Minimizing Cumulative Envy in Allocating a Sequence of Items

Paul W. Goldberg, Isaac Robinson, and Nicholas Teh

University of Oxford, UK
{paul.goldberg,isaac.robinson,nicholas.teh}@cs.ox.ac.uk

Abstract. We study temporal fair division with indivisible goods that arrive sequentially and must be allocated irrevocably. In contrast to the usual online model, we assume that valuations and future arrivals are known in advance, and ask how unfairness evolves during the process. We introduce *cumulative maximum envy*: the sum, over all rounds, of the maximum pairwise envy at that round. Equivalently, this is the area under the worst-envy curve, and it captures both the magnitude and the duration of envy. For a fixed arrival order, we show that the corresponding decision problem is strongly NP-complete and that minimizing this objective admits no constant-factor approximation unless P = NP, even under identical valuations and even under binary valuations. We complement these hardness results with a dynamic program that gives pseudopolynomial-time solvability for a constant number of agents, polynomial-time algorithms in further restricted settings, and an FPTAS for fixed n under identical integer valuations. We then study a sequencing variant where the algorithm may choose the arrival order. This variant remains NP-complete even for two agents with identical valuations; however, a simple greedy algorithm achieves a 3/2-approximation for $n = 2$ agents, an $n/(n-1)$-approximation for any number of agents, and an additive guarantee depending on the maximum value of any good.

1 Introduction

Fair division of indivisible goods is a fundamental, well-studied problem at the intersection of computer science and economics [9, 30], with applications ranging from divorce settlements to distributed load balancing. A central fairness notion in this context is *envy-freeness*, which requires that every agent values their allocated bundle at least as much as any other agent's bundle. However, this strong requirement is not always achievable; consider, for example, the case of two agents and a single good that both value positively, which must be allocated to one of them. As a result, recent years have seen considerable interest in relaxations of envy-freeness. Arguably, the two most prominent relaxations are *envy-freeness up to one good (EF1)* and *envy-freeness up to any good (EFX)*. An allocation is EF1 if, for any pair of agents, any envy can be eliminated by removing one good from the envied agent's bundle, where goods are compared using the envious agent's valuation; in particular, removing a most-valued good is sufficient. It is EFX if the envy is eliminated even after removing a least-valued good.

The concept of EF1 originates from the seminal work of Lipton et al. [28], which introduced the first algorithmic approach for computing approximately envy-free allocations of indivisible goods. Their *envy-cycle elimination* algorithm remains a foundational tool and has since been used as a building block in tackling some of the most challenging problems in fair division, including the recent breakthrough on the existence of EFX allocations for three agents [10]. Interestingly, EF1 emerged from an effort to bound envy: if each agent's maximum envy can be bounded by the largest value of a single good, then a polynomial-time algorithm can achieve this relaxed form of envy-freeness. This same line of reasoning also motivates the optimization problem of minimizing envy, which was considered by Lipton et al. [28]. The optimization perspective has received comparatively less attention than discrete relaxations such as EF1 and EFX.

While most prior works study the fair division problem in the *offline* setting (i.e., all goods are immediately available and are allocated to agents in a single-shot), many real-world systems do not allocate all resources at once. Instead, indivisible goods become available over time and must be assigned as they appear: affordable housing units are completed in phases, public extraction permits are issued across multiple periods, and platforms assign arriving tasks or opportunities to workers. In such settings, stakeholders care not only about the *final* allocation, but also about the *trajectory*: large early imbalances can create lasting disadvantages, distort downstream behavior, or erode trust.

Several recent works have considered an *online* model where goods arrive one at a time, and must immediately and irrevocably be allocated to agents. This captures numerous application scenarios where goods need to be allocated in a sequential manner, such as assigning donations to food banks [1], allocating resources in cloud computing environments [6], or allocating impressions in recommendation systems [31]. In the standard online fair division model, it is usually assumed that each good's valuation is not known until its arrival. Most work on online fair division evaluates fairness only at the end of the process, once all goods have arrived and been allocated. In practice, agents experience the allocation process as it unfolds, and limiting unfairness throughout the process can be important for maintaining balance, participation, and trust, especially in long-term allocation systems.

This motivation is central to recently-studied *temporal fair division* model [11, 14], where the goal is to guarantee fairness not merely at the end, but at *every prefix* during the allocation process given information about future arrivals and valuations for each good-agent pair. This perspective provides a different lens on sequential fair division. In the standard and commonly-studied online model, agents have limited to no information about future arrivals, and fairness is evaluated only at the end (after all goods have arrived). In contrast to these standard online models with unknown future arrivals, the temporal fair division model assumes complete information about valuations and future arrivals while striving for typically stronger, *cumulative* fairness guarantees [11, 14].[1] Consider the allocation of pre-built affordable housing units that become available over time: while the housing authority knows when each unit will be ready, ensuring equitable distribution throughout the multi-year process—not just at the end—is essential for maintaining public trust and preventing systematic disadvantage to any group. Similarly, when allocating extraction rights for public natural resources across multiple time periods to private companies, temporal fairness ensures that no firm gains disproportionate early advantages that could distort market competition.

To quantify experienced unfairness over time, we take an optimization view rooted in classic envy minimization [28]. For an allocation process, let A_i^t denote agent i's bundle after the first t goods, and define $\mathrm{envy}_{i,j}^t := \max\{0, v_i(A_j^t) - v_i(A_i^t)\}$. Let $c_t := \max_{i,j \in N} \mathrm{envy}_{i,j}^t$ be the maximum envy after the first t goods have been allocated. The sequence $(c_1, \ldots, c_m)$ is the worst-envy curve of the process. Our objective is $\sum_{t=1}^m c_t$, the area under this curve. This objective is a simple, parameter-free baseline: it uses the standard maximum-envy measure at each round and aggregates it over time, so it penalizes both large envy and envy that persists for many rounds. We do not claim that this is the only meaningful temporal objective. Rather, it is a natural optimization counterpart to prefix-based temporal fairness notions, and it remains meaningful even when maintaining a strong fairness guarantee at every prefix is impossible.

For each agent i, let $e_i^t := \max_{j \in N} \mathrm{envy}_{i,j}^t$ and $E_i := \sum_{t=1}^m e_i^t$. Since $c_t = \max_{i \in N} e_i^t$, we have $\max_{i \in N} E_i \le \sum_{t=1}^m c_t \le \sum_{i \in N} E_i \le n \max_{i \in N} E_i$. Thus, our objective upper-bounds the cumulative envy experienced by every agent and is within a factor n of the largest individual cumulative envy. In either of our two models, an optimal solution for our objective is therefore an n-approximation for directly minimizing $\max_{i \in N} E_i$ over the same feasible allocation processes. However, our objective need not distribute envy evenly across agents. Directly minimizing $\max_{i \in N} E_i$, $\sum_{i \in N} E_i$, or weighted versions of these quantities is a natural direction for future work. It would also be useful to study objectives that directly minimize the largest or total individual cumulative envy.

We investigate cumulative maximum envy minimization in two related settings. First, we study the standard temporal model where the arrival order is fixed and may be adversarially chosen as part of the input. Second, we study a sequencing variant in which the algorithm may choose the order of the goods before allocating them. This models settings where all goods are known and can be held briefly before release; for example, a food bank may have several donated pallets in storage and choose which pallet to allocate next, or a public agency may choose the order in which approved permits are issued.

[1] Note that without this "cumulative fairness" consideration, the problem reduces to the standard offline model.

1.1 Our Contributions

We begin in Section 3 by presenting several intractability results. We show that the fixed-order problem is NP-complete even under identical valuations and even under binary valuations. In both cases, we also prove that no constant-factor approximation is possible unless $\mathsf{P} = \mathsf{NP}$.

Motivated by these hardness results, in Section 4 we give a dynamic programming algorithm that computes an optimal allocation sequence in time $\mathcal{O}((m + 1)(V_{\max}m^2 + 1)(mV_{\max} + 1)^{n^2} \cdot n^3)$, where n is the number of agents, m is the number of goods, and $V_{\max}$ is the maximum value of any good to any agent. This gives pseudopolynomial-time solvability when n is constant, and polynomial-time solvability under binary valuations for constant n. If the number of goods is constant, an optimal solution can also be found by direct enumeration. We further show that binary identical valuations admit a simple polynomial-time optimal algorithm, and that identical integer valuations admit an FPTAS for every fixed n.

In Section 5, we explore a natural variant of the model in which the algorithm is allowed to choose the arrival order of goods (while an adversary preselects the agents' valuations over the goods). We show that this variant remains NP-complete even for just two agents and identical valuations. On the positive side, when agents have identical valuations, we design a polynomial-time algorithm that achieves: (i) a 3/2-approximation for $n = 2$ agents, (ii) a $n/(n-1)$-approximation for general n, and (iii) a $\frac{n-1}{n} V_{\max}$-additive approximation to the optimal cost.

1.2 Related Work

We highlight the works most relevant to our focus on envy minimization in the temporal setting.

Offline Fair Division. In the offline fair division model, all goods are available simultaneously and allocated at once. Lipton et al. [28] investigate envy minimization in this setting, aiming to bound the maximum envy between any pair of agents. They introduce the well-known envy cycle elimination algorithm, which runs in polynomial time and guarantees an allocation where the *maximum envy* is at most the highest value of any good. This result also gives rise to the well-known EF1 concept.

Online Fair Division. In the online fair division model, goods arrive sequentially in a fixed order, and the algorithm has no knowledge of future items. Due to this uncertainty, most works on this model focus on ensuring fairness guarantees at the final round, once all goods have arrived and been allocated. Several works look at minimizing the maximum envy in this setting. Benade et al. [7] study a fully adversarial setting in which item values are chosen adaptively: an adversary observes the algorithm and all decisions made thus far, and selects the value of each incoming good to maximize envy at the end. They prove that any online algorithm must incur a maximum envy of at least $\Omega(\sqrt{m})$, where m is the number of goods, even with just two agents.

Recognizing that adaptive adversaries may be unrealistically powerful, Halpern et al. [22] consider an *oblivious adversary* model, where item values are fixed in advance and chosen to be worst-case with respect to the algorithm, but do not depend on the actual allocation decisions made in previous rounds. In this setting, they show connections to the *online multicolor discrepancy* problem and, in the special case of two agents, to *online vector balancing* [33].

In contrast, our work assumes full information about the values of all goods from the outset, and focuses on cumulative envy minimization, aligning more with the temporal fair division model than with adversarial online models.

Temporal Fair Division. In the temporal fair division model (sometimes also referred to as the *fully informed* online fair division model), the sequence of goods is known in advance, allowing the algorithm to make allocation decisions with full knowledge of future arrivals. This setting enables us to introduce the study of fairness guarantees that are cumulative over time, rather than only at the final round. Cookson et al. [11] introduce temporal fair division and give existence/algorithmic guarantees for achieving EF1-type fairness simultaneously at multiple points in time, with stronger positive results in special cases (e.g., two agents or structured preferences). Elkind et al. [14] formalize temporal EF1 (TEF1) for sequential item arrivals, showing that TEF1 need not exist in general and that deciding existence is NP-hard, while providing polynomial-time algorithms in several structured cases. He et al. [23] maintain EF1 at every round by allowing limited reallocations (or what they call adjustments), and quantify how many such changes are necessary (including separations between informed vs. uninformed settings). In contrast to this constraint-based line of work (maintaining temporal EF1 at every prefix), we take an optimization view and minimize the cumulative "area under the envy curve", which remains meaningful even when strong prefix-fairness constraints are infeasible. Micheel and Wilczynski [29] study a related repeated house allocation model in which every round is a complete matching, so each agent receives one object. They focus on envy-based fairness across the repeated assignments, together with Pareto efficiency in each round. Their model differs from ours in two main ways: they make a complete assignment in every round, whereas we allocate one indivisible good irrevocably at each step; and they study fairness conditions across repeated assignments, whereas we minimize cumulative maximum envy.

2 Preliminaries

For each positive integer z, we denote $[z] := \{1, \ldots, z\}$. An instance of the *fair division* problem is a tuple $\mathcal{I} = (N, G, \mathbf{v} = (v_1, \ldots, v_n))$, where $N = [n]$ is the set of (fixed) *agents*, G is a set of m goods, and for each $i \in N$, the function $v_i \colon 2^G \to \mathbb{R}_{\geq 0}$ is the *valuation function* of agent i. We write v instead of v_i when all agents have identical valuation functions. In the *temporal fair division* problem, goods arrive sequentially according to some order π, and we must allocate each good (to an agent) as it arrives. Let Π denote the space of all such arrival orders. In this work, we consider both the setting where π is chosen by an adversary (i.e., the standard temporal fair division setting) and the setting where π can be chosen by the algorithm.

As with most works in the fair division literature, we assume that valuation functions are *additive*, i.e., for any subset of goods $B \subseteq G$, $v_i(B) = \sum_{g \in B} v_i(\{g\})$. For convenience, we write $v_i(g)$ instead of $v_i(\{g\})$ for a single good g. Every subset of goods in G can also be referred to as a *bundle*. An *allocation* $\mathcal{A} = (A_1, \ldots, A_n)$ is a partition of the goods into bundles, with agent $i \in N$ receiving bundle A_i. For each $i \in N$ and $t \in [m]$, let A_i^t denote the partial allocation of agent i *after* the t-th arriving good is allocated to an agent and let $A_i^0 = \varnothing$ be the initial empty allocation, and $A_i^m = A_i$. Equivalently, we can represent an allocation by an allocation sequence $S = (s_1, \ldots, s_m) \in N^m$, where for each $t \in [m]$, the t-th entry s_t specifies the agent that the t-th arriving good is allocated to. Let the space of all such sequences be $\mathcal{S}$.

Next, we define *envy* and the *envy minimization* objective. Let S be the allocation sequence corresponding to $\mathcal{A}$, given some order π. Then, for any $t \in [m]$, the *envy* of agent $i \in N$ toward agent $j \in N \setminus \{i\}$ is $\text{envy}_{i,j}^t(S, \pi) := \max\{0, v_i(A_j^t) - v_i(A_i^t)\}$; in particular, $\text{envy}_{i,i}^t(S, \pi) = 0$. The *per-round cost* is $c_t(S, \pi) := \max_{i,j \in N} \text{envy}_{i,j}^t(S, \pi)$, and the *cumulative maximum envy* is $\mathsf{cost}(S, \pi) := \sum_{t=1}^m c_t(S, \pi)$, which we seek to minimize.

Maximum envy is a standard quantitative objective in both offline envy minimization [28] and online envy minimization [8, 22]. In most online models, fairness is evaluated only after the last item has been allocated. In the fully informed temporal model, however, future arrivals and valuations are known, so it is natural to ask for fairness over the trajectory of the allocation, not just at the end. One stronger requirement would be to optimize every prefix simultaneously, or to impose a fairness constraint at every prefix as in temporal EF1. We instead optimize the total area under the worst-envy curve. This permits short-term imbalances when they are compensated later, but penalizes imbalances that are large or long-lasting. This matches settings where temporary disparities are acceptable but prolonged severe unfairness is not.

Finally, optimizing only the final maximum envy can be badly misaligned with minimizing cumulative exposure to envy. Example 1 gives a linear separation even under identical valuations; conversely, cumulative maximum envy always controls final envy because $c_m(S, \pi)$ is one of the terms in $\sum_t c_t(S, \pi)$.

Example 1. Consider an instance with $n = 2$ and identical valuation function v. Fix any integer $q \geq 2$, let $m := 2q + 1$ and $\varepsilon := \frac{99}{2q}$. Then m is odd and $(m-1)\varepsilon = 99$. Let $\pi = (g_1, \ldots, g_m)$ be a fixed arrival order where $v(g_t) = \varepsilon$ for every $t \leq m - 1$ and $v(g_m) = 100$. For $n = 2$, we have $c_t(S, \pi) = |v(A_1^t) - v(A_2^t)|$ for every $t \in [m]$.

- *An allocation minimizing the final round cost.* Allocate $\{g_1, \ldots, g_{m-1}\}$ to agent 1 and g_m to agent 2. Then $v(A_1^m) = 99$ and $v(A_2^m) = 100$, so $c_m(S, \pi) = 1$ (and no allocation can have $c_m(S, \pi) < 1$ since $v(G) = 199$). Moreover, for

each $t \leq m-1$ we have $v(A_1^t) = t\varepsilon$ and $v(A_2^t) = 0$, so $c_t(S,\pi) = t\varepsilon$. Thus, $\mathrm{cost}(S,\pi) = \sum_{t=1}^{m-1} t\varepsilon + 1 = \varepsilon\frac{m(m-1)}{2} + 1 = \frac{99m}{2} + 1$, which grows linearly in m.
- *An allocation with much smaller cumulative cost.* Alternately allocate the first $m-1$ goods between the agents. Then for every $t \leq m-1$, we maintain $|v(A_1^t) - v(A_2^t)| \leq \varepsilon$, so $c_t(S,\pi) \leq \varepsilon$; more precisely, $c_t(S,\pi) = \varepsilon$ on odd t and $c_t(S,\pi) = 0$ on even t. Since $m-1$ is even, $\sum_{t=1}^{m-1} c_t(S,\pi) = \frac{m-1}{2}\varepsilon$. Allocating g_m gives $c_m(S,\pi) = 100$, so $\mathrm{cost}(S,\pi) = \frac{m-1}{2}\varepsilon + 100 = \frac{99}{2} + 100 = \frac{299}{2} < \frac{99m}{2} + 1$ for all $m > 3$.

3 Intractability Results

We first present several results highlighting the computational intractability of envy minimization as an objective in the temporal fair division setting.

We begin by formalizing the decision problem.

Temporal-EnvyMin

Given an instance $\mathcal{I} = (N, G, \mathbf{v})$, an order $\pi = (g_1, \ldots, g_m)$ over the goods in G, and an integer $\gamma \in \mathbb{Z}_{\geq 0}$, can we find an allocation sequence S such that $\mathsf{cost}(S,\pi) \leq \gamma$?

Then, we show that the problem is computationally intractable even in the restricted setting where agents have *identical valuation functions*—that is, for any pair of agents $i, j \in N$ and good g, we have $v_i(g) = v_j(g) = v(g)$. Moreover, we go one step further by showing that a constant-factor approximation algorithm is unlikely to exist.

Theorem 1. *Temporal-EnvyMin is strongly NP-complete even when agents have identical valuations. Moreover, in this setting, there exists no constant-factor approximation algorithm, unless P = NP.*

Proof. We reduce from the 3-Partition problem, which is known to be strongly NP-complete [17]. An instance of 3-Partition is given by a multiset of positive integers $X = \{x_1, \ldots, x_{3k}\}$ (with $\sum_{j=1}^{3k} x_j = kT$); it is a yes-instance if there exists a partition of X into k subsets $X'_1, \ldots, X'_k$ (each containing three integers) such that $\sum_{x \in X'_j} x = T$ for all $j \in [k]$, and a no-instance otherwise. Note that for each integer $x \in X$, we can without loss of generality assume that $x < T$ (otherwise it is trivially a no-instance).

Construct an instance of our problem, where $N = [k+2]$ and we have $m := 9k+7$ goods arriving in the order $\pi = (g_1, g_2, \ldots, g_{3k+1}, g_{3k+2}, g_{3k+3}, \ldots, g_{9k+7})$, with valuations defined as follows:

$$v(g_j) = \begin{cases} 10T & \text{if } j = 1; \\ x_{j-1} + 3T & \text{if } j \in \{2, \ldots, 3k+1\}; \\ 10T & \text{if } j = 3k+2; \\ 0 & \text{if } j \in \{3k+3, \ldots, 9k+7\}. \end{cases}$$

We first show that Temporal-EnvyMin is NP-complete (it is easy to see that the problem is in NP, we thus focus on proving NP-hardness). We do so by proving that there exists a solution $X'_1, \ldots, X'_k$ to 3-Partition if and only if there exists an allocation $\mathcal{A}$ (and hence corresponding allocation sequence S) such that $\mathsf{cost}(S, \pi) \leq 10T \cdot (3k+1)$ for the above constructed instance.

($\Rightarrow$) Suppose there exists a solution $X'_1, \ldots, X'_k$ to 3-Partition. Consider the sequence $S = (s_1, \ldots, s_m)$:

$$s_j = \begin{cases} 1 & \text{if } j \in \{1, 3k+3, \ldots, 9k+7\}; \\ 2 & \text{if } j = 3k+2; \\ \ell + 2 & \text{if } x_{j-1} \in X'_\ell \text{ for } \ell \in [k] \end{cases}$$

Observe that for all $t \in \{1, \ldots, 3k+1\}$, every agent $i \geq 3$ holds at most three positively-valued goods, whose total value is at most $T + 9T = 10T$. Thus, $\max_{i \in N} v(A_i^t) = 10T$ and $\min_{i \in N} v(A_i^t) = 0$, so we have that $c_t(S, \pi) = 10T$. Moreover, since each agent in $N \setminus \{1, 2\}$ has bundle value $T + 3 \cdot 3T = 10T$, we get that $c_{3k+2}(S, \pi) = 0$. Then,

$$\begin{aligned} \mathsf{cost}(S, \pi) &= \sum_{t=1}^{3k+2} c_t(S, \pi) + \sum_{t=3k+3}^{9k+7} c_t(S, \pi) \\ &= 10T \cdot (3k+1) + 0 + 0 = 10T \cdot (3k+1). \end{aligned}$$

($\Leftarrow$) Suppose there does not exist a solution to 3-Partition. This means that there does not exist a partition of integers in $X = \{x_1, \ldots, x_{3k}\}$ into k subsets of equal sum T. Equivalently, there is no way to partition the $3k$ goods $\{g_2, \ldots, g_{3k+1}\}$ into k bundles each of total value $10T$.

Let $i^* \in N$ be the agent that receives good g_{3k+2}, i.e., $g_{3k+2} \in A_{i^*}$. We split our analysis into two cases.

Case 1: No good in $\{g_1, \ldots, g_{3k+1}\}$ allocated to agent i^*. Let g_1 be allocated to agent $i' \in N \setminus \{i^*\}$. Then, $A_{i'}^1 = \{g_1\}$ and $v(A_{i'}^1) = 10T$. Now, since no goods in $\{g_1, \ldots, g_{3k+1}\}$ was allocated to agent i^*, for all $t \in \{1, \ldots, 3k+1\}$, we have that $A_{i^*}^t = \varnothing$, giving us $c_t(S, \pi) \geq 10T$. At round $3k+2$, agent i^* receives a good valued at $10T$ and thus has value exactly $10T$ (since she received nothing earlier). If $c_{3k+2}(S, \pi) = 0$, then every agent must have value exactly $10T$ at round $3k+2$ (since the total value of the first $3k+2$ goods is $10T(k+2)$, equality across $k+2$ agents forces each to have value $10T$). The agent i' holding the first good valued at $10T$ (i.e., g_1) cannot hold any other positively-valued good (each such good has value $> 3T$), else it would exceed $10T$. Therefore, the remaining k agents must partition the $3k$ goods of values $x_{j-1} + 3T$ into k bundles each of value $10T$. Since each such good has value strictly between $3T$ and $4T$, each bundle must contain exactly 3 goods, and the x_j parts in each bundle must sum to T, which is a 3-Partition solution, giving us a contradiction. Thus, $c_{3k+2}(S, \pi) \geq 1$, and together with $\sum_{t=1}^{3k+1} c_t(S, \pi) \geq 10T(3k+1)$, we get $\mathsf{cost}(S, \pi) > 10T(3k+1)$.

Case 2: Some good in $\{g_1, \ldots, g_{3k+1}\}$ allocated to agent i^*. Let g_1 be allocated to agent $i' \in N$ (note that it could be $i' = i^*$). Since i^* is also allocated good g_{3k+2}, we have that $v(A_{i^*}^{3k+2}) \geq 3T + 10T = 13T$. If $i' = i^*$, then $3k$ goods are distributed among $k+1$ agents, so some agent gets at most two goods from $\{g_2, \ldots, g_{3k+1}\}$. If $i' \neq i^*$, then i^* must have taken at least one of the $3k$ goods (since g_1 went to i'), leaving at most $3k-1$ such goods for the remaining k agents in $N \setminus \{i', i^*\}$; and by pigeonhole one of them gets at most two goods from $\{g_2, \ldots, g_{3k+1}\}$. Now, for each $j \in \{2, \ldots, 3k+1\}$, we know that $v(g_j) < T + 3T = 4T$ (by our assumption that $x < T$ for each $x \in X$). Then, $v(A_i^{3k+2}) < 8T$. This means that $c_{3k+2}(S, \pi) \geq v(A_{i^*}^{3k+2}) - v(A_i^{3k+2}) > 13T - 8T = 5T$. There are $6k+5$ zero-valued goods after round $3k+2$, i.e., rounds $t = 3k+3, \ldots, 9k+7$. Since these goods do not change any bundle values, $c_t(S, \pi) = c_{3k+2}(S, \pi)$ for all such t. Thus,

$$\sum_{t'=3k+3}^{9k+7} c_{t'} = (6k+5) \cdot c_{3k+2}(S, \pi) > (6k+5) \cdot 5T > 10T(3k+1),$$

which implies $\text{cost}(S, \pi) > 10T(3k+1)$.

Next, we argue that in (the same) setting where agents have identical valuations, there exists no constant-factor approximation algorithm, unless P = NP.

Since 3-Partition is strongly NP-complete, meaning even when $T = \mathsf{poly}(n)$, the problem remains NP-complete. Thus, we can assume $T = \mathsf{poly}(n)$.

Then, note that in the reduction above, we showed that given an instance $\mathcal{I}$ and order π, determining whether we can find an allocation sequence S such that $\mathsf{cost}(S, \pi) \leq 10T \cdot (3k+1)$ is NP-complete. Equivalently,

- any yes-instance admits a sequence S with $\mathsf{cost}(S, \pi) \leq 10T \cdot (3k+1)$, and in particular $c_{3k+2}(S, \pi) = 0$;
- any no-instance will force any sequence S to have $c_{3k+2}(S, \pi) \geq 1$.

Let $B := 10T(3k+1)$ and fix any constant $\alpha \geq 1$. To amplify the gap, append $Z_\alpha := \lceil B(\alpha - 1) \rceil + 1$ zero-valued goods after g_{9k+7}. In a yes-instance, the allocation above still has cost at most B, because the envy is already zero from round $3k+2$ onward. In a no-instance, every allocation has positive envy after round $3k+2$; since all appended goods have value zero, this envy persists through the appended rounds. Thus, every allocation has cost strictly larger than $B + Z_\alpha > \alpha B$. Therefore, given a 3-Partition instance, run the assumed α-approximation on the corresponding Temporal-EnvyMin instance and compare its returned cost with αB. If the original 3-Partition instance is a yes-instance, the returned cost is at most αB. If it is a no-instance, every feasible allocation has cost greater than αB. The assumed approximation algorithm would therefore decide 3-Partition in polynomial time, which is impossible unless $\mathsf{P} = \mathsf{NP}$.

Notably, the above hardness result continues to hold even when the maximum value of any good is bounded by a polynomial in n, owing to the strong NP-completeness property of the 3-Partition problem used in the reduction.

We now turn to a different restricted setting in which agents have *binary valuation functions*—that is, for each agent $i \in N$ and good g, $v_i(g) \in \{0, 1\}$. In this case, we establish a second intractability result by reducing from a known NP-hard problem in the standard (offline) fair division literature: deciding whether there exists a complete envy-free allocation under binary valuations, a problem we refer to as BINARY-EF. This hardness was first shown implicitly by Aziz et al. [3] and later made explicit by Hosseini et al. [24]. As in the previous case, we show that a constant-factor approximation algorithm is unlikely to exist.

Theorem 2. *TEMPORAL-ENVYMIN is NP-complete even when agents have binary valuations. Moreover, in this setting, there exists no constant-factor approximation algorithm, unless P = NP.*

4 Tractability in Restricted Settings

Given the computational hardness of finding an envy-minimizing allocation sequence, even under two distinct restricted settings (identical valuations or binary valuations), a natural follow-up question is whether there exists any restricted setting for which an optimal solution can be computed efficiently.

We begin with a simple warm-up result. The most favorable remaining setting is when valuations are both binary *and* identical. While this does not exactly reduce the problem to one with identically-valued goods,[2] it is closely related. In this case, a straightforward greedy algorithm (which allocates each arriving good to an agent with the currently lowest total bundle value) produces an allocation with minimum cost.

Proposition 1. *Under binary and identical valuations, an optimal sequence can be computed in polynomial time.*

Next, for the main result of this section, we present a dynamic programming algorithm that runs in time $\mathcal{O}((m+1)(V_{\max}m^2+1)(mV_{\max}+1)^{n^2} \cdot n^3)$, where $V_{\max} := \max_{i \in N, g \in G} v_i(g)$ is the maximum value of any good in any agent's view (and we can assume valuations are encoded in unary and scaled to integers). For $t \in \{0, \ldots, m\}$, let $M^t \in \{0, \ldots, m \cdot V_{\max}\}^{n \times n}$ be the *valuation matrix* after the first t goods have arrived: $M^t_{i,j}$ is the value agent i has for agent j's bundle thus far, and M^0 is the all-zero matrix.

The algorithm is presented as follows (Algorithm 1).

Intuitively, the algorithm explores every possible way to assign the first t goods while precisely tracking the valuation matrix M and cost (i.e., envy) incurred c. At each step, it branches on which agent receives the next good, updates the corresponding column of M, accrues the cost incurred, and records a predecessor pointer. At the end (layer m), it scans through all reachable pairs (c, M) and returns the one minimizing cost, reconstructing a witnessing allocation by following the pointers backward. Our result is as follows.

[2] This setting has also been studied in other fair division contexts; see, e.g., Elkind et al. [13].

Algorithm 1 Dynamic programming algorithm that returns an optimal sequence

Input: Instance $\mathcal{I} = (N, G, \mathbf{v})$ and order $\pi = (g_1, \ldots, g_m)$.
Output: Allocation sequence $S = (s_1, \ldots, s_m)$.

1: Let M^0 be the all–zero $n \times n$ matrix
2: Create hash maps $\mathrm{DP}[0], \ldots, \mathrm{DP}[m]$ mapping keys (c, M) to predecessor pointers
3: Initialize $\mathrm{DP}[0][(0, M^0)] \leftarrow \varnothing$
4: **for** $t = 0, \ldots, m-1$ **do**
5:   **for all** states (c, M) in $\mathrm{DP}[t]$ **do**
6:     **for** $s = 1, \ldots, n$ **do**
7:       $M' \leftarrow M$ (deep copy)
8:       **for** $i = 1, \ldots, n$ **do**
9:         $M'_{i,s} \leftarrow M'_{i,s} + v_i(g_{t+1})$
10:       **end for**
11:       $c' \leftarrow c + \max_{i,j} \max\{0, M'_{i,j} - M'_{i,i}\}$
12:       **if** $(c', M') \notin \mathrm{DP}[t+1]$ **then**
13:         $\mathrm{DP}[t+1][(c', M')] \leftarrow ((c, M), s)$
14:       **end if**
15:     **end for**
16:   **end for**
17: **end for**
18: Let $(c^*, M^*) := \arg\min_{(c,M) \in \mathrm{DP}[m]} c$
19: Initialize $S = (s_1, \ldots, s_m) = (0, \ldots, 0)$ and $(c, M) \leftarrow (c^*, M^*)$
20: **for** $t = m, \ldots, 1$ **do**
21:   $((c_{\text{prev}}, M_{\text{prev}}), s_t) \leftarrow \mathrm{DP}[t][(c, M)]$
22:   $S[t] \leftarrow s_t$
23:   $(c, M) \leftarrow (c_{\text{prev}}, M_{\text{prev}})$
24: **end for**
25: **return** $S = (s_1, \ldots, s_m)$

Theorem 3. *Algorithm 1 returns an optimal sequence in time* $\mathcal{O}((m+1)(V_{\max} m^2 + 1)(mV_{\max} + 1)^{n^2} \cdot n^3)$.

Proof. We first prove correctness. Observe that the $\{(c, M) : (c, M) \in \mathrm{DP}[m]\}$ is the set of pairs attainable after all goods have arrived. Then, selecting (c^*, M^*) with minimum c^* (Line 18) corresponds to one with minimum cost (or cumulative envy). From there, our algorithm is able to reconstruct a corresponding allocation sequence (Lines 20–24).

Thus, it suffices for us to prove the following invariant. Let $\pi_t := (g_1, \ldots, g_t)$.

Invariant. For every $t \in \{0, \ldots, m\}$ and pair (c, M), $\mathrm{DP}[t][(c, M)]$ exists if and only if there is an allocation sequence X_t for the first t goods (prefix π_t) such that the resulting valuation matrix is M and $c = \sum_{\tau=1}^{t} c_\tau(X_t, \pi_\tau)$. Moreover, the predecessor pointer encodes one such sequence.

We prove the invariant by induction on $t \in \{0, \ldots, m\}$.

For the base case, at $t = 0$, $\mathrm{DP}[t]$ only contains $(0, M^0)$, where M^0 is the all-zero matrix; and the cost is 0. Thus the invariant holds trivially.

Next, we prove the inductive step. Assume the invariant holds for some $t < m$. Since for each round t, $c_t(S, \pi) \leq t \cdot V_{\max}$, we have $\sum_{t=1}^{m} c_t(S, \pi) \leq V_{\max} \cdot m(m+1)/2 < V_{\max} \cdot m^2$.

Consider any state $(c, M) \in \mathrm{DP}[t]$. By the induction hypothesis, there exists an allocation sequence $X_t = (x_1, \ldots, x_t)$ of the first t goods with valuation matrix M and cost $\mathsf{cost}(X_t, \pi_t) = c$. When good g_{t+1} arrives, Algorithm 1 considers every agent $s \in N$ and for each s, constructs M' by increasing column s by $v_i(g_{t+1})$. Hence every feasible extension is represented in $\mathrm{DP}[t+1]$.

Conversely, suppose (c', M') is inserted into $\mathrm{DP}[t+1]$ from predecessor (c, M) using agent s. By the induction hypothesis, (c, M) arises from some allocation sequence $(x_1, \ldots, x_t)$ with matrix M and cost $\mathsf{cost}(X_t, \pi_t)$; extending by $x_{t+1} = s$ gives us matrix M' and updating the cost to c', so (c', M') corresponds to a feasible allocation of $t + 1$ goods. Note that after updating column s, M' is exactly the valuation matrix after $t + 1$ items under the extended sequence. Therefore, the maximum envy at round $t + 1$ is

$$c_{t+1} = \max_{i,j} \max\{0,\, M'_{i,j} - M'_{i,i}\},$$

so the cumulative cost is consistent with Line 11 of the algorithm. Thus the invariant holds for $t + 1$.

By induction, the invariant holds for all $t \leq m$, and optimality follows.

Next, we prove the running time. The state space is indexed by (t, c, M) with $t \in \{0, \ldots, m\}, c \in \{0, \ldots, V_{\max} \cdot m^2\}$, and $M \in \{0, \ldots, m \cdot V_{\max}\}^{n \times n}$. Thus, the number of states is

$$(m + 1) \cdot (V_{\max} \cdot m^2 + 1) \cdot (m \cdot V_{\max} + 1)^{n^2}.$$

From each state, there are n agents the algorithm can consider allocating the incoming good to, updates one column of M and computes the new cost. Both operations (i.e., per transition) take $\mathcal{O}(n^2)$ time. Since there are n transitions per state, it takes $\mathcal{O}(n^3)$ per state. Thus, the total running time is $\mathcal{O}((m + 1)(V_{\max} m^2 + 1)(m V_{\max} + 1)^{n^2} \cdot n^3)$. Reconstruction follows m pointers and is $\mathcal{O}(m)$, which does not change the bound.

This result has two important implications. First, for fixed n, the running time is polynomial in m and $V_{\max}$; thus the algorithm is pseudopolynomial. In particular, if item values are polynomially bounded integers (or given in unary), the algorithm runs in polynomial time.

Corollary 1. *TEMPORAL-ENVYMIN is XP with respect to (n, V_{max}). Equivalently, it is solvable in pseudopolynomial time when n is bounded by a constant.*

Second, in the case of binary valuations,[3] the size of the state space in the dynamic program becomes more structured. Then, the number of agents is constant, the problem is polynomial-time solvable (i.e. slicewise polynomial in n).

[3] Binary valuations are very well-studied in online fair division [1, 2, 35] and in the broader fair division literature [24, 21, 34]

Corollary 2. *Under binary valuations,* Temporal-EnvyMin *is XP with respect to n.*

These results show that, despite the general computational hardness of the problem, efficient algorithms are achievable under natural and practically motivated restrictions.

That being said, an intriguing open question remains: is the problem polynomial-time solvable when the number of agents is constant and agents have identical valuation functions? We conjecture that the answer is 'no'.

As a partial positive result, we provide an FPTAS for identical valuations (with integer item values) when n is fixed.

Theorem 4. *For any fixed $n \geq 2$, under identical valuations with integer-valued items, and for every $\varepsilon \in (0, 1)$, there exists a $(1+\varepsilon)$-approximation algorithm whose running time is polynomial in m and $1/\varepsilon$.*

5 Choosing the Arrival Sequence

The results presented in Section 3 show that computing a cost-minimizing allocation sequence is generally intractable, even under restricted settings such as identical or binary valuations. While Section 4 presents some positive algorithmic results under stronger assumptions, the overall hardness results align with existing impossibility results in the literature on online and temporal fair division: fairness objectives remain elusive in the presence of fixed input sequences. In fact, we illustrate with an example how an adversary deciding the arrival order can make a significant difference when it comes to minimizing cumulative envy.

Example 2. Consider an instance with $n = 2$ and identical valuation function v. Fix any integer $q \geq 2$, let $m := 2q + 1$, and set $\varepsilon := 99/(2q)$. Thus, m is odd and $(m - 1)\varepsilon = 99$. There is one good of value 100 and $m - 1$ goods of value ε.

- If the arrival order is $\pi = (g_1, \ldots, g_m)$ with $v(g_1) = 100$ and $v(g_t) = \varepsilon$ for all $t \geq 2$, then after round 1, $c_1(S, \pi) = 100$ for every allocation sequence S. Furthermore, each subsequent ε-good can reduce the gap $|v(A_1^t) - v(A_2^t)|$ by at most ε, so for $k = 0, 1, \ldots, m-1$, $c_{1+k}(S, \pi) \geq 100 - k\varepsilon$. Allocating every ε-good to the currently lagging agent achieves equality at every round, and therefore $\min_{S \in \mathcal{S}} \mathrm{cost}(S, \pi) = \sum_{k=0}^{m-1}(100 - k\varepsilon) = 100m - \varepsilon\frac{m(m-1)}{2} = \frac{101m}{2}$, which is $\Theta(m)$.
- In contrast, if the ε-goods arrive first (e.g., $\pi' = (g_2, \ldots, g_m, g_1)$), then allocating the first $m - 1$ goods alternately keeps $c_t(S, \pi') \leq \varepsilon$ for all $t \leq m - 1$, and the last step contributes $c_m(S, \pi') = 100$. Consequently, $\mathrm{cost}(S, \pi') = \frac{m-1}{2}\varepsilon + 100 = \frac{299}{2}$, which is $O(1)$ in m.

Thus, when one item dominates in value, placing it early can force cumulative cost that is a factor $\Theta(m)$ larger than an order that places it last.

Motivated by this example, we study a variant in which the algorithm chooses the order of the goods, while the agents' valuations are fixed in advance. This models settings where the full set of goods is known and goods can be held briefly before release. For example, a food bank may have several donated pallets in storage and choose which pallet to allocate next, or a public agency may choose the order in which approved permits are issued. We ask whether control over the order can reduce cumulative envy.[4]

Importantly, this setting is not inherently "stronger" or "weaker" than the standard one; it is simply of a different structure. Although the ability to control the arrival order offers additional flexibility, the benchmark for optimization also shifts: the optimal solution under this new setting may differ from that in the fixed-order setting. As a result, the optimization problem (e.g., obtaining approximation results) does not necessarily become easier—the performance is still measured relative to a stronger baseline, and the underlying computational challenges remain significant.

Observe that the cost objective is cumulative, so even when the final allocation can be envy-free, early rounds may still incur substantial envy. In the special case of identical valuations, Lemma 1 (stated below) implies the universal lower bound $\mathsf{OPT}(\mathcal{I}) \geq (1 - 1/n) \cdot v(G)$. One might hope that by choosing the arrival order, the algorithm could always achieve this bound. Our next result shows that even deciding whether this lower bound is attainable is NP-complete.

Temporal-EnvyMin-Seq

Given an instance $\mathcal{I} = (N, G, \mathbf{v})$ and an integer $\gamma \in \mathbb{Z}_{\geq 0}$, can we find a permutation π over the goods in G and an allocation sequence S such that $\mathsf{cost}(S, \pi) \leq \gamma$?

Theorem 5. *Temporal-EnvyMin-Seq is NP-complete even for two agents with identical valuations.*

Theorem 5 shows that allowing the algorithm to choose the arrival order does not eliminate computational intractability: the problem remains NP-hard even under identical valuations.[5] Nevertheless, this setting admits simple algorithms with provable approximation and additive guarantees.

We first state a universal lower bound (independent of the chosen order), which will serve as a benchmark for both hardness and approximation results. Importantly, the bound holds for any number of agents, assuming identical valuations. Given an instance $\mathcal{I}$, let $\mathsf{OPT}(\mathcal{I}) := \min_{S \in \mathcal{S}, \pi \in \Pi} \mathsf{cost}(S, \pi)$ denote the minimum possible cost achievable over all arrival orders π and allocation sequence S.

[4] Many related online problems assume a *random arrival order* as a way to circumvent impossibility results that arise under adversarial input models [4, 12, 15, 16, 18, 20, 26, 27].

[5] This setting has also been extensively studied both in sequential fair division models [25, 14] and in the broader fair division literature [5, 32].

Lemma 1. *For any instance $\mathcal{I} = (N, G, \mathbf{v})$ where $n \geq 2$ and agents have identical valuations, $\mathsf{OPT}(\mathcal{I}) \geq (1 - 1/n)\, v(G)$. Moreover, the bound is tight.*

Next, we introduce the following algorithm (Algorithm 2), designed for the setting where agents have identical valuations. The algorithm proceeds by first sorting the goods in increasing order of value (that will be the π), then iteratively allocating each good to the agent with the currently smallest bundle value. Despite its simplicity, we will show that this greedy approach is able to give us several positive approximation guarantees.

Algorithm 2 Identical-Lowest-First

Input: Instance $\mathcal{I} = (N, G, \mathbf{v})$.
Output: An order π and allocation sequence $S = (s_1, \ldots, s_m)$.

Initialize $\mathcal{A}^{(0)} \leftarrow (\varnothing, \ldots, \varnothing)$ and label goods such that $v(g_1) \leq \cdots \leq v(g_m)$
$\pi \leftarrow (g_1, \ldots, g_m)$
for $t = 1, \ldots, m$ **do**
  Let $i^* \in \arg\min_{i \in N} v(A_i^{t-1})$, with ties broken in favor of the lowest-index agent
  $s_t \leftarrow i^*$
  $A_{i^*}^t \leftarrow A_{i^*}^{t-1} \cup \{g_t\}$
  $A_i^t \leftarrow A_i^{t-1}$ for all $i \in N \setminus \{i^*\}$
end for
return π and allocation sequence $S = (s_1, \ldots, s_m)$

Scheduling interpretation. Under identical valuations, Algorithm 2 has the same steps as shortest-processing-time-first list scheduling on identical machines: goods are jobs, agents are machines, and $v(g)$ is the processing time of job g. The objectives are different, however. List scheduling is usually analyzed using the final makespan, that is, the largest final machine load. In our setting, c_t is the difference between the largest and smallest loads after the first t jobs, and we minimize $\sum_t c_t$. Thus, the $3/2$ factor in Theorem 6 does not follow from the usual list-scheduling analysis, although the common greedy rule may help explain why the same constant appears [19].

First, we show that in the case of two agents, we can get a $3/2$-approximation to the optimal cost.

Theorem 6. *When $n = 2$ and the agents have identical valuations, Algorithm 2 achieves a $3/2$-approximation to the optimal cost.*

We can extend the analysis to any n, obtaining an $\frac{n}{n-1}$-approximation (which approaches 1 as n grows).

Theorem 7. *For all $n \geq 2$, if the agents have identical valuations, then Algorithm 2 achieves a $\frac{n}{n-1}$-approximation to the optimal cost.*

For $n = 2$, the 3/2-approximation guarantee of Theorem 6 is tight; see Appendix C.5. For $n \geq 3$, Appendix C.6 provides an instance on which IDENTICAL-LOWEST-FIRST achieves approximation ratio $\frac{2n-1}{2n-2}$.

Finally, we complement the multiplicative bounds with an additive approximation guarantee, stated in terms of $V_{\max}$.

Theorem 8. *For all $n \geq 2$, if agents have identical valuations, then Algorithm 2 outputs an order π and allocation sequence S such that $\mathsf{cost}(S, \pi) \leq \mathsf{OPT}(\mathcal{I}) + \left(1 - \frac{1}{n}\right) V_{\max}$.*

In Appendix C.8, we present a matching lower bound.

6 Conclusion

In this paper, we initiated the study of envy minimization in temporal fair division from an optimization perspective, departing from the traditional focus on discrete relaxations like EF1 and EFX and on the prefix-based notions studied in earlier temporal work. In the standard model where the order of goods is adversarially chosen and fixed, we show that minimizing this objective is computationally intractable and cannot be approximated within any constant factor unless $\mathsf{P} = \mathsf{NP}$. In spite of this, we provide a dynamic programming algorithm that computes an optimal solution in several special cases. We then explored a natural variant in which the algorithm is allowed to choose the arrival order of goods, while an adversary pre-selects the valuations. Even with this additional flexibility, we show that the problem remains NP-complete for just two agents and identical valuations. We complement this with efficient approximation algorithms.

Our results contribute to the growing understanding of temporal fairness in sequential allocation problems. Several directions remain open. In particular, it would be interesting to study discounted cumulative envy where recent unfairness matters more than past inequities.

References

1. Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2540–2546, 2015.
2. Martin Aleksandrov and Toby Walsh. Pure nash equilibria in online fair division. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 42–48, 2017.
3. Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. Fair assignment of indivisible objects under ordinal preferences. *Artificial Intelligence*, 227:71–92, 2015.
4. Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *Proceedings of the 10th International Workshop on Approximation (APPROX) and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques (RANDOM)*, pages 16–28, 2007.

5. Siddharth Barman and Ranjani G. Sundaram. Uniform welfare guarantees under identical subadditive valuations. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 46–52, 2020.
6. Xiaohui Bei, Zihao Li, and Junjie Luo. Fair and efficient multi-resource allocation for cloud computing. In *Proceedings of the 18th Conference on Web and Internet Economics (WINE)*, pages 169–186, 2022.
7. Gerdus Benadè, Aleksandr M. Kazachkov, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. Fair and efficient online allocations. *Operations Research*, 72(4):1438–1452, 2023.
8. Gerdus Benadè, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. How to make envy vanish over time. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 593–610, 2018.
9. Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.
10. Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. Efx exists for three agents. *Journal of the ACM*, 71(1):1–27, 2024.
11. Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Temporal fair division. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13727–13734, 2025.
12. Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, pages 71–78, 2009.
13. Edith Elkind, Ayumi Igarashi, and Nicholas Teh. Fair division of chores with budget constraints. In *Proceedings of the 17th International Symposium on Algorithmic Game Theory (SAGT)*, pages 55–71, 2024.
14. Edith Elkind, Alexander Lam, Mohamad Latifian, Tzeh Yuan Neoh, and Nicholas Teh. Temporal fair division of indivisible items. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 676–685, 2025.
15. Jon Feldman, Nitish Korula, Vahab Mirrokni, S. Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In Stefano Leonardi, editor, *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE)*, pages 374–385, 2009.
16. Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 117–126, 2009.
17. Michael R. Garey and David S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975.
18. Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 982–991, 2008.
19. Ronald L. Graham. Bounds for certain multiprocessing anomalies. *The Bell System Technical Journal*, 45:1563–1581, 1966.
20. MohammadTaghi Hajiaghayi, MohammadReza Khani, Debmalya Panigrahi, and Max Springer. Online algorithms for the Santa Claus problem. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 30732–30743, 2022.
21. Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *Proceedings of the*

16th Conference on Web and Internet Economics (WINE), pages 370–383, 2020. Extended version available at arXiv:2007.06073v2.
22. Daniel Halpern, Alexandros Psomas, Paritosh Verma, and Daniel Xie. Online envy minimization and multicolor discrepancy: Equivalences and separations. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, page 188, 2025.
23. Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. Achieving a fairer future by changing the past. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 343–349, 2019.
24. Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, Hejun Wang, and Lirong Xia. Fair division through information withholding. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2014–2021, 2020.
25. Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9781–9789, 2024.
26. Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 587–596, 2011.
27. Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 630–631, 2005.
28. Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.
29. Karl Jochen Micheel and Anaëlle Wilczynski. Fairness in repeated house allocation. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3549–3556, 2024.
30. Hervé Moulin. *Fair Division and Collective Welfare.* MIT Press, 2003.
31. Riley Murray, Christian Kroer, Alex Peysakhovich, and Parikshit Shah. Robust market equilibria with uncertain preferences. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2192–2199, 2020.
32. Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.
33. Joel Spencer. Balancing games. *Journal of Combinatorial Theory, Series B*, 23(1):68–74, 1977.
34. Warut Suksompong and Nicholas Teh. On maximum weighted Nash welfare for binary valuations. *Mathematical Social Sciences*, 117:101–108, 2022.
35. Yuanyuan Wang and Tianze Wei. Online fair allocations with binary valuations and beyond. *arXiv preprint arXiv:2505.24321*, 2025.

Appendix

A Omitted Proofs from Section 3

A.1 Proof of Theorem 2

We reduce from the Binary-EF problem, which is known to be NP-hard [3, 24]. An instance of Binary-EF is given by a set of agents $\widehat{N} = \{1, \ldots, \widehat{n}\}$, a set of goods $\widehat{G} = \{\widehat{g}_1, \ldots, \widehat{g}_{\widehat{m}}\}$, and a valuation profile $\widehat{\mathbf{v}} = (\widehat{v}_1, \ldots, \widehat{v}_{\widehat{n}})$ where agents have additive binary valuations over the set of goods; it is a yes-instance if there exists an allocation $\widehat{\mathcal{A}}$ of goods to agents such that $\widehat{\mathcal{A}}$ is envy-free, i.e., $\widehat{v}_i(\widehat{A}_i) \geq \widehat{v}_i(\widehat{A}_j)$ for every pair of agents $i, j \in \widehat{N}$, and a no-instance otherwise.

Construct an instance of our problem, where $N = [\widehat{n}]$ and we have $\widehat{m} + \widehat{m}^3$ goods arriving in the order $\pi = (g_1, \ldots, g_{\widehat{m}+\widehat{m}^3})$, with valuations defined as follows:

$$v_i(g_j) = \begin{cases} \widehat{v}_i(\widehat{g}_j) & \text{if } j \in \{1, \ldots, \widehat{m}\}; \\ 0 & \text{otherwise.} \end{cases}$$

We first show that Temporal-EnvyMin is NP-complete (it is easy to see that the problem is in NP, we thus focus on proving NP-hardness). We do so by proving that there exists an envy-free allocation $\widehat{\mathcal{A}}$ for Binary-EF if and only if there exists an allocation $\mathcal{A}$ (and hence corresponding allocation sequence S) such that $\mathsf{cost}(S, \pi) \leq \widehat{m}^3 - 1$ for the above constructed instance.

The zero-valued goods make any envy present after the first $\widehat{m}$ rounds persist. If the Binary-EF instance is a yes-instance, we allocate the first $\widehat{m}$ goods according to an envy-free allocation, so all later rounds have zero envy. If it is a no-instance, every allocation has maximum envy at least 1 at round $\widehat{m}$, and this remains true while the zero-valued goods arrive. The same idea, with additional zero-valued goods, gives the constant-factor inapproximability result.

($\Rightarrow$) Suppose there exists an envy-free allocation $\widehat{\mathcal{A}}$ for Binary-EF. Then, consider the sequence $S = (s_1, \ldots, s_{\widehat{m}+\widehat{m}^3})$ whereby

$$s_j = \begin{cases} \ell & \text{if } j \leq \widehat{m} \text{ and } \widehat{g}_j \in \widehat{A}_\ell, \text{ for } \ell \in [\widehat{n}]; \\ 1 & \text{if } j > \widehat{m}. \end{cases}$$

Observe that for each $t \in [\widehat{m} - 1]$,[6] $c_t(S, \pi) \leq t$, and $c_{\widehat{m}}(S, \pi) = 0$, since $\widehat{\mathcal{A}}$ is envy-free, which means under $\mathcal{A}^{\widehat{m}}$, there is no envy (and cost is 0). Thus, we

[6] This is because under binary valuations, at round t, each bundle contains at most t goods and each good is worth at most 1 to any agent.

also get that for $t' \in \{\widehat{m}+1, \dots, \widehat{m}+\widehat{m}^3\}$, $c_{t'}(S,\pi) = 0$, giving us

$$\begin{aligned}\mathsf{cost}(S,\pi) &= \sum_{t=1}^{\widehat{m}+\widehat{m}^2} c_t(S,\pi) \\ &= \sum_{t=1}^{\widehat{m}-1} c_t(S,\pi) + c_{\widehat{m}}(S,\pi) + \sum_{t=\widehat{m}+1}^{\widehat{m}+\widehat{m}^3} c_t(S,\pi) \\ &\leq \sum_{t=1}^{\widehat{m}-1} c_t(S,\pi) + 0 + 0 = \frac{\widehat{m}(\widehat{m}-1)}{2}.\end{aligned}$$

Then, $\frac{\widehat{m}(\widehat{m}-1)}{2} \leq \widehat{m}^3 - 1$ for all $\widehat{m} \geq 1$.

($\Leftarrow$) Suppose there does not exist an envy-free allocation $\widehat{\mathcal{A}}$ for Binary-EF. This means that for any sequence S, the allocation induced on the first $\widehat{m}$ goods is not envy-free, and $c_{\widehat{m}}(S,\pi) \geq 1$. Since the next $\widehat{m}^3$ goods $g_{\widehat{m}+1}, \dots, g_{\widehat{m}+\widehat{m}^3}$ are zero-valued, we have that $c_{\widehat{m}+1}(S,\pi) = \cdots = c_{\widehat{m}+\widehat{m}^3}(S,\pi) \geq 1$. Thus, for any sequence S,

$$\mathsf{cost}(S,\pi) \geq \sum_{t=\widehat{m}}^{\widehat{m}+\widehat{m}^3} c_t(S,\pi) \geq \widehat{m}^3 + 1 > \widehat{m}^3 - 1.$$

Next, we argue that in (the same) setting where agents have binary valuations, there exists no constant-factor approximation algorithm, unless P = NP.

Note that in the reduction above, we showed that given an instance $\mathcal{I}$ and order π, determining whether we can find an allocation sequence S such that $\mathsf{cost}(S,\pi) \leq \widehat{m}^3 - 1$ is NP-complete. Equivalently,

- any yes-instance admits a sequence S with $\mathsf{cost}(S,\pi) \leq \widehat{m}^3 - 1$;
- any no-instance will force any sequence S to have $\mathsf{cost}(S,\pi) \geq \widehat{m}^3 + 1$.

Observe that for yes-instances, $c_{\widehat{m}}(S,\pi) = 0$, which means the cost of $\widehat{m}^3 - 1$ is incurred at earlier rounds, whereas for no-instances, $c_{\widehat{m}}(S,\pi) \geq 1$.

Let $B := m_b^3 - 1$ and fix any constant $\alpha \geq 1$. We may assume $m_b \geq 2$, so $B > 0$. Append

$$Z_\alpha := \lceil B(\alpha - 1) \rceil + 1$$

additional zero-valued goods. If the Binary-EF instance is a yes-instance, the same allocation still has cost at most B. If it is a no-instance, then envy is at least 1 from round m_b onward, and therefore it remains at least 1 throughout the appended zero-valued rounds. Thus every no-instance has cost at least

$$m_b^3 + 1 + Z_\alpha > \alpha B.$$

Thus, running an α-approximation on the constructed Temporal-EnvyMin instance and comparing its returned cost with αB would decide the original Binary-EF instance in polynomial time. This is impossible unless $\mathsf{P} = \mathsf{NP}$.

B Omitted Proofs from Section 4

B.1 Proof of Proposition 1

Consider the greedy algorithm that, at each round $t \in [m]$, allocates the arriving good to an agent with minimum current bundle value $v(A_i^{t-1})$. This can be implemented in $\mathcal{O}(mn)$ time.

Since valuations are binary and identical, each bundle value $v(A_i^t)$ is an integer that represents the number of goods of value 1 assigned to agent i among the first t arrivals. Let q_t be the number of goods of value 1 among the first t goods in π. Under identical valuations,

$$c_t(S,\pi) = \max_{i,j\in N}\left(v(A_j^t) - v(A_i^t)\right) = \max_{i\in N} v(A_i^t) - \min_{i\in N} v(A_i^t).$$

Then, since $\sum_{i\in N} v(A_i^t) = q_t$ and all $v(A_i^t)$ are integers,

$$\max_{i\in N} v(A_i^t) \geq \lceil q_t/n \rceil \text{ and } \min_{i\in N} v(A_i^t) \leq \lfloor q_t/n \rfloor.$$

Thus, for every allocation,

$$c_t(S,\pi) \geq \lceil q_t/n \rceil - \lfloor q_t/n \rfloor,$$

which equals 0 if and only if $q_t \equiv 0 \pmod{n}$, and equals 1 otherwise.

We claim the greedy algorithm maintains

$$\max_{i\in N} v(A_i^t) - \min_{i\in N} v(A_i^t) \leq 1 \text{ for all } t,$$

proved by induction on t: initially it is 0. If the arriving good has value 0, bundle values do not change. If it has value 1, the algorithm adds 1 to a minimum bundle value, so the gap cannot increase beyond 1. Thus, $c_t(S,\pi) \leq 1$ always. Moreover, if $q_t \not\equiv 0 \pmod{n}$, the bundle values cannot be equal integers summing to q_t, so $c_t(S,\pi) \neq 0$ and thus $c_t(S,\pi) = 1$. If $q_t \equiv 0 \pmod{n}$, the gap ≤ 1 forces equality, giving us $c_t(S,\pi) = 0$. To see why, if $\max_{i\in N} v(A_i^t) - \min_{i\in N} v(A_i^t) \leq 1$, then all $v(A_i^t) \in \{\min_{i\in N} v(A_i^t), \min_{i\in N} v(A_i^t) + 1\}$. Writing r for the number of agents with value $\min_{i\in N} v(A_i^t)+1$, we have $q_t = n\min_{i\in N} v(A_i^t)+r$. If $q_t \equiv 0 \pmod{n}$, then $r \equiv 0 \pmod{n}$, but $0 \leq r \leq n-1$, thus $r = 0$ and all values equal.

Thus, the greedy algorithm achieves a per-round lower bound for every t, and summing over t shows that it minimizes $\sum_{t=1}^{m} c_t(S,\pi)$.

B.2 Proof of Theorem 4

Fix a constant $n \geq 2$, an instance $\mathcal{I} = (N, G, v)$ with identical valuation function v, and a fixed arrival order $\pi = (g_1, \ldots, g_m)$. Let $\mathsf{OPT} := \min_{S\in\mathcal{S}} \mathsf{cost}_v(S,\pi)$ denote the optimal cumulative maximum envy for this fixed order under v.

If $v(G) = 0$, then every bundle has value 0 at every round, so $c_t(S,\pi) = 0$ for all t and all S. Hence OPT $= 0$, and any allocation sequence is optimal. We therefore assume $v(G) > 0$. Set

$$\delta := \frac{\varepsilon(1-1/n)\cdot v(G)}{m(m+1)},$$

and define a scaled (integer) valuation $\tilde{v}$ by

$$\tilde{v}(g) := \lfloor v(g)/\delta \rfloor \quad \text{for each } g \in G.$$

Let $\tilde{\mathcal{I}} := (N, G, \tilde{v})$ be the scaled instance. Compute an allocation sequence $\tilde{S}$ that optimizes $\mathsf{cost}_{\tilde{v}}(S, \pi)$ for the instance $\tilde{\mathcal{I}}$ and the same order π (e.g., using Algorithm 1 together with Theorem 3), and output $\tilde{S}$ for the original instance.

For every $g \in G$,

$$\tilde{v}(g) \leq \frac{v(g)}{\delta} \leq \frac{v(G)}{\delta} = \frac{m(m+1)}{\varepsilon(1-1/n)},$$

so $\tilde{V}_{\max} := \max_{g \in G} \tilde{v}(g) = \mathcal{O}(m^2/\varepsilon)$ (for fixed n). By Theorem 3, for fixed n, Algorithm 1 runs in time polynomial in m and $\tilde{V}_{\max}$, hence polynomial in m and $1/\varepsilon$.

Now, we prove the approximation guarantee. For any allocation sequence S and any round t, write

$$x_i^t := v(A_i^t) \quad \text{and} \quad \tilde{x}_i^t := \tilde{v}(A_i^t) \quad \text{for each } i \in N.$$

Since $\delta\tilde{v}(g) \leq v(g) < \delta(\tilde{v}(g)+1)$ for each good g and $|A_i^t| \leq t$ (because at most t goods have arrived by round t), we have that for every $i \in N$ and $t \in [m]$,

$$\delta\, \tilde{x}_i^t \ \leq\ x_i^t \ <\ \delta\, \tilde{x}_i^t + t\delta. \tag{1}$$

Define $U_t := \max_{i \in N} x_i^t$ and $L_t := \min_{i \in N} x_i^t$, and similarly $\tilde{U}_t := \max_{i \in N} \tilde{x}_i^t$ and $\tilde{L}_t := \min_{i \in N} \tilde{x}_i^t$. Under identical valuations we have

$$c_t(S, \pi) = U_t - L_t \quad \text{and} \quad \tilde{c}_t(S, \pi) = \tilde{U}_t - \tilde{L}_t.$$

From (1), we get $U_t \leq \delta\tilde{U}_t + t\delta$ and $L_t \geq \delta\tilde{L}_t$, thus

$$c_t(S, \pi) = U_t - L_t \leq \delta(\tilde{U}_t - \tilde{L}_t) + t\delta = \delta\tilde{c}_t(S, \pi) + t\delta.$$

Also $U_t \geq \delta\tilde{U}_t$ and $L_t \leq \delta\tilde{L}_t + t\delta$, hence

$$c_t(S, \pi) = U_t - L_t \geq \delta(\tilde{U}_t - \tilde{L}_t) - t\delta = \delta\tilde{c}_t(S, \pi) - t\delta.$$

Therefore, for every S,

$$\begin{aligned} \delta\mathsf{cost}_{\tilde{v}}(S, \pi) - \delta\sum_{t=1}^{m} t \leq \mathsf{cost}_v(S, \pi) \\ \leq \delta\mathsf{cost}_{\tilde{v}}(S, \pi) + \delta\sum_{t=1}^{m} t. \end{aligned} \tag{2}$$

Since $\sum_{t=1}^{m} t = m(m+1)/2$, (2) becomes

$$\begin{aligned} &\delta\mathsf{cost}_{\tilde{v}}(S, \pi) - \delta\frac{m(m+1)}{2} \\ &\leq \mathsf{cost}_v(S, \pi) \\ &\leq \delta\mathsf{cost}_{\tilde{v}}(S, \pi) + \delta\frac{m(m+1)}{2}. \end{aligned} \tag{3}$$

Let S^* be optimal for the original instance (so $\mathsf{cost}_v(S^*, \pi) = \mathsf{OPT}$). By optimality of $\tilde{S}$ for the scaled instance, $\mathsf{cost}_{\tilde{v}}(\tilde{S}, \pi) \le \mathsf{cost}_{\tilde{v}}(S^*, \pi)$. Applying (3) first to $\tilde{S}$ and then to S^* gives us

$$\begin{aligned}\mathsf{cost}_v(\tilde{S}, \pi) &\le \delta \mathsf{cost}_{\tilde{v}}(\tilde{S}, \pi) + \delta \frac{m(m+1)}{2} \\ &\le \delta \mathsf{cost}_{\tilde{v}}(S^*, \pi) + \delta \frac{m(m+1)}{2} \\ &\le \mathsf{cost}_v(S^*, \pi) + \delta m(m+1) \\ &= \mathsf{OPT} + \delta m(m+1).\end{aligned}$$

It remains to lower bound OPT in terms of $v(G)$. Fix any sequence S and define x_i^t, U_t, L_t as above. If g_t is allocated to agent j, then $x_j^t - x_j^{t-1} = v(g_t)$. Since $U_t \ge x_j^t$ and $L_{t-1} \le x_j^{t-1}$,

$$v(g_t) \;=\; x_j^t - x_j^{t-1} \;\le\; U_t - L_{t-1}.$$

Using $U_t = (U_t - L_t) + L_t = c_t(S, \pi) + L_t$, we get

$$v(g_t) \le c_t(S, \pi) + (L_t - L_{t-1}).$$

Summing over $t = 1, \ldots, m$ and using $L_0 = 0$ gives us

$$v(G) = \sum_{t=1}^{m} v(g_t) \le \sum_{t=1}^{m} c_t(S, \pi) + L_m = \mathsf{cost}_v(S, \pi) + L_m.$$

Since $L_m = \min_{i \in N} v(A_i^m) \le \frac{1}{n} \sum_{i \in N} v(A_i^m) = v(G)/n$, it follows that $\mathsf{cost}_v(S, \pi) \ge (1 - \frac{1}{n}) v(G)$. Minimizing over S yields

$$\mathsf{OPT} \ge (1 - \tfrac{1}{n})\, v(G). \tag{4}$$

Finally, by our choice of δ,

$$\delta m(m+1) = \varepsilon(1 - \tfrac{1}{n}) v(G) \le \varepsilon \mathsf{OPT},$$

where the inequality follows from (4). Consequently,

$$\mathsf{cost}_v(\tilde{S}, \pi) \le \mathsf{OPT} + \varepsilon \mathsf{OPT} = (1 + \varepsilon)\mathsf{OPT},$$

which proves the claimed $(1+\varepsilon)$-approximation and completes the proof.

C Omitted Proofs from Section 5

C.1 Proof of Theorem 5

Membership in NP is easy to see: given $\pi = (g_1, \ldots, g_m)$ and $S = (s_1, \ldots, s_m)$, we can compute all partial bundles $(A_i^t)_{i \in N, t \in [m]}$ and the values $c_t(S, \pi)$ for each

$t \in [m]$, and verify whether $\sum_{t=1}^{m} c_t(S,\pi) \leq \gamma$ in polynomial time. Thus, we focus on showing NP-hardness.

We reduce from the NP-hard PARTITION problem. An instance of PARTITION consists of positive integers $x_1, \ldots, x_q$ with total sum $T := \sum_{i=1}^{q} x_i$; we may assume T is even (otherwise it is a trivial no-instance). We construct an instance $\mathcal{I} = (N, G, \mathbf{v})$ of TEMPORAL-ENVYMIN-SEQ with $N = \{1,2\}$ and identical additive valuations (so $v_1 = v_2 = v$). Let $M := T + 1$. Create $2q$ goods (thus the number of goods is $m = |G| = 2q$), consisting of pairs (a_i, b_i) for each $i \in [q]$, where

$$v(a_i) = M \quad \text{and} \quad v(b_i) = M + x_i.$$

Then the total value is

$$v(G) = \sum_{i=1}^{q} (v(a_i) + v(b_i)) = \sum_{i=1}^{q} (2M + x_i) = 2qM + T,$$

and let

$$\gamma := \frac{v(G)}{2} = qM + \frac{T}{2},$$

which is an integer since T is even.

We will use the following fact, which holds for every order π and sequence S in this constructed instance. Since $n = 2$ and valuations are identical, for every $t \in [m]$ we have

$$c_t(S,\pi) = |v(A_1^t) - v(A_2^t)|.$$

Define $D_t := v(A_1^t) - v(A_2^t)$, so $c_t(S,\pi) = |D_t|$. For each $t \geq 2$, the t-th arriving good g_t is allocated to exactly one agent, so $D_t = D_{t-1} \pm v(g_t)$ and therefore $|D_t - D_{t-1}| = v(g_t)$. By the triangle inequality, we have that

$$v(g_t) = |D_t - D_{t-1}| \leq |D_t| + |D_{t-1}| = c_t(S,\pi) + c_{t-1}(S,\pi) \tag{5}$$

for all $t \geq 2$. Summing (5) over $t = 2, \ldots, m$ we get

$$\sum_{t=2}^{m} (c_{t-1}(S,\pi) + c_t(S,\pi)) \geq \sum_{t=2}^{m} v(g_t) = v(G) - v(g_1).$$

The left-hand side equals $c_1(S,\pi) + c_m(S,\pi) + 2\sum_{t=2}^{m-1} c_t(S,\pi) = 2\sum_{t=1}^{m} c_t(S,\pi) - c_1(S,\pi) - c_m(S,\pi)$, so

$$2\mathsf{cost}(S,\pi) - c_1(S,\pi) - c_m(S,\pi) \geq v(G) - v(g_1).$$

Moreover, after the first good arrives exactly one agent has received g_1 and the other has an empty bundle, thus $c_1(S,\pi) = v(g_1)$. Substituting gives us

$$2\mathsf{cost}(S,\pi) - c_m(S,\pi) \geq v(G).$$

Since $c_m(S,\pi) \geq 0$, we conclude that for every (S,π),

$$\mathsf{cost}(S,\pi) \geq \frac{v(G)}{2} = \gamma. \tag{6}$$

We now prove correctness of the reduction.

($\Rightarrow$) Suppose the PARTITION instance is a yes-instance, so there exists $P \subseteq [q]$ with $\sum_{i \in P} x_i = T/2$. We define a permutation π of the $2q$ goods by listing the goods in q consecutive pairs as follows:

- for each $i \in P$, place the pair (a_i, b_i) in π,
- for each $i \notin P$, place the pair (b_i, a_i) in π,

with all pairs corresponding to indices $i \in P$ appearing first (in arbitrary order among themselves), followed by all pairs corresponding to indices $i \notin P$ (in arbitrary order among themselves).

Next define an allocation sequence S by the greedy rule: at each round t, allocate g_t to an agent in $\arg\min_{i \in [2]} v(A_i^{t-1})$ (break ties arbitrarily). We analyze the contribution of each consecutive pair to the total cost. Fix a pair and let d be the absolute difference $|v(A_1^{t-1}) - v(A_2^{t-1})|$ immediately before the first good of the pair is allocated.

We first consider pairs (a_i, b_i) with $i \in P$. During this part of the sequence, the difference d is the sum of the x-values of some already-processed indices in P, hence $d \leq \sum_{i \in P} x_i = T/2 < M$. Assume w.l.o.g. that $v(A_1^{t-1}) \leq v(A_2^{t-1})$, so $v(A_2^{t-1}) - v(A_1^{t-1}) = d$. The greedy rule gives a_i to agent 1, so the new difference is $M - d$ and the incurred cost at that round is $M - d$. Now agent 2 has the smaller bundle, so b_i is given to agent 2; the new difference becomes

$$(M + x_i) - (M - d) = d + x_i,$$

and the cost at the next round is $d + x_i$. Therefore, this pair contributes $(M - d) + (d + x_i) = M + x_i$ to $\mathsf{cost}(S, \pi)$, and the absolute difference after the pair increases by x_i.

After all $i \in P$ are processed, the absolute difference equals $\sum_{i \in P} x_i = T/2$.

Next, we consider pairs (b_i, a_i) with $i \notin P$. At the start of this part, the difference is $T/2$, and we maintain the invariant that after processing some subset $Q \subseteq ([q] \setminus P)$, the current absolute difference is $\sum_{i \in ([q] \setminus P) \setminus Q} x_i$. In particular, before processing any remaining $i \notin P$, we have $d \geq x_i$. Again assume w.l.o.g. that agent 1 has the smaller bundle before the pair, so agent 2 is ahead by d. The greedy rule assigns b_i to agent 1. Since $M + x_i > d$, the new difference is $(M + x_i) - d$ and the cost at that round is $M + x_i - d$. Then the greedy rule assigns a_i to agent 2, and the new difference becomes

$$\big|(M + x_i - d) - M\big| = |x_i - d| = d - x_i,$$

using $d \geq x_i$. The pair therefore contributes

$$(M + x_i - d) + (d - x_i) = M$$

to $\mathsf{cost}(S, \pi)$ and decreases the absolute difference by x_i, preserving the invariant. After all $i \notin P$ are processed, the absolute difference returns to 0.

Summing contributions over all q pairs,

$$\begin{aligned}\mathsf{cost}(S,\pi) &= \sum_{i\in P}(M+x_i) + \sum_{i\notin P} M \\ &= qM + \sum_{i\in P} x_i = qM + \frac{T}{2} = \gamma.\end{aligned}$$

Thus, there exists (S,π) with $\mathsf{cost}(S,\pi) \leq \gamma$.

($\Leftarrow$) Conversely, suppose there exist π and S with $\mathsf{cost}(S,\pi) \leq \gamma$. By (6), we must have $\mathsf{cost}(S,\pi) = \gamma = v(G)/2$. From $2\mathsf{cost}(S,\pi) - c_m(S,\pi) \geq v(G)$ and $\mathsf{cost}(S,\pi) = v(G)/2$, we get $c_m(S,\pi) = 0$. Also, define for each $t \geq 2$ the slack $\delta_t := \big(c_{t-1}(S,\pi) + c_t(S,\pi)\big) - v(g_t) \geq 0$ from (5). Since equality holds in the summed inequality, we must have $\sum_{t=2}^{m} \delta_t = 0$, and therefore $\delta_t = 0$ for every $t \geq 2$, i.e., for all $t = 2, \dots, m$,

$$c_{t-1}(S,\pi) + c_t(S,\pi) = v(g_t). \tag{7}$$

Rearranging gives us $c_t(S,\pi) = v(g_t) - c_{t-1}(S,\pi)$ for all $t \geq 2$, and unrolling this recurrence, we get

$$\begin{aligned}&c_m(S,\pi) \\ &= v(g_m) - v(g_{m-1}) + v(g_{m-2}) - \cdots + (-1)^{m-1} v(g_1).\end{aligned}$$

Since $m = 2q$ is even and $c_m(S,\pi) = 0$, we get

$$\sum_{t\in[m],t \text{ is even}} v(g_t) = \sum_{t\in[m],t \text{ is odd}} v(g_t).$$

Every good in G has value either M or $M + x_i$. Write $v(g_t) = M + y_t$, where $y_t \in \{0, x_1, \dots, x_q\}$ and each x_i appears exactly once among the y_t's (coming from the unique good b_i of value $M+x_i$). Since there are exactly q even positions and q odd positions, the M-terms cancel, giving us

$$\sum_{t\in[m],t \text{ is even}} y_t = \sum_{t\in[m],t \text{ is odd}} y_t.$$

Because $\sum_{t=1}^{m} y_t = \sum_{i=1}^{q} x_i = T$, each side must equal $T/2$. Let $P \subseteq [q]$ be the set of indices i such that the good b_i (the unique good of value $M + x_i$) appears in an even position of π. Then

$$\sum_{i\in P} x_i = \sum_{t\in[m],t \text{ is even}} y_t = \frac{T}{2},$$

so the original PARTITION instance is a yes-instance.

This completes the reduction, and proves our result.

C.2 Proof of Lemma 1

Fix any arrival order $\pi = (g_1, \ldots, g_m)$ and allocation sequence S. For each $t \in \{0, 1, \ldots, m\}$ define

$$x_i^t := v(A_i^t), \quad L_t := \min_{i \in N} x_i^t, \quad U_t := \max_{i \in N} x_i^t.$$

Under identical valuations,

$$c_t(S, \pi) = \max_{i,j \in N} \max\{0, x_j^t - x_i^t\} = U_t - L_t.$$

Now fix any $t \in \{1, \ldots, m\}$ and let j be the agent who receives g_t. By additivity, $x_j^t - x_j^{t-1} = v(g_t)$. Since $U_t \geq x_j^t$ and $L_{t-1} \leq x_j^{t-1}$, we have

$$v(g_t) = x_j^t - x_j^{t-1} \;\leq\; U_t - L_{t-1}.$$

Using $U_t = (U_t - L_t) + L_t = c_t(S, \pi) + L_t$, this becomes

$$v(g_t) \;\leq\; c_t(S, \pi) + (L_t - L_{t-1}).$$

Summing over $t = 1, \ldots, m$ and using $L_0 = \min_i v(A_i^0) = v(\varnothing) = 0$, we get

$$\begin{aligned} v(G) = \sum_{t=1}^{m} v(g_t) &\leq \sum_{t=1}^{m} c_t(S, \pi) + \sum_{t=1}^{m} (L_t - L_{t-1}) \\ &= \mathsf{cost}(S, \pi) + L_m. \end{aligned}$$

Rearranging gives us

$$\mathsf{cost}(S, \pi) \geq v(G) - L_m = v(G) - \min_i v(A_i^m).$$

Finally,

$$\min_i v(A_i^m) \leq \frac{1}{n} \sum_{i \in N} v(A_i^m) = \frac{v(G)}{n},$$

so $\mathsf{cost}(S, \pi) \geq (1 - \frac{1}{n}) v(G)$. Taking a minimum over (S, π) gives us the same bound for $\mathsf{OPT}(\mathcal{I})$.

Next, we prove the factor $1 - 1/n$ is tight. For any $k \geq 1$, take $m = kn$ goods each with value 1, so $v(G) = kn$. Consider the round-robin allocation (one good per agent each cycle). At every round t that is not a multiple of n, the bundle values differ by exactly 1, so $c_t = 1$; at rounds t that are multiples of n, all bundle values are equal and $c_t = 0$. Thus $\mathsf{cost}(S, \pi) = kn - k = k(n-1) = (1 - \frac{1}{n}) \cdot v(G)$, matching the lower bound.

C.3 Proof of Theorem 6

Consider any instance $\mathcal{I} = (N, G, \mathbf{v})$ where $N = \{1, 2\}$ and agents have identical valuations. Let $S = (s_1, \ldots, s_m)$ be the allocation sequence induced from the order π and allocation $\mathcal{A}$ returned by the algorithm.

Also consider any $t \in \{2, \ldots, m\}$. Suppose, without loss of generality, that $v(A_1^{t-1}) \leq v(A_2^{t-1})$. Then,

$$c_{t-1}(S, \pi) = v(A_2^{t-1}) - v(A_1^{t-1}).$$

We first prove the following claim.

Claim. Under Algorithm 2 with $n = 2$ and goods ordered so that $v(g_1) \leq \cdots \leq v(g_m)$, we have $c_{t-1}(S, \pi) \leq v(g_t)$ for every $t \geq 2$. Thus, after allocating g_t to the agent with the smaller current bundle value, that agent becomes (weakly) the one with a larger bundle value, and therefore $c_t(S, \pi) = v(g_t) - c_{t-1}(S, \pi)$ (equivalently, $c_t(S, \pi) + c_{t-1}(S, \pi) = v(g_t)$).

Proof (Proof of Claim). We prove this claim by induction. For the base case: $c_1(S, \pi) = v(g_1) \leq v(g_2)$, so $c_1(S, \pi) \leq v(g_2)$. For the inductive step: assume $c_{t-1}(S, \pi) \leq v(g_t)$. Then, after allocating g_t to the agent with the smaller bundle, $c_t(S, \pi) = v(g_t) - c_{t-1}(S, \pi)$. In particular, $c_t(S, \pi) \leq v(g_t) \leq v(g_{t+1})$. Thus, $c_t(S, \pi) \leq v(g_{t+1})$, i.e., $c_t(S, \pi) \leq v(g_{t+1})$.

Thus, Algorithm 2 will allocate g_t to agent 1, giving us

$$A_1^t = A_1^{t-1} \cup \{g_t\} \quad \text{and} \quad A_2^t = A_2^{t-1}.$$

Then, we get that

$$\begin{aligned} c_t(S, \pi) = v(A_1^t) - v(A_2^t) &= v(A_1^{t-1}) + v(g_t) - v(A_2^{t-1}) \\ &= -c_{t-1}(S, \pi) + v(g_t). \end{aligned}$$

Thus, for all $t \in \{2, \ldots, m\}$,

$$c_t(S, \pi) + c_{t-1}(S, \pi) = v(g_t).$$

Together with the fact that $c_1(S, \pi) = v(g_1)$, we get

$$\begin{aligned} v(G) &= v(g_1) + \sum_{t=2}^{m} v(g_t) \\ &= c_1(S, \pi) + \sum_{t=2}^{m} (c_t(S, \pi) + c_{t-1}(S, \pi)) \\ &= c_m(S, \pi) + 2 \cdot \sum_{t=1}^{m-1} c_t(S, \pi). \end{aligned}$$

Algebraic manipulation gives us

$$\frac{v(G) + c_m(S, \pi)}{2} = \sum_{t=1}^{m} c_t(S, \pi) = \mathsf{cost}(S, \pi).$$

From Lemma 1, we know that

$$\mathsf{OPT}(\mathcal{I}) \geq \frac{1}{2} \cdot v(G).$$

If $c_m(S,\pi) \leq \frac{v(G)}{2}$, then

$$\begin{aligned}\frac{\mathsf{cost}(S,\pi)}{\mathsf{OPT}(\mathcal{I})} &\leq \frac{\left(v(G) + c_m(S,\pi)\right)/2}{v(G)/2} \\ &\leq \frac{v(G) + v(G)/2}{v(G)} = \frac{3}{2}.\end{aligned}$$

If $c_m(S,\pi) > \frac{v(G)}{2}$, then given that $\mathsf{OPT}(\mathcal{I}) \geq v(g_m) \geq c_m(S,\pi)$, we have

$$\mathsf{OPT}(\mathcal{I}) \geq v(g_m) \geq c_m(S,\pi).$$

To see why $\mathsf{OPT}(\mathcal{I}) \geq v(g_m)$: For any order π and sequence S, let g_τ be a maximum-valued item (so $v(g_\tau) = V_{\max} = v(g_m)$). If $\tau = 1$ then $\mathsf{cost}(S,\pi) \geq c_1 = V_{\max}$. If $\tau \geq 2$, then by the inequality $c_{\tau-1}+c_\tau \geq v(g_\tau)$,[7] we get $\mathsf{cost}(S,\pi) \geq c_{\tau-1} + c_\tau \geq V_{\max}$. Thus, $\mathsf{OPT}(\mathcal{I}) \geq V_{\max} = v(g_m)$.

Using this, we obtain

$$\begin{aligned}\frac{\mathsf{cost}(S,\pi)}{\mathsf{OPT}(\mathcal{I})} &= \frac{(v(G) + c_m(S,\pi))/2}{\mathsf{OPT}(\mathcal{I})} \\ &\leq \frac{(v(G) + c_m(S,\pi))/2}{c_m(S,\pi)} \\ &= \frac{1}{2}\left(\frac{v(G)}{c_m(S,\pi)} + 1\right) \\ &< \frac{1}{2}(2+1) = \frac{3}{2},\end{aligned}$$

where the strict inequality follows from the fact that $c_m(S,\pi) > v(G)/2 \implies v(G)/c_m(S,\pi) < 2$.

Thus, in both cases, we get that

$$\mathsf{cost}(S,\pi) \leq \frac{3}{2} \cdot \mathsf{OPT}(\mathcal{I}),$$

as desired.

C.4 Proof of Theorem 7

Consider any instance $\mathcal{I} = (N, G, \mathbf{v})$ in which agents have identical valuations. Let $\pi = (g_1, \ldots, g_m)$ be the order produced by Algorithm 2 (i.e., the goods are labeled so that $v(g_1) \leq \cdots \leq v(g_m)$), and let $S = (s_1, \ldots, s_m)$ be the allocation

[7] To see this inequality, let $D_{t-1} = v(A_1^{t-1}) - v(A_2^{t-1})$. After item g_t is allocated, $D_t = D_{t-1} \pm v(g_t)$. Thus, $c_{t-1}(S,\pi) + c_t(S,\pi) = |D_{t-1}| + |D_t| \geq |D_t - D_{t-1}| = v(g_t)$.

sequence induced by Algorithm 2 on this order. For each $t \in [m]$ and $i \in N$, let A_i^t denote agent i's bundle after the first t goods of π have been allocated.

For each $t \in [m]$, define

$$U_t := \max_{i \in N} v(A_i^t) \text{ and } L_t := \min_{i \in N} v(A_i^t).$$

Since valuations are identical, for every $t \in [m]$ we have

$$c_t(S, \pi) = \max_{i,j \in N} \max\{0,\, v(A_j^t) - v(A_i^t)\} = U_t - L_t.$$

Indeed, for any $i, j \in N$, $v(A_j^t) - v(A_i^t) \leq U_t - L_t$, and equality is achieved by choosing $j \in \arg\max_{\ell \in N} v(A_\ell^t)$ and $i \in \arg\min_{\ell \in N} v(A_\ell^t)$.

We now prove that

$$c_t(S, \pi) \leq v(g_t) \text{ for all } t \in [m],$$

by induction on t.

Base case ($t = 1$). After allocating g_1, some agent has value $v(g_1)$, and since $n \geq 2$, at least one agent has value 0. Thus, $U_1 = v(g_1)$ and $L_1 = 0$. Therefore,

$$c_1(S, \pi) = U_1 - L_1 = v(g_1).$$

Inductive step. Fix $t \in \{2, \ldots, m\}$ and assume $c_{t-1}(S, \pi) \leq v(g_{t-1})$. Since Algorithm 2 sorts goods by nondecreasing value, $v(g_{t-1}) \leq v(g_t)$, and thus

$$U_{t-1} - L_{t-1} = c_{t-1}(S, \pi) \leq v(g_t),$$

which implies

$$U_{t-1} \leq L_{t-1} + v(g_t).$$

Algorithm 2 allocates g_t to some agent $j \in \arg\min_{i \in N} v(A_i^{t-1})$, so $v(A_j^{t-1}) = L_{t-1}$. After allocating g_t we have

$$v(A_j^t) = v(A_j^{t-1}) + v(g_t) = L_{t-1} + v(g_t),$$

and for every $i \neq j$, $v(A_i^t) = v(A_i^{t-1}) \geq L_{t-1}$. Thus,

$$L_t = \min_{i \in N} v(A_i^t) \geq L_{t-1}.$$

Moreover, since $v(A_j^t) = L_{t-1} + v(g_t) \geq U_{t-1}$ and all other bundle values remain unchanged (and are therefore at most U_{t-1}), we have

$$U_t = \max_{i \in N} v(A_i^t) = v(A_j^t).$$

Therefore,

$$\begin{aligned} c_t(S, \pi) = U_t - L_t = v(A_j^t) - L_t &= L_{t-1} + v(g_t) - L_t \\ &\leq v(g_t), \end{aligned}$$

where the last inequality follows from the fact that $L_t \geq L_{t-1}$. This completes the induction.

Summing the per-round bound gives us

$$\mathsf{cost}(S,\pi) = \sum_{t=1}^{m} c_t(S,\pi) \leq \sum_{t=1}^{m} v(g_t) = v(G),$$

since π is a permutation of G. By Lemma 1, $\mathsf{OPT}(\mathcal{I}) \geq (1 - 1/n)\, v(G)$, and therefore

$$\frac{\mathrm{cost}(S,\pi)}{\mathsf{OPT}(\mathcal{I})} \leq \frac{v(G)}{(1-1/n)\, v(G)} = \frac{n}{n-1}.$$

so Algorithm 2 is a $n/(n-1)$-approximation.

C.5 Tightness of Identical-Lowest-First Analysis for $n = 2$.

We provide an example for which the Identical-Lowest-First algorithm has an approximation ratio of 3/2 for $n = 2$

Example 3. Consider an instance with $n = 2$ and $m = 3$ where agents have identical valuations and the multiset of good values is $\{1, 1, 2\}$.

- *Identical-Lowest-First (Algorithm 2):* Sorting gives the order $\pi = (1, 1, 2)$. Using the algorithm with its stated tie-breaking rule (lowest-index agent), the induced allocation sequence is $S = (1, 2, 1)$, which gives us $c_1(S,\pi) = 1, c_2(S,\pi) = 0, c_3(S,\pi) = 2$, and $\mathsf{cost}(S,\pi) = 3$.
- *Optimal:* Let the order be $\pi' = (1, 2, 1)$ and allocate according to $S' = (1, 2, 1)$. Then, $c_1(S',\pi') = 1, c_2(S',\pi') = 1, c_3(S',\pi') = 0$, and so $\mathsf{cost}(S',\pi') = 2$. By Lemma 1 (with $n = 2$), $\mathsf{OPT}(\mathcal{I}) \geq (1 - \frac{1}{2})\, v(G) = 2$, so this is optimal.

Therefore, on this instance Algorithm 2 has approximation ratio 3/2, matching the bound in Theorem 6.

C.6 Approximation Ratio Lower Bound of $\frac{2n-1}{2n-2}$ for $n \geq 3$.

We provide an example for which the Identical-Lowest-First algorithm has an approximation ratio of $\frac{2n-1}{2n-2}$ for $n \geq 3$.

Example 4. Fix any $n \geq 3$. Consider an instance with identical valuation function v and $m = n + 1$ goods. Let two goods have value 1 and $n - 1$ goods have value 2. Thus, we have that $v(G) = 2n$.

- *Identical-Lowest-First (Algorithm 2):* Sorting gives the order $\pi = (1, 1, 2, 2, \ldots, 2)$. Using the algorithm with its stated tie-breaking rule (lowest-index agent), after allocating the first two goods of value 1, we have bundle values $(1, 1, 0, \ldots, 0)$. Then the next $n-2$ goods each of value 2 are allocated one by one to agents $3, 4, \ldots, n$, giving us bundle values $(1, 1, 2, \ldots, 2)$

after round n; the final 2 is then allocated to agent 1, giving us $(3, 1, 2, \ldots, 2)$ after round $n+1$. Let S be the induced allocation sequence.
Under identical valuations, the per-round cost is the gap between the maximum and minimum bundle values. Thus the maximum envy values are $c_1(S, \pi) = 1$, $c_2(S, \pi) = 1$, $c_t(S, \pi) = 2$ (for $t = 3, \ldots, n-1$, $c_n = 1$), and $c_{n+1}(S, \pi) = 2$. Therefore,

$$\begin{aligned}\mathsf{cost}(S, \pi) &= \sum_{t=1}^{n+1} c_t(S, \pi) \\ &= 1 + 1 + 2(n-3) + 1 + 2 = 2n - 1.\end{aligned}$$

– *Optimal solution.* Consider the arrival order $\pi^* = (1, 2, 2, \ldots, 2, 1)$, and allocate the first and last value 1 goods to agent 1, while allocating the $n-1$ value 2 goods to agents $2, 3, \ldots, n$ (one each). Denote the resulting allocation sequence by S^*. Then the per-round maximum envy values are $c_1(S^*, \pi^*) = 1$, $c_t(S^*, \pi^*) = 2$ for $t = 2, \ldots, n-1$, $c_n(S^*, \pi^*) = 1$, $c_{n+1}(S^*, \pi^*) = 0$. Consequently, $\mathsf{cost}(S^*, \pi^*) = 1 + 2(n-2) + 1 + 0 = 2n - 2$. By Lemma 1, $\mathsf{OPT}(\mathcal{I}) \geq (1 - \frac{1}{n})\, v(G) = (1 - \frac{1}{n}) \cdot 2n = 2n - 2$, giving us $\mathsf{OPT}(\mathcal{I}) = 2n - 2$.

Thus, on this instance, Algorithm 2 has approximation ratio

$$\frac{\mathsf{cost}(S, \pi)}{\mathsf{OPT}(\mathcal{I})} = \frac{2n-1}{2n-2}.$$

C.7 Proof of Theorem 8

Fix an instance $\mathcal{I} = (N, G, v)$ with identical valuations, where $n \geq 2$. Let Algorithm 2 order the goods as $\pi = (g_1, \ldots, g_m)$ with $v(g_1) \leq \cdots \leq v(g_m)$, and let S be the induced allocation sequence. For each $t \in \{0, 1, \ldots, m\}$ define

$$L_t := \min_{i \in N} v(A_i^t) \quad \text{and} \quad U_t := \max_{i \in N} v(A_i^t)$$

where A_i^t denotes agent i's bundle after round t (and $L_0 = U_0 = 0$). Under identical valuations, $c_t(S, \pi) = U_t - L_t$.

We claim that $c_t(S, \pi) \leq v(g_t)$ for all $t \in [m]$. For $t = 1$, after allocating g_1 we have $U_1 = v(g_1)$ and $L_1 = 0$ (since $n \geq 2$ means some agent is still empty after round 1), thus, $c_1(S, \pi) = v(g_1)$. Now fix $t \geq 2$ and assume inductively that $c_{t-1}(S, \pi) \leq v(g_{t-1})$. Since the goods are sorted, $v(g_{t-1}) \leq v(g_t)$, so $c_{t-1}(S, \pi) \leq v(g_t)$. Algorithm 2 allocates g_t to an agent j with minimum bundle value at time $t-1$, i.e., $v(A_j^{t-1}) = L_{t-1}$. After allocation,

$$v(A_j^t) = L_{t-1} + v(g_t) \geq \ L_{t-1} + c_{t-1}(S, \pi) = \ U_{t-1},$$

so j becomes a (weak) maximizer and therefore $U_t = L_{t-1} + v(g_t)$. Also $L_t \geq L_{t-1}$. Thus

$$c_t(S, \pi) = U_t - L_t = (L_{t-1} + v(g_t)) - L_t \leq v(g_t),$$

proving the claim.

In particular, $c_m(S, \pi) \le v(g_m) = V_{\max}$.

From $U_t = L_{t-1} + v(g_t)$ we have, for every $t \in [m]$, $c_t(S, \pi) = U_t - L_t = (L_{t-1} + v(g_t)) - L_t$; equivalently, $v(g_t) = c_t(S, \pi) + (L_t - L_{t-1})$.

Summing over $t = 1, \ldots, m$ gives us

$$\begin{aligned} v(G) = \sum_{t=1}^{m} v(g_t) &= \sum_{t=1}^{m} c_t(S, \pi) + \sum_{t=1}^{m} (L_t - L_{t-1}) \\ &= \mathsf{cost}(S, \pi) + L_m, \end{aligned}$$

and thus

$$\mathsf{cost}(S, \pi) = v(G) - L_m. \tag{8}$$

Let $i^* \in \arg\min_{i \in N} v(A_i^m)$, so $v(A_{i^*}^m) = L_m$, while every other bundle is $\le L_m + c_m$. Thus,

$$\begin{aligned} v(G) = \sum_{i \in N} v(A_i^m) &\le L_m + (n-1)(L_m + c_m) \\ &= nL_m + (n-1)c_m. \end{aligned}$$

Rearranging, we get

$$L_m \ge \frac{v(G) - (n-1)c_m}{n}.$$

Plugging this into (8) gives us

$$\begin{aligned} \mathsf{cost}(S, \pi) = v(G) - L_m &\le v(G) - \frac{v(G) - (n-1)c_m}{n} \\ &= \left(1 - \frac{1}{n}\right) v(G) + \left(1 - \frac{1}{n}\right) c_m. \end{aligned}$$

Using $c_m \le V_{\max}$, we get

$$\mathsf{cost}(S, \pi) \le \left(1 - \frac{1}{n}\right) v(G) + \left(1 - \frac{1}{n}\right) V_{\max}.$$

By Lemma 1, $\mathsf{OPT}(\mathcal{I}) \ge \left(1 - \frac{1}{n}\right) v(G)$. Therefore,

$$\mathsf{cost}(S, \pi) \le \mathsf{OPT}(\mathcal{I}) + \left(1 - \frac{1}{n}\right) V_{\max},$$

which completes the proof.

C.8 Tightness of the Additive Approximation Guarantee.

We provide an example showing that the additive error term in Theorem 8 is tight.

Example 5. Fix any $n \geq 2$. Consider an instance with identical valuation function v and the following multiset of good values: for each $k \in \{1, \dots, n-1\}$ there are exactly n goods of value k, and there are $n-1$ goods of value n. Thus $V_{\max} = n$, the number of goods is $m = n(n-1) + (n-1) = n^2 - 1$, and

$$\begin{aligned} v(G) = n\sum_{k=1}^{n-1} k + (n-1)\cdot n &= n \cdot \frac{n(n-1)}{2} + n(n-1) \\ &= \frac{n(n-1)(n+2)}{2}. \end{aligned}$$

- Identical-Lowest-First *(Algorithm 2)* Sorting gives the order

$$\pi = (\underbrace{1, \dots, 1}_{n}, \underbrace{2, \dots, 2}_{n}, \dots, \underbrace{n-1, \dots, n-1}_{n}, \underbrace{n, \dots, n}_{n-1}).$$

For each $k \in \{1, \dots, n-1\}$, when the n goods of value k arrive, all bundle values are equal at the start of this block, so the greedy rule gives one such good to each agent. Thus, during this block, the gap (maximum minus minimum bundle value) is $c_t(S, \pi) = k$ for the first $n-1$ arrivals and $c_t(S, \pi) = 0$ on the n-th, contributing $(n-1)k$ to the cumulative cost. After the first $n(n-1)$ arrivals, every agent has value

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}.$$

The remaining $n-1$ goods (each of value n) are then given to $n-1$ distinct agents; after each such round, the gap is exactly n. Therefore,

$$\begin{aligned} \mathsf{cost}(S, \pi) &= \sum_{k=1}^{n-1}(n-1)k + (n-1)\cdot n \\ &= (n-1)\left(\sum_{k=1}^{n-1} k + n\right) \\ &= (n-1)\left(\frac{n(n-1)}{2} + n\right) \\ &= \frac{n(n-1)(n+1)}{2}. \end{aligned}$$

- *Optimal solution.* We show a pair (S^*, π^*) that achieves the universal lower bound of Lemma 1. Define an arrival order π^* in $(n-1)$ blocks as follows: for each $k = n-1, n-2, \dots, 1$, append the block

$$\pi^* = (k,\ \underbrace{(k+1), (k+1), \dots, (k+1)}_{k \text{ times}},\ \underbrace{k, k, \dots, k}_{(n-k) \text{ times}}).$$

Note that for each $k \leq n-1$, value k appears $n-k+1$ times in block k (once at the start, and $n-k$ times at the end), and appears $k-1$ times in block $k-1$.

Thus, the total number of times value k appears is $(n-k+1)+(k-1)=n$. Also, value n appears exactly $n-1$ times in block $n-1$.

Let S^* allocate each arriving good to an agent with currently minimum bundle value (break ties arbitrarily).

Claim. After the block for k is processed, the multiset of bundle values is

$$\{\underbrace{L_k,\dots,L_k}_{k \text{ agents}},\underbrace{L_k+(k-1),\dots,L_k+(k-1)}_{n-k \text{ agents}}\},$$

where $L_{n-1}=n$ and $L_{k-1}=L_k+k$ for $k\in\{2,\dots,n-1\}$.

Proof of claim. For $k=n-1$, the first block is $(n-1,\underbrace{n,\dots,n}_{n-1},n-1)$, and allocating each good to a minimum-valued agent yields $n-1$ agents at value n and one agent at value $2n-2=n+(n-2)$, so $L_{n-1}=n$. For the inductive step, assume the claim holds after the block for k (so the current gap is $k-1$). The next block is

$$\bigl(k-1,\underbrace{k,\dots,k}_{k-1},\underbrace{k-1,\dots,k-1}_{n-k+1}\bigr).$$

The first $(k-1)$ goes to a minimum agent, bringing her up to the current maximum. Then the $(k-1)$ goods of value k are given to the remaining minimum agents, making them the new maxima. Finally, each of the $n-k+1$ goods of value $(k-1)$ is given to a (new) minimum agent; after all are assigned, we obtain $k-1$ agents at value L_k+k and $n-k+1$ agents at value $(L_k+k)+(k-2)$, i.e., the claim with $L_{k-1}=L_k+k$. □

In particular, after the final block $(k=1)$, all agents have equal value L_1. Unrolling the recurrence,

$$L_1=n+(n-1)+\cdots+2=\frac{(n-1)(n+2)}{2}=\frac{v(G)}{n}.$$

Moreover, each arriving good is allocated to an agent with minimum bundle value. As in the proof of Lemma 1, this implies

$$\begin{aligned}\mathsf{cost}(S^*,\pi^*)&=v(G)-\min_i v(A_i)\\&=v(G)-\frac{v(G)}{n}=\left(1-\frac{1}{n}\right)v(G).\end{aligned}$$

By Lemma 1, $\mathsf{OPT}(\mathcal{I})\ge(1-\frac{1}{n})v(G)$, so we conclude $\mathsf{OPT}(\mathcal{I})=(1-\frac{1}{n})v(G)$.

Finally, we have that

$$\begin{aligned}
&\mathsf{cost}(S,\pi) - \mathsf{OPT}(\mathcal{I}) \\
&= \left(v(G) - \frac{n(n-1)}{2}\right) - \left(v(G) - \frac{v(G)}{n}\right) \\
&= \frac{v(G)}{n} - \frac{n(n-1)}{2} \\
&= \frac{(n-1)(n+2)}{2} - \frac{n(n-1)}{2} \\
&= n - 1 = \left(1 - \frac{1}{n}\right) V_{\max}.
\end{aligned}$$

Thus, the additive term $\left(1 - \frac{1}{n}\right) V_{\max}$ cannot be improved in general.