$ with $|H| = n_t$, let $f_H(\cdot)$ denote the mapping that produces the unique permutation $\sigma_2$ over $[m] \setminus H$ obtained by deleting the goods in $H$ from $\sigma_1$; that is, $f_H(\sigma_1) = \sigma_2$. For any pair $(\sigma_1, f_H(\sigma_1))$, we show that if running Algorithm 3 on $\sigma_1$ leaves $L_t$ as the set of goods in $S_t$ that remain unallocated upon agent $t$'s arrival, and running Algorithm 4 on $f_H(\sigma_1)$ leaves $L'_t$ as the corresponding leftover set, then

$$|S_t| - |L_t| \leq 2\big(|S_t| - |L'_t|\big).$$

Equivalently, up to round $t$, the number of goods from $S_t$ allocated before agent $t$ arrives in the original algorithm is at most twice the number allocated under the analysis algorithm. Formally, we state the following lemma.

**Lemma 4.6.** *For any pair of permutations $\big(\sigma_1, f_H(\sigma_1)\big)$, let $L_t$ (resp., $L'_t$) denote the set of goods in $S_t$ that remain unallocated upon agent $t$'s arrival when running Algorithm 3 on $\sigma_1$ (resp., Algorithm 4 on $f_H(\sigma_1)$). Then*

$$|S_t| - |L_t| \leq 2\big(|S_t| - |L'_t|\big).$$

Proof Sketch. Consider the case where $H = \emptyset$. We can view the charging argument as a table where the columns correspond to the goods and the rows correspond to the agents. If good $j$ is allocated to agent $i$ by Algorithm 3, we color the $(i, j)$ cell blue. Any good $j$ allocated to agent $i$ by Algorithm 4 is colored red. To show that we allocated at most twice the number of goods of $S_t$ in Algorithm 3 as compared to Algorithm 4, we charge a blue cell for any $j \in S_t$ to a red cell of some $j' \in S_t$. At a high level, this charging is done as follows. If a good $j \in S_t$ allocated by Algorithm 3 is allocated to an agent $i' < i$ by Algorithm 4, we charge it to $(i', j)$. Otherwise, the only reason Algorithm 4 does not allocate this good is because it has completed its allocation to $i$ before reaching $j$. Therefore we try to find a unique good in the same row from $S_t$ that is red colored. If we do not find that, we map it to some unique good $j'$ that Algorithm 4 allocated to $i$. Since this good is not in $S_t$, it must be in $S_i$ and the reason Algorithm 3 didn't allocate it to $i$ is because it was allocated to some agent $i' < i$. We charge this therefore to $(i', j')$. Now, $j'$ is allocated to agent $i'$ by Algorithm 3 but not by Algorithm 4. Therefore, we are again in the same case where we are trying to charge a blue cell to a red cell in same row or redirect it to a unique red cell. Since the arrows only go leftwards or upwards, this process must end at a unique red cell. □

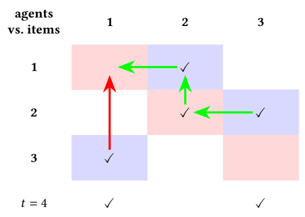

Fig. 1. An illustration of how the charging argument to couple number of allocated goods of agent $t$ from Algorithm 3 and Algorithm 4.

Recall that $X'_{tj}$ is the indicator random variable for the event that good $j$ is available when agent $t$ arrives. We show the following claim that the variables $(X'_{tj})_{j\in S_t\setminus H}$ are exchangeable.

**Claim 4.7.** *For a fixed $H \in S_t$, and a uniformly random $\sigma_1$, consider $(\sigma_1, f_H(\sigma_1))$, the process of Algorithm 4, the variables $(X'_{tj})_{j\in S_t\setminus H}$ are exchangeable.*

Proof. Note that as $\sigma_1$ is uniformly random, and $H$ is a specific subset removed from $\sigma_1$, $f_H(\sigma_1)$ also follows a uniformly random distribution of the goods $[m] \setminus H$. Restricting $\sigma$ to the subset $S_t \setminus H$, the induced permutation on $S_t \setminus H$ is therefore uniform over all permutations of $S_t \setminus H$. All agents arriving before agent $t$ value every good in $S_t \setminus H$. Moreover, the allocation rule for each agent depends only on the relative order of the remaining goods under $\sigma$, and not on their identities. Consequently, the allocation decisions of agents $1, \dots, t-1$ treat all goods in $S_t \setminus H$ symmetrically. Formally, for any permutation $\pi$ of $S_t \setminus H$, the joint distribution of the availability indicators $(L'_{tj})_{j\in S_t\setminus H}$ is identical to that of $(L'_{t,\pi(j)})_{j\in S_t\setminus H}$. This invariance under permutations of $S_t \setminus H$ implies that the random variables $(L'_{tj})_{j\in S_t\setminus H}$ are exchangeable. □

We are now ready to show that we can bound the probability of failure of Algorithm 3 with a constant.

To do this, first we introduce some notation. We keep a parameter $x \in [0, 1]$. We define:

- An agent $t$ is *good* if when agent $t$ arrives, the number of remaining goods of the agent $t$ is at least $(1-x)(n-n_t)k_t$ when $k_t \geq 2$ and is at least 1 if $k_t = 1$ in the run of Algorithm 3.
- $T(t)$ : We define $T(t)$ to be the smallest probability that agents 1 to $t$ in Algorithm 3 are *good* among all possible instances $\mathcal{I}$.

Recall that $n_t$ is the number of special rounds that occur before agent $t$ arrives and $k_t$ is the MMS value of agent $t$. We state the following two lemmas for bounding the difference between $T(t)$ and $T(t-1)$.

**Lemma 4.8.** *Fix an agent $t \in [n]$. For $x = \frac{1}{4}$, we have $T(t) \geq T(t-1) - \frac{1}{2^{2(n-n_t)}}$ if $k_t \geq 2$ and $T(t) \geq T(t-1) - \frac{1}{2^{\frac{n-n_t}{10}}}$ if $k_t = 1$.*

Proof Sketch. The proof of this lemma interleaves between Algorithm 3 and Algorithm 4. We define failure of algorithm as some agent is not good. We show that if Algorithm 3 does not fail for $t-1$ agents then Algorithm 4 will also not fail for $t-1$ agents. Then since Algorithm 4 maintains exchangeability of the variables of agent $t$, we can show that Algorithm 4 does not fail for $t+1$ agents. This allows us to use the charging argument to show that Algorithm 3 does not fail for $t+1$ agents. We proceed in this way, bounding the probability of failure with larger numbers as we move further into the algorithm. Full proof is in Appendix C.2.3. □

The bound of the above lemma is very weak for the last few agents and we will not be able to union bound to a reasonably small number. We therefore give a different bound in the next lemma.

**Lemma 4.9.** *Fix an agent $t \in [n-79, n]$. For $x = \frac{1}{4}$, we have $T(t) \geq T(t-1) - \frac{1}{2^{2(n-n_t)}}$ if $k_t \geq 2$ and $T(t) \geq T(t-1) - \frac{1}{256(1-x)}$ if $k_t = 1$.*

Proof Sketch. For the last few agents, when $k_t \geq 2$, the bound from previous lemma works. For the agents with $k_t = 1$, we simply use union bound to guarantee the probability of failure. Full proof is in Appendix C.2.4. □

Proof of Theorem 4.5. Combining the previous two claim, we can compute that the total probability of success of Algorithm 3 is

$$T(n) \geq T(1) - \Big( \sum_{t\in[n-80]} \frac{1}{2^{\frac{n-n_t}{10}}} + \sum_{t\in[n-79,n]} \max\Big(\frac{1}{192}, \frac{1}{2^{2(n-n_t)}}\Big) \tag{1}$$

$$\geq 1 - \Big( \sum_{t\in[n-80]} \frac{1}{2^{\frac{n-t+1}{10}}} + \sum_{t\in[n-79,n]} \max\Big(\frac{1}{192}, \frac{1}{2^{2(n-t+1)}}\Big)\Big) = \Theta(1) \tag{2}$$

□

## References

Hannaneh Akrami and Jugal Garg. 2024. Breaking the 3/4 barrier for approximate maximin share. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 74–91.

Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: Analysing a food bank problem. *arXiv preprint arXiv:1502.07571* (2015).

Martin Aleksandrov and Toby Walsh. 2020. Online fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13557–13562.

Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence* 322 (2023), 103965.

Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A Voudouris. 2022. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2202.07551* (2022).

Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. 2017. Algorithms for max-min share fair allocation of indivisible chores. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*. 335–341.

Siddhartha Banerjee, Chamsi Hssaine, and Sean R Sinclair. 2023. Online fair allocation of perishable resources. *ACM SIGMETRICS Performance Evaluation Review* 51, 1 (2023), 55–56.

Nikhil Bansal and Maxim Sviridenko. 2006. The Santa Claus problem. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (Seattle, WA, USA) *(STOC '06)*. Association for Computing Machinery, 31–40. https://doi.org/10.1145/1132516.1132522

Rina Foygel Barber. 2024. Hoeffding and Bernstein inequalities for weighted sums of exchangeable random variables. *Electronic Communications in Probability* (2024). https://doi.org/10.1214/24-ecp620

Siddharth Barman and Sanath Kumar Krishnamurthy. 2020. Approximation Algorithms for Maximin Fair Division. arXiv:1703.01851 [cs.GT] https://arxiv.org/abs/1703.01851

Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 7–13.

Gerdus Benade, Aleksandr M Kazachkov, Ariel D Procaccia, and Christos-Alexandros Psomas. 2018. How to make envy vanish over time. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 593–610.

Anna Bogomolnaia, Hervé Moulin, and Fedor Sandomirskiy. 2019. A simple online fair division problem. *arXiv preprint arXiv:1903.10361* 1 (2019).

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

Steven J Brams and Peter C Fishburn. 1978. Approval voting. *American Political Science Review* 72, 3 (1978), 831–847.

Steven J. Brams and Alan D. Taylor. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press.

Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

Kate Donahue and Jon Kleinberg. 2020. Fairness and utilization in allocating resources with uncertain demand. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 658–668.

Uriel Feige. 2022. Maximin fair allocations with two item values. https://www.wisdom.weizmann.ac.il/~feige/mypapers/MMSab.pdf.

Uriel Feige, Ariel Sapir, and Laliv Tauber. 2021. A tight negative example for MMS fair allocations. arXiv:2104.04977 [cs.GT] https://arxiv.org/abs/2104.04977

Eric Friedman, Christos-Alexandros Psomas, and Shai Vardi. 2015. Dynamic fair division with minimal disruptions. In *Proceedings of the sixteenth ACM conference on Economics and Computation*. 697–713.

Eric Friedman, Christos-Alexandros Psomas, and Shai Vardi. 2017. Controlled dynamic fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 461–478.

Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2018. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 539–556.

Daniel Halpern, Ariel D Procaccia, Alexandros Psomas, and Nisarg Shah. 2020. Fair division with binary valuations: One rule to rule them all. In *International Conference on Web and Internet Economics*. Springer, 370–383.

Daniel Halpern, Alexandros Psomas, Paritosh Verma, and Daniel Xie. 2025. Online Envy Minimization and Multicolor Discrepancy: Equivalences and Separations. *arXiv preprint arXiv:2502.14624* (2025).

Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. 2019. Achieving a Fairer Future by Changing the Past. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, 343–349.

Ian Kash, Ariel D Procaccia, and Nisarg Shah. 2014. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research* 51 (2014), 579–603.

Emil Kauder. 2015. *History of marginal utility theory*. Princeton University Press.

Pooja Kulkarni, Rucha Kulkarni, and Ruta Mehta. 2024. Approximating APS Under Submodular and XOS Valuations with Binary Marginals. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 1057–1065.

Pooja Kulkarni, Ruta Mehta, and Parnian Shahkar. 2025. Online Fair Division: Towards Ex-Post Constant MMS Guarantees. In *Proceedings of the 26th ACM Conference on Economics and Computation*. 638–638.

Hervé Moulin. 2004. *Fair division and collective welfare*. MIT press.

Hervé Moulin. 2019. Fair division in the internet age. *Annual Review of Economics* 11, 1 (2019), 407–441.

Ariel D Procaccia, Benjamin Schiffer, and Shirley Zhang. 2024. Honor among bandits: No-regret learning for online fair division. *arXiv preprint arXiv:2407.01795* (2024).

Ariel D Procaccia and Junxing Wang. 2014. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*. 675–692.

Parnian Shahkar and Jugal Garg. 2025. Improved MMS Approximations for Few Agent Types. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, James Kwok (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4057–4064. https://doi.org/10.24963/ijcai.2025/452 Main Track.

Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. 2022. Sequential fair allocation: Achieving the optimal envy-efficiency tradeoff curve. *ACM SIGMETRICS Performance Evaluation Review* 50, 1 (2022), 95–96.

Gilad Ben Uziahu and Uriel Feige. 2023. On Fair Allocation of Indivisible Goods to Submodular Agents. arXiv:2303.12444 [cs.GT] https://arxiv.org/abs/2303.12444

Jan Vondrak. 2010. A note on concentration of submodular functions. arXiv:1005.2791 [cs.DM] https://arxiv.org/abs/1005.2791

Toby Walsh. 2011. Online cake cutting. In *Algorithmic Decision Theory: Second International Conference, ADT 2011, Piscataway, NJ, USA, October 26-28, 2011. Proceedings 2*. Springer, 292–305.

Hakuei Yamada, Junpei Komiyama, Kenshi Abe, and Atsushi Iwasaki. 2024. Learning Fair Division from Bandit Feedback. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3106–3114.

David Zeng and Alexandros Psomas. 2020. Fairness-efficiency tradeoffs in dynamic fair division. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 911–912.

Shengwei Zhou, Rufan Bai, and Xiaowei Wu. 2023. Multi-agent online scheduling: MMS allocations for indivisible items. In *International Conference on Machine Learning*. PMLR, 42506–42516.

## A Normalizations of Definition 2.4 Without Loss of Generality

First consider normalization for additive valuations as given in Definition 2.4. Given a non-normalized input instance $\mathcal{I} = ([n],[m],\{v_i\}_{i\in[n]})$, for any valuation function $v_i$, the corresponding normalized valuation $v_i'$ is constructed by computing an MMS partition $P^i = (P_1^i,\dots,P_n^i)$ for $v_i$ and then rescaling the valuations so that for every $i \in [n]$ and for every good $j \in P_k^i$, $v_i'(j) = \frac{v_i(j)}{v_i(P_k^i)}$. Thereby, the corresponding normalized input instance is defined as $\mathcal{I}' = ([n],[m],\{v_i'\}_{i\in[n]})$. Lemma 4 in [] shows that for any set of goods $B \subseteq [m]$, $v_i(B) \geq v_i'(B)\cdot \mathsf{MMS}_{v_i}^n$. By construction, the MMS value of every type in the normalized instance is 1, i.e. for all $i \in [n]$, $\mathsf{MMS}_{v_i'}^n = 1$. Therefore, if $v_i'(B) \geq \alpha\mathsf{MMS}_{v_i'}^n$, we have that $v_i(B) \geq \alpha \cdot \mathsf{MMS}_{v_i}^n$. This implies that an $\alpha$ approximation of the MMS value according to a normalized instance guarantees that each agent receives a bundle valued at least $\alpha$ approximation of her MMS value in the original (non-normalized) instance.

Next, consider normalization for submodular valuations as given in Definition 2.4. Here we simply create a new valuation by assuming $v_i'(S) = \frac{1}{\mathrm{MMS}_i^n} v_i(S)$. For the MMS partition, this scales each bundle to have value at least 1 and the smallest of these bundles has value 1. On the other hand, if the normalized instance had an allocation such that the value of each bundle in this partition is more than 1, the original instance has a partition where the value of each bundle is more than $\mathsf{MMS}_i^n$ which is not possible. Therefore, we can assume wlog that instances are normalized.

## B Fractional Extensions for Submodular Functions

**Multilinear Extension of Submodular Valuations.** The multilinear extension of a submodular function models the expected value an agent receives when goods are allocated independently according to the marginal probabilities $(p_j)_{j\in[m]}$.

**Definition B.1** (Multilinear Extension). Let $f : 2^{[m]} \to \mathbb{R}_+$ be a set function. The multilinear extension of $f$, denoted by $F : [0,1]^m \to \mathbb{R}_+$, is defined for $\vec{p} \in [0,1]^m$ as

$$F(\vec{p}) \;=\; \sum_{S\subseteq[m]} f(S) \prod_{j\in S} p_j \prod_{j\in[m]\setminus S} (1-p_j).$$

Equivalently,

$$F(\vec{p}) = \mathbb{E}[f(R)],$$

where $R$ is a random subset of $[m]$ obtained by including each element $j \in [m]$ independently with probability $p_j$.

**Concave Extension of Submodular Valuations.** The concave extension of a submodular valuation function models the maximum expected value an agent can receive under any correlated rounding scheme that preserves the marginal probabilities of each good.

**Definition B.2** (Concave Extension). Let $f : 2^{[m]} \to \mathbb{R}_+$ be a submodular function. The concave extension of $f$, denoted by $f^+$, is defined for $\vec{p} \in [0,1]^m$ as

$$f^+(\vec{p}) \;=\; \max\left\{ \sum_{S\subseteq[m]} \alpha_S f(S) \;\middle|\; \sum_{S\subseteq[m]} \alpha_S = 1,\; \sum_{S: j\in S} \alpha_S = p_j \;\; \forall j \in [m],\; \alpha_S \geq 0 \right\}.$$

It is known that for any submodular function $f$ defined over a ground set $[m]$, $F(x) \geq \left(1-\frac{1}{e}\right) f^+(x)$ for all $x \in [0,1]^m$ i.e., the correlation gap is bounded by $\left(1-\frac{1}{e}\right)$.

## C Omitted Proofs

### C.1 Proofs From Section 3

#### *C.1.1 Proof of Lemma 3.3.*

Proof. We'll use $t_i$ to denote the number of special rounds that happens before agent $i$, $S_i$ to denote the set of goods that is not allocated in those $t_i$ special rounds (note this is not $R_{i-1}$). We'll use $H_i$ to denote the set of goods that is considered high-valued by agent $i$ in $S_i$ and $h_i = |H_i|$.

Note there are exactly $i - 1 - t_i$ non-special rounds before agent $i$. Since $t_i \leq n - \frac{4}{\alpha}$, each of these non-special rounds allocates any good with probability at most $\frac{1}{2(n-t_i)}$ since each time we are allocating with probability $\frac{1}{2(n-t)}$ for some $t \leq t_i$. Therefore, using union bound, each good in $S_i$ is allocated with probability at most $\frac{i-1-t_i}{2(n-t_i)} < \frac{1}{2}$. Thus for every good $j \in S_i$, $\mathbb{P}[j \in R_{i-1}] > \frac{1}{2}$.

We denote $\mathcal{I}_1$ as the indicator of that there is no good in $R_{i-1}$ which is considered high-valued by agent $i$ and $\mathcal{I}_2$ as the indicator of that if we allocate all the low-valued goods in $R_{i-1}$ with probability $\frac{1}{2(n-t_i)}$, the value of this random set is below $\alpha$. If $\mathcal{I}_1 \neq 1$, then we have at least one high-valued good left, thus we'll allocate a high-valued good to $i$. If $\mathcal{I}_2 \neq 1$, then if we are allocating low-valued goods to agent $i$, the random set will have value more than $\alpha$. Thus $p_i \geq 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1])$

Observe that $v_i(S_i) \geq n - t_i > n - (n - \frac{4}{\alpha}) = \frac{4}{\alpha}$ since $S_i$ only loses $t_i$ goods from $[m]$ and each of them has value at most 1.

**Case 1:** $|H_i| = h_i \geq \frac{2}{\alpha}$. Since each good in $H_i$ will remain in $R_{i-1}$ with probability at least $\frac{1}{2}$, $\mathbb{P}[I_1 = 1] < (\frac{1}{2})^{h_i} \leq \frac{1}{n^2}$. Thus $p_i \geq 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \geq 1 - \mathbb{P}[\mathcal{I}_1 = 1] > 1 - \frac{1}{n^2}$.

**Case 2:** $|H_i| = h_i < \frac{2}{\alpha}$. Then the total value of small-valued goods in $S_i$ is at least $v_i(S_i) - h_i > n - t_i - h_i$. Since we allocate every good with probability $\frac{1}{2(n-t_i)}$, we can compute $\mathbb{E}[v_i(A_i)] = \frac{1}{2}(n - t_i - h_i)\frac{1}{2(n-t_i)} \geq \frac{1}{8}$. The last inequality is because $n - t_i \geq \frac{4}{\alpha}$ while $h_i < \frac{2}{\alpha}$.

$$\begin{aligned}
\mathbb{P}[\mathcal{I}_2 = 1] &= \mathbb{P}[v_i(A_i) < \alpha] \\
&\leq \mathbb{P}[v_i(A_i) < 8\alpha \cdot \mathbb{E}[v_i(A_i)]] \\
&\leq \exp\left(-\frac{(1-8\alpha)^2(1/8)}{\alpha}\right) \qquad \text{(Applying Lemma 2.5)} \\
&\leq \frac{1}{n^2} \qquad \text{(since } \alpha \leq 1/128\log n\text{))}.
\end{aligned}$$

Therefore, $p_i \geq 1 - \min(\mathbb{P}[\mathcal{I}_1 = 1], \mathbb{P}[\mathcal{I}_2 = 1]) \geq 1 - \mathbb{P}[\mathcal{I}_2 = 1] > 1 - \frac{1}{n^2}$. □

### C.2 Proofs from Section 4

#### *C.2.1 Proof of Lemma 4.6.*

Proof. To prove this claim, we use a *charging* argument. Specifically, we construct a bipartite graph $G_t = (U_t, V_t, E_t)$, where both left and right side contain a distinct copy of the goods in $S_t$. For each good $i \in U_t$, we add an edge to an good $j \in V_t$ whenever $i$ is allocated by the original algorithm, and $j$ is allocated by the analysis algorithm; in this case, we say that $i$ is *charged* to $j$. We show that every good $j \in V_t$ receives charge from at most two goods in $U_t$. It follows that the number of goods allocated from $S_t$ by the original algorithm (i.e., the number of charged goods in $U_t$) is at most twice the number of goods allocated from $S_t$ in the right copy (under the analysis algorithm).

First, we construct an auxiliary bipartite graph $G = (U, V, E)$, where each side contains a copy of the good set $[m]$. Next, consider a $([t-1] \times [m])$-grid in which each row corresponds to an agent in $[t-1]$ and each column corresponds to an good, ordered according to the permutation $\sigma_1$. For any good $j$ that is allocated to agent $i$ by Algorithm 3, we color the cell $(i, j)$ in the grid blue; for any good $j$ that is allocated to agent $i$ by Algorithm 4, we color the cell $(i, j)$ red. If the same good $j$ is allocated to the same agent $i$ under both algorithms, then the cell $(i, j)$ is colored *both* red and blue.

Fix an agent $i \in [t-1]$. Partition the goods allocated to $i$ by Algorithm 3 (i.e., the blue goods) into three sets according to how Algorithm 4 allocates them: $S^1$, $S^2$, and $S^3$. Here, $S^1$ consists of the goods that Algorithm 4 allocates to some earlier agent, $S^2$ consists of the goods that Algorithm 4 allocates to agent $i$, and $S^3$ contains all remaining goods.

We charge every good in $S^1 \cup S^2$ to itself; equivalently, in the graph $G$ we add an edge from the copy of each $j \in S^1 \cup S^2$ in the left partition $U$ to its corresponding copy in the right partition $V$. For each $j \in S^1$, by definition there exists an $i' < i$ such that Algorithm 4 allocates $j$ to $i'$. In the grid, we draw an *upward* arrow from $(i, j)$ to $(i', j)$. For each $j \in S^2$, we draw a self-loop at $(i, j)$. Thus, these arrows appear as either self-loops or vertical upward arrows from blue cells to red cells. Let $m_v(\cdot)$ denote the (vertical) mapping induced by this charging; in particular, since goods in $S^1 \cup S^2$ are charged to themselves, we have $m_v(j) = j$ whenever $j$ is vertically charged. Each good is vertically charged at most once, so $m_v$ is injective; consequently, $m_v^{-1}(j) = j$ for every vertically charged good $j$.

Now consider any good $j \in S^3$. Such an good is allocated to agent $i$ by the original algorithm, so $j \in S_i$. Moreover, since $j \in S^3$, it is not allocated to any agent $i' < i$ under Algorithm 4, therefore when agent $i$ arrives, good $j$ is still available under Algorithm 4, which implies $j \in L_i'$. Hence, the only reason $j$ is not selected by Algorithm 4 is dictated by the value of $\mathsf{MMS}_i$, leading to the following two cases.

**(a)** If $\mathsf{MMS}_i < 256$, then under Algorithm 4 agent $i$ is allocated an arbitrary good $j' \in H$. In this case, we charge $j$ to $j'$.

**(b)** If $\mathsf{MMS}_i \geq 256$, then agent $i$ receives sufficiently many goods of smaller index under Algorithm 4; in particular, every good allocated to $i$ by Algorithm 4 has index smaller than $j$. Since every good in $S^2$ is allocated to agent $i$ by *both* algorithms, let $S'$ denote the set of goods allocated to agent $i$ by Algorithm 4, excluding those in $S^2$. Because the two algorithms allocate the same total number of goods to agent $i$, we have

$$|S'| = |S^1| + |S^3|,$$

and in particular $|S'| \geq |S^3|$. Thus, we can charge each $j \in S^3$ to a distinct good $j' \in S'$.

In both cases, for every $j \in S^3$ we charge $j$ to a unique good $j' \in S'$. Equivalently, we add a directed edge $(j, j')$ in the graph $G$, and draw a horizontal arrow in the grid from $(i, j)$ to $(i, j')$. In case **(b)**, since $j'$ has smaller index than $j$, this arrow is *leftward* (from a blue cell to a red cell); in case **(a)**, the horizontal arrow may point either left or right.

Let $m_h(\cdot)$ denote the horizontal mapping induced by this charging; namely, if good $j$ is charged to $j'$ horizontally, we set $m_h(j) = j'$. Since each good is charged horizontally at most once, $m_h$ is injective. Hence, for every $j'$ in the image of $m_h$, the inverse is well-defined and satisfies $m_h^{-1}(j') = j$.

After applying the above charging procedure to every row $i \in [t-1]$ of the grid, each blue cell has exactly one outgoing arrow in the grid, either vertical-upward, or horizontal-leftward (with the exception of horizontal arrows pointing to an good in $H$ which might be rightward). Equivalently, in the bipartite graph $G$, the left copy of every good allocated to some agent before $t$ under Algorithm 3 has exactly one outgoing edge to a right copy of an good that is allocated to some agent before $t$ under Algorithm 4. Moreover, each right copy receives at most two incoming edges: due to injectivity of both horizontal and vertical mappings, an good can be charged vertically at most once and horizontally at most once. Hence, in graph $G = (U, V, E)$, every node in $V$ has in-degree at most 2, while every node in $U$ has out-degree exactly 1 by construction. It follows that the number of goods allocated under Algorithm 3 is at most twice the number of goods allocated under Algorithm 4.

Given this auxiliary graph $G$, we obtain $G_t$ by restricting $G$ to the goods in $S_t$; that is, $G_t$ contains only copies of the goods in $S_t$ in both partitions.

To construct $G_t$, we charge every good $j \in S_t$ that is allocated to some agent $i \in [t-1]$ under Algorithm 3 to (the right copy of) another good in $S_t$ that is allocated to some agent $i \in [t-1]$ under Algorithm 4. We begin from the unique outgoing edge of $j$ in the auxiliary graph $G$. Recall that every such $j$ is charged either vertically or horizontally.

If $j$ is charged vertically, then the vertical mapping is the identity, so $m_v(j) = j \in S_t$. In this case, we simply add the edge from the copy of $j$ in $U_t$ to the copy of $j$ in $V_t$.

If $j$ is charged horizontally and $m_h(j) \in S_t$, then we add the edge from the copy of $j$ in $U_t$ to the copy of $m_h(j)$ in $V_t$. In particular, when $\mathsf{MMS}_i < 256$, we have $m_h(j) \in H \subseteq S_t$, so this condition holds and we add the edge $(j, m_h(j))$ to $G_t$.

It remains to consider the case where $j$ is charged horizontally but $m_h(j) \notin S_t$. This can occur only when $\mathsf{MMS}_i \geq 256$, in which case the grid contains a leftward arrow from the blue cell $(i, j)$ to the red cell $(i, m_h(j))$, and the index of $m_h(j)$ is smaller than that of $j$. Since $m_h(j)$ is allocated to agent $i$ under Algorithm 4, the good $m_h(j)$ is available upon $i$'s arrival in that run; moreover, because $m_h(j) \notin S_t$, it must be that $m_h(j) \in L'_i \subseteq S_i$, and hence $m_h(j) \in S_i$.

Now, $m_h(j)$ is liked by agent $i$ and has smaller index than $j$, yet it is not allocated to $i$ under Algorithm 3. The only possibility is that $m_h(j)$ was already allocated to some earlier agent $i' < i$ under Algorithm 3. Thus, $(i', m_h(j))$ is a blue cell in the grid, and $m_h(j) \in L_{i'} \subseteq S_{i'}$.

Under Algorithm 4, however, the same good $m_h(j)$ is allocated to agent $i > i'$ even though agent $i'$ likes it. This can happen only for one of the following reasons:

- **(a)** $\mathsf{MMS}_{i'} < 256$, in which case $m_h(j)$ is charged horizontally to some good in $H \subseteq S_t$, and hence $m_h(m_h(j)) \in S_t$.
- **(b)** $\mathsf{MMS}_{i'} \geq 256$, in which case $m_h(j)$ is not taken by $i'$ under Algorithm 4 because sufficiently many smaller-index goods are already allocated to $i'$. Consequently, the grid contains a leftward arrow from $(i', m_h(j))$ to the red cell $(i', m_h(m_h(j)))$.

In either case, we can *redirect* the charge from $m_h(j)$ to $m_h(m_h(j))$. If $m_h(m_h(j)) \in S_t$, we add the edge from the left copy of $j$ to the right copy of $m_h(m_h(j))$ in $G_t$. Otherwise, we iterate this redirection: continue applying $m_h$ until reaching the smallest $\ell \geq 1$ such that $m_h^\ell(j) \in S_t$, where $m_h^\ell$ denotes $\ell$ consecutive applications of $m_h$. We then add the edge from the left copy of $j$ to the right copy of $m_h^\ell(j)$ in $G_t$.

This procedure must terminate. Each redirection step follows an arrow in the grid that is either upward or leftward, unless it reaches an good in $H$, in which case it immediately lands in $S_t$ and halts. Since the grid is finite and the process never revisits a cell (we strictly move upward or leftward), it cannot continue indefinitely. Therefore, for every $j \in S_t$ there exists a minimum $\ell$ such that $m_h^\ell(j) \in S_t$, and the redirected charge is well-defined.

Finally, after applying the above recharging procedure and constructing $G_t = (U_t, V_t, E_t)$, we observe that each node $j \in V_t$ receives at most one horizontal charge. Indeed, since $m_h(\cdot)$ is injective, no good can be the horizontal image of two distinct goods. Moreover, if $j \in V_t$ receives a (possibly redirected) horizontal charge, the corresponding charged good can be identified by applying $m_h^{-1}(\cdot)$ repeatedly until reaching the first preimage that lies in $S_t$. In addition, each $j \in V_t$ receives at most one vertical charge, because vertical charging maps an good only to itself, i.e., it can only originate from the copy of $j$ in $U_t$. Consequently, every node in $V_t$ has in-degree at most 2.

Note that in $G_t$, the left copy of every good in $S_t$ that is allocated during the first $t-1$ rounds by Algorithm 3 has an outgoing edge (i.e., is charged) to the right copy of an good in $S_t$ that is allocated during the first $t-1$ rounds by Algorithm 4. Since each node on the right has in-degree at most 2, the number of goods from $S_t$ allocated in the first $t-1$ rounds under Algorithm 3 is at most twice the number allocated from $S_t$ in the first $t-1$ rounds under Algorithm 4. This yields

$$|S_t| - |L_t| \;\le\; 2\big(|S_t| - |L'_t|\big),$$

as claimed. In Section C.2.2 we will provide an example to illustrate how charging works. □

*C.2.2 Example for charging argument.* Consider an illustrative instance in which the analysis algorithm is evaluated at $t = 4$. Thus, we focus on the items allocated to agents $\{1, 2, 3\}$ by Algorithms 3 and 4. In Figure 2, each row corresponds to an agent and each column corresponds to an item (we display only the top three items in the ranking). A tick in cell $(i, j)$ indicates that agent $i$ likes item $j$, i.e., $j \in S_i$. From the figure, we have $2 \in S_1$, $\{2, 3\} \subseteq S_2$, $1 \in S_3$, and $\{1, 3\} \subseteq S_4$.

Assume that each of the first three agents must receive exactly one item. Under Algorithm 3, each agent $i \in [3]$ receives the first unallocated item she likes. Hence, item 2 is allocated to agent 1, item 3 to agent 2, and item 1 to agent 3; we color the corresponding cells blue. In contrast, under Algorithm 4, each agent $i \in [3]$ receives the first unallocated item from $S_i \cup S_t$. Consequently, item 1 is allocated to agent 1, item 2 to agent 2, and item 3 to agent 3; we color the corresponding cells red.

We now apply the charging argument, which charges each blue cell to a red cell. Executing the charging procedure yields the following charges: item 2 is charged to item 1, item 1 is charged to itself, and item 3 is charged to item 2, these charges through horizontal and vertical mappings are shown with red arrows in Figure 2. The resulting bipartite graph $G$, restricted to items $\{1, 2, 3\}$, is shown in Figure 3.

To pass from $G$ to $G_t$, which is restricted to items in $S_t$, we must *recharge* the edge emanating from item 3, since its current charge target is item 2, and $2 \notin S_t$. Observe that item 2 cannot be allocated to agent 2 under Algorithm 3 because it has already been allocated to agent 1. We therefore follow the existing charge of item 2, namely, the item to which 2 is charged, which is item 1. Consequently, we redirect the charge of item 3 from item 2 to item 1. This redirection is illustrated by the green arrows in Figure 4.

The resulting graph $G_t$, restricted to items $[3]$, is shown in Figure 5. After this procedure, every blue cell $(i, j)$ with $j \in S_t$ is charged to a red cell $(i', j')$ with $j' \in S_t$, as required.

*C.2.3 Proof of Lemma 4.8.*

Proof. Recall that $T(t)$ is the probability that agents 1 to $t$ in Algorithm 3 are good and $T(t-1)$ is the probability that agents 1 to $t-1$ in Algorithm 3 are good.

First note that, Algorithm 4 removes $n_t$ goods to allocate to the special agents upfront. After this, for the remaining non-special agents, the MMS value can only go up in the reduced instance. Further, Algorithm 4 also includes $S_t \setminus H_t$ in the set of goods that the non -special agents like further potentially increasing the set of goods liked by these agents. As a result, the run of Algorithm 4 never fails on the special days and on non-special days, the failure probability can only decrease. Therefore, Algorithm 4's failure probability for the first $t-1$ agents is at most $T(t-1)$.

Therefore, we have that with probability $T(t-1)$, each non-special agent $i < t$ has at least $(1-x)(n-n_t)k_i$ goods. The agents are taking $\frac{k_i}{256}$ goods out of these goods. Therefore, the probability of allocation of a good $j \in S_t \setminus H_t$ in one round is

$$\mathbb{P}\,[\text{good j allocated in one round}] \le \frac{k_i/256}{(1-x)(n-n_t)k_i}.$$

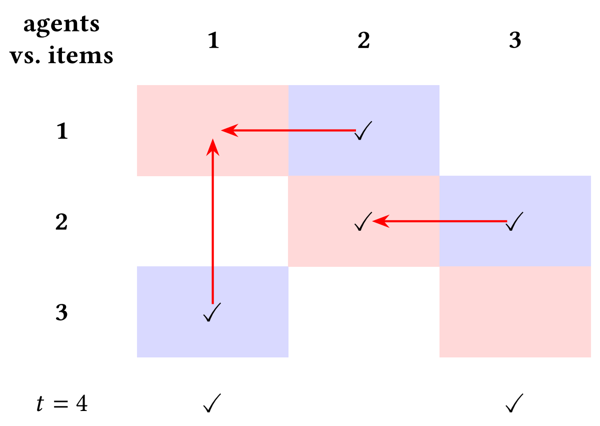

Fig. 2. Running example at $t = 4$: grid representation of preferences and allocations in the first $t - 1 = 3$ rounds. Rows correspond to agents $1, 2, 3$ and columns to items (shown in rank order, truncated to the top three items). A tick in cell $(i, j)$ indicates $j \in S_i$. Blue cells denote allocations made by Algorithm 3, and red cells denote allocations made by Algorithm 4. Red arrows indicate the initial charging; each blue cell $(i, j)$ is charged to a red cell $(i', j')$ specified by the construction.

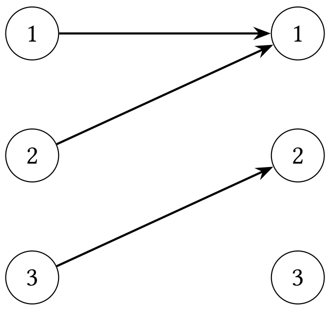

Fig. 3. Auxiliary bipartite graph $G = (U, V, E)$ induced by the charging procedure for the running example, restricted to items $\{1, 2, 3\}$. An edge $(j, j')$ indicates that the left copy of item $j$ (allocated under Algorithm 3) is charged to the right copy of item $j'$ (allocated under Algorithm 4).

As a results, we can union bound the probability that any good $j \in S_t \setminus H_t$ is available agent $t$ as

$$\mathbb{P}\left[\text{good j is available for agent } t\right] \geq 1 - \frac{1}{256(1-x)}.$$

Therefore,

$$\mathbb{E}[v_t(A_t)] \geq \left(1 - \frac{1}{256(1-x)}\right)(nk_t - n_t)$$

Now, recall that the indicator random variables $X'_{jt}$ are exchangeable. Therefore, we can show concentration of the agent $t$'s value around its expected value and obtain that with probability at least $\delta$, the number of goods of agent $t$ left when $t$ arrives is at least

$$\left(1 - \frac{1}{256(1-x)}\right)(nk_t - n_t) - \sqrt{2(nk_t - n_t)\log{}^1/_\delta}$$

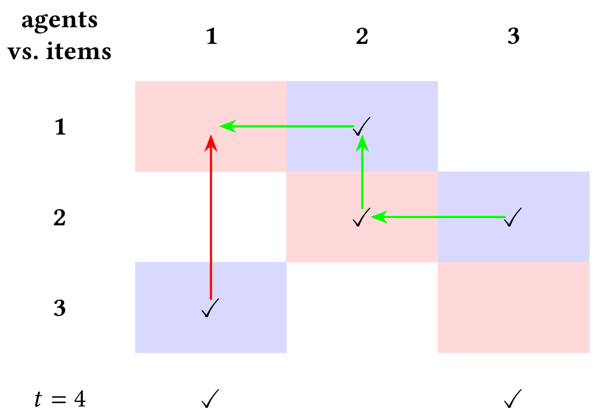

Fig. 4. Recharging step for constructing $G_t$ in the running example. Green arrows show how charges targeting items outside $S_t$ are redirected by following the charge chain until reaching an item in $S_t$. In particular, the charge of item 3 is redirected from item 2 $\notin S_t$ to item 1 $\in S_t$.

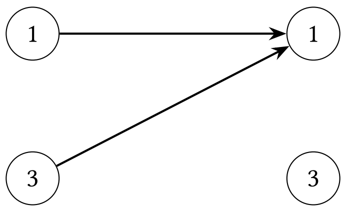

Fig. 5. Directed bipartite graph $G_t = (U_t, V_t, E_t)$ for the running example, restricted to items $\{1, 2, 3\} \subseteq S_t$. Edges represent the final (possibly redirected) charges from items allocated by Algorithm 3 in $S_t$ to items in $S_t$ allocated by Algorithm 4. After recharging, the left copy of every item in $S_t$ allocated before round $t$ by Algorithm 3 has exactly one outgoing edge, and each node on the right has in-degree at most 2.

If $k_t \geq 2$ then $nk_t - n_t \geq n$ and therefore the first term dominates the term under the square root for $\delta = \frac{1}{2^{2(n-n_t)}}$ and we get that with this probability, for $\epsilon$ arbitrarily close to 0, agent $t$ has at least $\left(1 - \frac{1}{256(1-x)}\right)(nk_t - n_t)(1-\epsilon)$ goods left.

On the other hand, when $k_t = 1$, to ensure that the second term is sufficiently smaller than the first, we must use $\delta = \frac{1}{2^{\frac{n-n_t}{10}}}$. Using this $\delta$, we get that agent $t$ has at least $\left(1 - \frac{1}{256(1-x)} - \frac{1}{\sqrt{5}}\right)(nk_t - n_t) > \frac{1}{2}(n - n_t)$.

We need to prove that this maintains that the agent $t$ is now *good* for Algorithm 3 and then we are done. Clearly, for the case when $k_t \leq 1$, we have less than half of agent $t$'s goods allocated. Therefore, using Lemma 4.6, we get that the number of goods allocated for agent $t$ in the run of Algorithm 3 is less than $n$. Therefore, $t$ has at least one good left and is good as per our definition.

We therefore only need to consider the case when $k_t \geq 2$. For this case, number of goods os $S_t$ allocated in Algorithm 4 is at most $nk_t - (1 - \frac{1}{256(1-x)}(nk_t - n_t)(1-\epsilon))$. Therefore, again using Lemma 4.6, the number of goods of $S_t$ allocated in Algorithm 3 is at most $2\left(nk_t - (1 - \frac{1}{256(1-x)}(nk_t - n_t)(1-\epsilon))\right)$.

Therefore, an agent is good in this case if we have

$$2\left(1-\frac{1}{256(1-x)}\right)(nk_t-n_t)(1-\epsilon)-nk_t \geq (1-x)(n-n_t)k_t$$

One can verify that for $x = {}^1\!/\!_4$, the above inequality stands true for an appropriately small $\epsilon$. □

### C.2.4 *Proof of Lemma 4.9.*

Proof. If $k_t \geq 2$, then all the proofs from the previous claim work exactly the same and we have $T(t) \geq S(t) - \frac{1}{2^{2(n-n_t)}}$.

We'll focus on the case when $k_t = 1$. Note that we still have that the probability of allocation of a good $j \in S_t \setminus H_t$ in one round is

$$\mathbb{P}\left[\text{good j allocated in one round}\right] \leq \frac{k_i/256}{(1-x)(n-n_t)k_i}.$$

Fix an single good $j_t$ from $S_t \setminus H_t$. Note that we have $t-1-n_t$ non-special agents before $t$, using union bound we know that

$$\mathbb{P}\left[\text{good } j_t \text{ is available for agent } t\right] \geq 1-\frac{t-1-n_t}{256(1-x)(n-n_t)} > 1-\frac{1}{256(1-x)}.$$

Note agent $t$ is good if $j_t$ is available for her. Thus using union bound we know $T(t) \geq T(t-1) - \frac{1}{256(1-x)}$. □