Fairly Allocating (Contiguous) Dynamic Indivisible Items with Few Adjustments

Mingwei Yang
Peking University
Beijing, China
yangmingwei@pku.edu.cn

ABSTRACT

We study the problem of dynamically allocating indivisible items to a group of agents in a fair manner. Due to the negative results to achieve fairness when allocations are irrevocable, we allow adjustments to make fairness attainable with the objective to minimize the number of adjustments. For restricted additive or general identical valuations, we show that *envy-freeness up to one item (EF1)* can be achieved at no cost. For additive valuations, we give an EF1 algorithm that requires $O(mT)$ adjustments, where m is the maximum number of different valuations for items among all agents.

We further impose the contiguity constraint on items and require that each agent obtains a consecutive block of items. We present extensive results to achieve either *proportionality* with an additive approximate factor or EF1. In particular, we establish matching lower and upper bounds for identical valuations to achieve approximate proportionality. We also show that it's hopeless to make any significant improvement when valuations are nonidentical.

Our results exhibit a large discrepancy between the identical and nonidentical cases in both contiguous and noncontiguous settings. All our positive results are computationally efficient.

1 INTRODUCTION

Fair division is one of the most fundamental and well-studied topics in Computational Social Choice with much significance and several applications in many real-life scenarios [16, 25]. Generally, there are some resources and our objective is to divide them among a group of competing agents in a fair manner. Arguably, the most compelling fairness notion is *envy-freeness*, which is defined as each agent weakly preferring his own bundle to any other agent's bundle.

However, in the indivisible regime, the existence of envy-free solutions is not guaranteed. For instance, if there are two agents but only one item, the agent who receives the item is certainly envied by the other one. One of the natural relaxations of envy-freeness is *envy-freeness up to one item (EF1)* proposed by [12]. In an EF1 allocation, every agent i may envy another agent j, but the envy could be eliminated by removing one item from agent j's bundle. EF1 allocations not only exist in any instance but also can be computed in polynomial time [20].

Imposing some constraints to the model will make the fairness objective less tractable. A series of works focus on the setting where items lie on a line and each agent obtains a consecutive block of items. For monotone valuations, [8] presents polynomial-time algorithms to compute contiguous EF1 allocations for any number of agents with identical valuations or at most three agents, and shows the existence of contiguous EF1 allocations for four agents. It

is not until recently that the existence of contiguous EF1 allocations for any number of agents is proved by [19].

Another natural generalization assumes that items arrive online. Furthermore, the types of future items are unknown and decisions have to be made instantly. Recently, [17] investigates this model with the requirement that the allocations returned after the arrival of each item are EF1. However, due to the negative results against adversaries when allocations are irrevocable [7], adjustments are necessary to achieve EF1 deterministically. Notably, the $O(T^2)$ of adjustments is attained trivially by redistributing all items in each round, where T is the number of items. [17] shows that the $\Omega(T)$ of adjustments is inevitable for more than two agents, even if the information of all items is known upfront. On the positive side, they give two algorithms with respectively $O(T^{1.5})$ and $O(nmT)$ adjustments, where n is the number of agents and m denotes the maximum number of distinct valuations for items among all agents.

In this work, we adopt the model proposed by [17]. Furthermore, we also impose the contiguity constraint on items. More precisely, items are arranged on a line by the order they arrive and each agent obtains a contiguous block of items. To motivate this, consider a library that has several bookshelves with books of the same types being placed together. Moreover, the numbers of books of two different types should not differ by a large amount. With more and more bookshelves being deployed, the library needs to reallocate a consecutive block of bookshelves to each certain type of book and redistribute the books according to the new allocation. In this case, the objective is to minimize the cost of moving the books or equivalently, the number of adjustments. We study the number of adjustments necessary to achieve some fairness guarantee in both contiguous and noncontiguous settings.

1.1 Our Contributions

We first describe our positive results in the noncontiguous setting. We show that if valuations are limited to be restricted additive [1] or general identical, EF1 can be achieved at no cost in the sense that no adjustments are required. In addition, we give an EF1 algorithm for additive valuations that requires $O(mT)$ adjustments. Note that if m is a constant, it matches the $\Omega(T)$ lower bound and thus is optimal.

With the contiguity constraint, EF1 is too stringent and we start with a weaker fairness notion. [26] shows that contiguous $(\frac{n-1}{n} \cdot v^{\max})$-*proportional (PROPa)* allocations can be computed efficiently, where $v^{\max}$ is the maximum valuation of items, and this additive approximate factor is tight in some sense. We first consider identical valuations. We give a PROPa algorithm that requires $O(nT)$ adjustments. Then we establish the matching lower bound to show that our algorithm is optimal. When it comes to EF1, for

Table 1: Results for the contiguous setting with additive valuations. The result for two agents in the identical case also works for general valuations.

	Identical				Nonidentical	
	PROPa	EF1			PROPa	EF1
		Lower	Upper			
$n = 2$		$\Theta(T)$				$\Theta(T^2)$
$n > 2$	$\Theta(nT)$	$\Omega(nT)$	$O\left(\frac{R}{L} \cdot n^2 T\right)$		$\Omega(T^2/n)$	$\Theta(T^2)$

two agents with general valuations, we show that EF1 is achievable with $O(T)$ adjustments, which is optimal. If the valuation of each item is assumed to lie in $[L, R]$ such that $0 < L \leq R$, we give an EF1 algorithm that requires $O\left((R/L) \cdot n^2 T\right)$ adjustments. By contrast, in the nonidentical case, we show that it is hopeless to make any significant improvement. Specifically, we give instances to establish the lower bounds of $\Omega(T^2/n)$ to achieve PROPa and $\Omega(T^2)$ for EF1. Our results in the contiguous setting with additive valuations are summarized in Table 1.

Our results exhibit a large discrepancy between the identical and nonidentical cases in both contiguous and noncontiguous settings. In addition, all the algorithms given in this paper can be implemented in polynomial time.

1.2 Related Work

Even though both divisible and indivisible models are extensively studied, we only focus on the indivisible setting, which is more relevant to our work.

Dynamic fair division. Our work belongs to the vast literature of *dynamic* or *online* fair division [4]. Under the assumption that valuations are normalized to $[0, 1]$ and items are allocated irrevocably, [7] gives an algorithm that achieves maximum envy of $\tilde{O}(\sqrt{T/n})$ deterministically and shows that this bound is tight asymptotically. To bypass the negative results, motivated by the notion of *disruptions* [13, 14], [17] introduces adjustments to achieve EF1 deterministically.

Without the ability to adjust, another popular measure of compromise is assuming that agents' valuations are stochastic. When the valuation of each agent to each item is drawn i.i.d. from some distribution, the algorithm of allocating each item to the agent with the maximum valuation for it is envy-free with high probability and ex-post Pareto optimal [21, 22]. Besides, [5] provides the same guarantee for asymmetric agents, i.e., the valuation of an item for each agent is independently drawn from an agent-specific distribution. Furthermore, [27] assumes that the valuations of different agents for the same item are correlated and shows that Pareto efficiency and fairness are also compatible. More recently, [6] studies the partial-information setting where only the ordinal information is revealed. Another series of works [2, 3] resort to random allocations to achieve ex-ante fairness together with some efficiency and incentive guarantees.

Fair division of contiguous blocks. The online fair division model with the contiguity requirement concerned in our work is a strict extension of the fair division of contiguous blocks problem. In the offline setting, [8, 19, 24] study the existence of contiguous EF1 allocations and [26] focuses on the approximate versions of proportionality, envy-freeness, and equitability. More recently, [23] designs an algorithm to compute a contiguous EQ1 allocation with the egalitarian welfare guarantee. Besides, the *price of fairness* of contiguous allocations for both goods and chores are respectively established by [26] and [18]. More generally, [9, 10] allow the connectivity relation among items to form a graph that possesses some structures with a path being a special case.

2 PRELIMINARIES

There are n agents and let $N = [n]$ be the set of all agents. There are T items arriving one by one. Let g_t be the t-th item and $M_t = \{g_1, \ldots, g_t\}$ be the set of the first t items that arrive. In particular, define $M = M_T$. Let $M_{l,r} = \{g_{l+1}, \ldots, g_r\}$ be the set of items whose indexes are in $[l + 1, r]$ with $M_{0,r} = M_r$. Each agent i has a non-negative valuation $v_i : 2^M \rightarrow \mathbb{R}_+$, which reflects his evaluation for each subset of items. Denote $m = \max_{i \in N} |\{v_i(g) \mid g \in M\}|$ as the maximum number of distinct valuations for items among all agents. To simplify the notations, for $S \subseteq M$ and $g \in M$, we use $v_i(g)$, $v_i(S + g)$ and $v_i(S - g)$ to respectively represent $v_i(\{g\})$, $v_i(S \cup \{g\})$ and $v_i(S \setminus \{g\})$. An allocation for a set of items M' is a tuple $(A_1, A_2, \ldots, A_n)$ satisfying $A_i \cap A_j = \emptyset$ for all $i \neq j$ and $A_1 \cup A_2 \cup \ldots \cup A_n = M'$, where A_i is agent i's bundle.

After the arrival of each item, algorithms should return an allocation for all the existing items. We use the number of adjustments to measure the performance of algorithms. Formally, let the allocations returned by an algorithm be $A^1, \ldots, A^T$, where A^t is an allocation for M^t. The number of adjustments used in round $t \geq 1$ is defined as the number of items $g \in M_{t-1}$ such that g belongs to different agents in A^{t-1} and A^t. The number of adjustments required by an algorithm is defined as the total number of adjustments used across all T rounds.

Now we enumerate different types of valuation functions that are considered in this paper. A valuation v is *additive* if for all $S \subseteq M$, we have $v(S) = \sum_{g \in S} v(g)$. Valuations $v_1, \ldots, v_n$ are *restricted additive* if they are additive and each item g has an inherent valuation v_g so that $v_i(g) \in \{0, v_g\}$ for every agent i. A valuation v is *general* if it is only required to be monotone, i.e, $v(S + g) \geq v(S)$ for all $S \subseteq M$ and $g \in M$. Valuations $v_1, \ldots, v_n$ are *identical* if we have $v_1(\cdot) = \cdots = v_n(\cdot)$. We assume valuations to be additive if without specification.

We formally define the fairness criteria including EF1 and PROPa in the following.

Definition 2.1 (EF1). An allocation A is *envy-free up to one item (EF1)* if for any $i, j \in N$, either $v_i(A_i) \geq v_i(A_j)$ or there exists $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j - g)$.

Definition 2.2 (PROPa). Let $v^{\max} = \max_{i \in N, g \in M} v_i(g)$. An allocation A is $(\frac{n-1}{n} \cdot v^{\max})$-*proprtional (PROPa)* if we have

$$v_i(A_i) \geq \frac{1}{n} \cdot v_i(M) - \frac{n-1}{n} \cdot v^{\max}$$

for any $i \in N$.

3 NONCONTIGUOUS SETTING

In this section, we consider the noncontiguous setting. We first present greedy algorithms for restricted additive and general identical valuations to show the surprising result that EF1 can be achieved at no cost in both cases. Next, we give an algorithm that requires $O(mT)$ adjustments. In Appendix A, we show that when there are only two agents, we can achieve the optimal $O(T)$ adjustments even if the valuations are allowed to be negative.

3.1 Restricted Additive or General Identical Valuations

For additive valuations, [17] proves that every EF1 algorithm requires at least $\Omega(T)$ adjustments. Surprisingly, we show that if we limit the valuations to restricted additive or general identical valuations, EF1 can be achieved with no cost: simple greedy algorithms is EF1 and require no adjustments for these two cases. Notably, restricted additive valuations can be viewed as a slightly more general version of additive identical valuations by adding the assumption that each agent may be indifferent to some items. As a consequence, there is a large discrepancy between the identical and nonidentical cases in the noncontiguous setting.

Firstly, we give a greedy algorithm for restricted additive valuations in Algorithm 1. The idea of Algorithm 1 is simply giving the new item to the agent with the minimum valuation among the agents with positive valuations for it. Without loss of generality, we assume that for every item, there exists an agent that has a positive valuation for it. Since otherwise all agents are indifferent to the item and we just discard it.

Algorithm 1 Greedy Algorithm for Restricted Additive Valuations

---

Require: v_i for each agent a_i
1: $A = \emptyset$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad k \leftarrow \arg\min_{i:v_i(g_t)>0} v_i(A_i)$
4: $\quad A_k \leftarrow A_k \cup \{g_t\}$
5: $\quad A^t \leftarrow A$
6: **return** $[A^1, A^2, \ldots, A^T]$

Theorem 3.1. *Algorithm 1 is an EF1 algorithm for restricted additive valuations that requires no adjustments.*

PROOF. Algorithm 1 makes no adjustments since once an item is assigned to an agent, it will be allocated to the same agent in the future. Now we prove that every returned allocation is EF1. First we prove the property that, for any $i, j \in N$, $v_i(A_j) \leq v_j(A_j)$ always holds. In other words, each bundle is valued the most by the agent who owns it. Since Algorithm 1 only assign an item to the agent who has positive valuation for it, for any agent j, we have $v_j(A_j) = \sum_{g \in A_j} v_g$, where v_g is the inherent valuation of item g. Since agent i may be indifferent to some items in A_j, we have $v_i(A_j) \leq \sum_{g \in A_j} v_g = v_j(A_j)$.

Next, we prove that each allocation is EF1 by induction. Before the arrival of the first item, the allocation is trivially EF1. We will show that if the current allocation is EF1, then after the arrival of a new item, the newly generated allocation is also EF1. When a new

item g comes, define $N_1 = \{i \in N \mid v_i(g) > 0\}$ as the set of agents who are interested in g and $N_2 = N \setminus N_1$ as the set of agents who are not. Since the agents in N_2 are indifferent to g, their existence is not likely to lead to the violation of EF1 no matter which agent g is assigned to. Therefore, we are only concerned about the agents in N_1. Let $k \in N_1$ be the agent who gets g. By the choice of k, for any $i \in N_1$, we have $v_i(A_k) \leq v_k(A_k) \leq v_i(A_i)$ before the assignment of g, which means that each agent in N_1 does not envy agent k. Thus after allocating g to agent k, each agent in N_1 will not envy agent k up to one item. Therefore, the new allocation is also EF1. □

Then we give an algorithm for general identical valuations in Algorithm 2. The allocating strategy of Algorithm 2 is assigning the new item to the agent with the minimum valuation.

Algorithm 2 Greedy Algorithm for General Identical Valuations

---

Require: A valuation v for all agents
1: $A = \emptyset$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad k \leftarrow \arg\min_i v(A_i)$
4: $\quad A_k \leftarrow A_k \cup \{g_t\}$
5: $\quad A^t \leftarrow A$
6: **return** $[A^1, A^2, \ldots, A^T]$

Theorem 3.2. *Algorithm 2 is an EF1 algorithm for general identical valuations that requires no adjustments.*

PROOF. Algorithm 2 makes no adjustments since once an item is assigned to an agent, it will be allocated to the same agent in the future. Now we prove that every returned allocation is EF1 by induction. When there are no items, the allocation is trivially EF1. We will prove that if the current allocation is EF1, then after the arrival of a new item, the newly generated allocation is also EF1. When a new item g comes, suppose that agent k is the agent with the minimum valuation and thus gets g. Since all agents do not envy agent k before the assignment by the choice of k, they will not envy agent k up to one item in the new allocation. Therefore, the new allocation is also EF1. □

3.2 Additive Valuations

For additive valuations, we present an EF1 algorithm that requires $O(mT)$ adjustments in Algorithm 3. Note that when m is a constant, Algorithm 3 is optimal.

The idea is based on the round-robin algorithm. Recall that in the round-robin algorithm, agents pick their favorite items alternately in each round. The resulting allocation is EF1 since, in each round, every agent prefers the item that he picks in this round to any item picked by some agent in the next round. Based on this observation, we aim to maintain the above property for the allocation in each round[1]. To this end, we maintain a round-robin structure by dividing items into multiple layers satisfying that each agent gets exactly one item in each layer. In particular, each agent gets no more than one item in the last layer. Moreover, it holds that in each layer, each

---

[1]This approach is applied in [11] to show that the allocation induced by recursively finding a maximum-weight matching between agents and items is EF1.

agent prefers his item in this layer to any item in the next layer. In this case, the allocation induced by this structure is EF1.

Algorithm 3 is the realization of the above idea. We use C^k to store the allocation in layer k, where C_a^k denotes the item obtained by agent a. In particular, let $C_a^i = 0$ if agent i does not obtain any item in layer i, and let $v_i(0) = 0$ for all $i \in N$. When a new item arrives, the algorithm updates each layer sequentially. Whenever there is an agent who prefers the new item to his item in the current layer, we swap the new item with his item in the current layer (Line 4–Line 6). After all the replacements in the current layer are finished, we move to the next layer and repeat the above process. In the last layer, we assign the new item to an arbitrary agent who has not obtained any item in this layer (Line 7). Finally, we return the allocation collectively induced by all layers (Line 8).

Algorithm 3 Layer Updating Algorithm

Require: v_i for each agent a_i
1: **for** $t = 1, \ldots, T$ **do**
2: $\quad k \leftarrow \lceil \frac{t}{n} \rceil$
3: $\quad$ **for** $i = 1, \ldots, k - 1$ **do**
4: $\quad\quad$ **while** $G := \{a \in N \mid v_a(g_t) > v_a(C_a^i)\} \neq \emptyset$ **do**
5: $\quad\quad\quad \widehat{a} \leftarrow \arg\min_{a \in G} v_a(C_a^i)$
6: $\quad\quad\quad \text{swap}(g_t, C_{\widehat{a}}^i)$
7: $\quad C_{t-n*(k-1)}^k \leftarrow g_t$
8: $\quad$ Let A^t be the combination of allocations $C^1, C^2, \ldots, C^k$
9: **return** $[A^1, A^2, \ldots, A^T]$

Theorem 3.3. *Algorithm 3 is an EF1 algorithm that requires $O(mT)$ adjustments.*

Proof. We first prove that throughout the algorithm, for each layer k, each agent prefers the item that he obtains in this layer to any item in layer $k + 1$, i.e., $v_i(C_i^k) \geq v_i(C_j^{k+1})$ for any $i, j \in N$. Then we show that this condition is sufficient for EF1.

Now we use induction to prove that

$$v_i(C_i^k) \geq v_i(C_j^{k+1}), \quad \forall i, j \in N, \forall k > 0. \quad (1)$$

When there are no items, (1) is trivially satisfied. We will show that if (1) holds for the current allocation, then after the arrival of a new item, (1) also holds for the newly generated allocation. Note that during the update of a layer, each replacement will strictly increase the valuation of the chosen agent for his item in this layer. Thus after each replacement, the chosen agent still prefers his item in this layer to any item in the next layer. Moreover, the update for a layer being finished indicates that each agent prefers his item in this layer to the new item. As a result, we can safely move on and update the next layer with the new item.

We show that (1) implies EF1. Suppose that $(C^1, C^2, \ldots)$ satisfies (1) and let A be the allocation induced by $(C^1, C^2, \ldots)$. Note that for any $i \in N$, we have $A_i = \{C_i^1, C_i^2, \ldots\}$. Thus for any agents $i, j \in N$, we have

$$v_i(A_i) = \sum_{k \geq 1} v_i(C_i^k) \geq \sum_{k \geq 1} v_i(C_j^{k+1}) = v_i(A_j - C_j^1),$$

which means that agent i does not envy agent j up to one item. Therefore, A is EF1.

Finally, we prove that the algorithm uses $O(mT)$ adjustments in total. Notice that the number of adjustments is upper bounded by the number of replacements. Since a replacement will make the valuation of the chosen agent for his item in the current layer strictly larger and there are at most m distinct valuations of each agent for items, the number of replacements for each agent in a certain layer is at most m. Therefore, the number of adjustments is $\lceil T/n \rceil \cdot n \cdot m = O(mT)$ since there are $\lceil T/n \rceil$ layers. □

We also notice that all the positive results in the noncontiguous setting share the property that the valuation of each agent will not decrease across the entire period. In other words, agents will only be better off if they follow the dictation of adjustments and, as a result, they have the incentive to participate. By contrast, in the contiguous setting, the incentive property is not compatible with EF1. See Appendix B for more discussion about the incentive issue.

4 CONTIGUOUS SETTING WITH IDENTICAL VALUATIONS

In this section, we assume that all items are arranged on a line by the order they arrive. Moreover, we impose the constraint on the allocation that it has to be contiguous, i.e., each agent gets a contiguous block of items. We assume valuations to be identical in this section, which can be represented by a valuation v shared by all agents, and agent i always gets the i-th block from left to right. We will establish lower and upper bounds for both PROPa and EF1. All the omitted proofs in this section can be found in Appendix D.

4.1 Upper Bounds for PROPa

In the offline model, for any line of items M and n agents with non-identical valuations $v_1, \ldots, v_n$, [26] designs an efficient algorithm to compute a contiguous allocation $A = (A_1, A_2, \ldots, A_n)$ satisfying

$$v_i(A_i) \geq \frac{1}{n} \cdot v_i(M) - \frac{n-1}{n} \cdot v_i^{\max}(M) \quad (2)$$

for all $i \in N$, where $v_i^{\max}(M) = \max_{g \in M} v_i(g)$. In particular, the computed allocation is PROPa. [26] also show that this is the best additive approximation factor even for identical valuations and without the contiguity requirement.

Lemma 4.1 ([26]). *Suppose that there are n agents and the items in M are arranged on a line. Then the contiguous allocation $(A_1, A_2, \ldots, A_n)$ satisfying (2) for all $i \in N$ always exists and can be computed in polynomial time.*

We start by describing the offline algorithm given by [26]. The algorithm maintains a current block, iterates all items sequentially, and adds the new item to the current block. In the beginning, all agents are active. Whenever there exists an agent i satisfying (2) for the current block, then the current block is obtained by agent i. After that, agent i is deactivated and the current block becomes empty. Finally, when all agents become inactive, the remaining items are assigned arbitrarily.

For identical valuations in the online setting, we give an algorithm based on the offline algorithm that achieves PROPa with $O(nT)$ adjustments in Algorithm 4. Intuitively, we intend to use the offline algorithm as a subroutine to compute an allocation in each round independently. If we successfully prove that each

separating point between two adjacent blocks is nondecreasing throughout all rounds, then the $O(nT)$ upper bound is established. A sufficient condition for the nondecreasing property of the separating points is that the constraint factor in round t, defined as $B_t := \frac{1}{n} \cdot v(M_t) - \frac{n-1}{n} \cdot v^{\max}(M_t)$, is nondecreasing. Unfortunately, this may not hold. For example, if $n = 3$ and there are two items g_1, g_2 such that $v(g_1) = 0$ and $v(g_2) = 1$, then we have $B_1 = 0$ and $B_2 = -1/3 < B_1$. The essential observation is that when $B_{t+1} < B_t$ and A^t satisfies the constraint factor B_t, adding an item to some block will only enable the allocation to be feasible for a larger constraint factor. Therefore, if $B_{t+1} < B_t$, we just arbitrarily assign the new item to obtain A^{t+1} from A^t. In fact, we will show that if $B_{t+1} < B_t^{\max}$ where $B_t^{\max} := \max_{i \in [t]} B_i$, then we are allowed to allocate the new item arbitrarily.

In Algorithm 4, for any $i \in [n]$, p_i serves as the index of the last item in the i-th block, where $p_0 = 0$ and $p_n = t$ are omitted. We update p_i sequentially by moving p_i backward until the i-th block satisfies the proportional constraint, i.e., $v(M_{p_{i-1},p_i}) \geq B_t$ (Line 5–Line 6).

Algorithm 4 PROPa Algorithm for Identical Valuations

Require: A valuation function v for all agents
1: $p_i \leftarrow 0$ for $i = 1, 2, \ldots, n-1$
2: **for** $t = 1, \ldots, T$ **do**
3:      $B_t \leftarrow \frac{1}{n} \cdot v(M_t) - \frac{n-1}{n} \cdot v^{\max}(M_t)$
4:      **for** $i = 1, \ldots, n-1$ **do**
5:          **while** $v\left(M_{p_{i-1},p_i}\right) < B_t$ **do**
6:              $p_i \leftarrow p_i + 1$
7:          $A_i^t \leftarrow M_{p_{i-1},p_i}$
8:      $A_n^t \leftarrow M_{p_{n-1},t}$
9: **return** $[A^1, A^2, \ldots, A^T]$

Theorem 4.2. *Algorithm 4 is a PROPa algorithm for identical valuations that requires $O(nT)$ adjustments.*

4.2 Upper Bounds for EF1

In this section, we consider EF1 as the fairness criterion, which, as we will show later, is stronger than PROPa. We first give an EF1 algorithm for two agents with general identical valuations that requires $O(T)$ adjustments in Algorithm 5. At each round t, we first find the minimum index i such that the total valuation of items on the left of g_i (inclusively) is at least the total valuation of the remaining items[2], i.e., $v(M_i) \geq v(M_{i,t})$ (Line 2). Then the line of items is separated into two blocks by g_i, where g_i belongs to the block with the smaller valuation (Line 3–Line 6).

Theorem 4.3. *Algorithm 5 is an EF1 algorithm for two agents with general identical valuations that requires $O(T)$ adjustments.*

Proof. First, the i specified in Line 2 is well-defined since $t \in \{j \mid v(M_j) \geq v(M_{j,t})\}$. We show that A^t is EF1 for any $t \in [T]$. Fix $t \in [T]$ and define i as in Line 2. Let $L = M_{i-1}$ and $R = M_{i,t}$, and we have $v(L + g_i) \geq v(R)$ by definition. Note that $v(R + g_i) > v(L)$

---
[2]Such i is called *lumpy tie* in [8], which possesses some desirable properties that are useful to establish contiguous fair allocations.

Algorithm 5 EF1 Algorithm for Two Agents with General Identical Valuations

---

Require: A valuation function v for all agents
1: **for** $t = 1, \ldots, T$ **do**
2:      $i = \min\{j \mid v(M_j) \geq v(M_{j,t})\}$
3:      **if** $v(M_{i-1}) \leq v(M_{i,t})$ **then**
4:          $A^t = (M_i, M_{i,t})$
5:      **else**
6:          $A^t = (M_{i-1}, M_{i-1,t})$
7: **return** $[A^1, A^2, \ldots, A^T]$

---

holds since otherwise $v(L) \geq v(R + g_i)$ and we know that $i - 1 \in \{j \mid v(M_j) \geq v(M_{j,t})\}$, which contradicts the minimality of i. Without loss of generality, suppose that $A^t = (L \cup \{g_i\}, R)$ by symmetry, which means that $v(L) \leq v(R)$. Thus agent 2 does not envy agent 1 up to one item. On the other hand, $v(L + g_i) \geq v(R)$ and thus agent 1 does not envy agent 2. As a result, A^t is EF1 for any $t \in [T]$.

It remains to prove the required number of adjustments. Define $f(t) = \min\{j \mid v(M_j) \geq v(M_{j,t})\}$ and $h(t)$ such that agent 1 obtains exactly the first $h(t)$ items in A^t. By the monotonicity of v, we know that $f(t)$ is nondecreasing. Due to the allocation rule, we have $f(t) = h(t)$ or $f(t) = h(t) - 1$. It suffices to prove that $h(t)$ is nondecreasing since in this case, the total number of adjustments is

$$\sum_{t=1}^{T-1} |h(t+1) - h(t)| = \sum_{t=1}^{T-1} (h(t+1) - h(t)) = h(T) - h(1) \leq T.$$

Fix $t < T$. If $f(t) < f(t+1)$, it holds that

$$h(t) \leq f(t) \leq f(t+1) - 1 \leq h(t+1).$$

Besides, if $f(t) = f(t+1)$ and $h(t) = f(t) - 1$, then $h(t) \leq h(t+1)$ trivially holds. Lastly, if $f(t) = f(t+1)$ and $h(t) = f(t)$, we have $v(M_{f(t)-1}) \leq v(M_{f(t),t})$. Since

$$v(M_{f(t+1)-1}) = v(M_{f(t)-1}) \leq v(M_{f(t),t})$$
$$\leq v(M_{f(t),t+1}) = v(M_{f(t+1),t+1}),$$

$g_{f(t+1)}$ is allocated to agent 1 in A^{t+1}. It follows that $h(t+1) = f(t+1) = h(t)$. Therefore, $h(t)$ is nondecreasing. □

For any number of agents, we fail to establish such a nondecreasing property for the separating points, by which we can similarly achieve $O(nT)$ adjustments. Nevertheless, under the assumption that the valuation of each item lies in some range $[L, R]$ such that $0 < L \leq R$, we employ the slightly modified version of the algorithm given by [8] for any number of agents with additive identical valuations as a subroutine to obtain an EF1 allocation in each round independently. This leads to an EF1 algorithm for additive identical valuations that requires $O\left((R/L) \cdot n^2 T\right)$ adjustments.

We present the offline algorithm in Algorithm 6 and the online algorithm in Algorithm 7. For a contiguous allocation A, define $P(A) = (\ell_0, \ldots, \ell_n)$ such that $\ell_0 = 0$ and $P_i(A) = \ell_i$ is the index of the last item in A_i. Denote the *leximin (lexicographic maximin)* allocation as the allocation that maximizes the lowest valuation among all agents, subject to which it maximizes the second-lowest one, and so on. We say that A is leximinmin if A is the leximin allocation with lexicographically minimum $P(A)$. In Algorithm 6,

we first find a contiguous leximinmin allocation[3] A as the initial allocation and fix an agent i with the minimum valuation in A. Then we iterate from 1 to $i-1$ and from n to $i+1$ separately. During each iteration j, whenever agent i envies agent j even up to one item, the item in A_j that is closest to A_i is moved to the adjacent block. The only difference between Algorithm 6 and the algorithm given by [8] is that Algorithm 6 uses the leximinmin allocation as the initial allocation rather than an arbitrary leximin allocation. Intuitively, choosing the leximinmin allocation guarantees the property that each separating point between two adjacent blocks is nondecreasing (Lemma 4.7), which is crucial for proving the upper bound.

Algorithm 6 EF1 Offline Algorithm

Require: A valuation function v for all agents and an item set M
1: Let $A = (A_1, \ldots, A_n)$ be the contiguous leximinmin allocation of M
2: Fix $i = \arg\min_{j \in N} v(A_j)$
3: **for** $j = 1, \ldots, i-1$ **do**
4:     **while** agent i envies agent j even up to one item **do**
5:         move the rightmost item of A_j to A_{j+1}
6: **for** $j = n, \ldots, i+1$ **do**
7:     **while** agent i envies agent j even up to one item **do**
8:         move the leftmost item of A_j to A_{j-1}
9: **return** A

Algorithm 7 EF1 Online Algorithm

Require: A valuation function v for all agents
1: **for** $t = 1, \ldots, T$ **do**
2:     $A^t \leftarrow$ the allocation returned by Algorithm 6 running with v and M_t
3: **return** $[A^1, A^2, \ldots, A^T]$

We first present the property of Algorithm 6.

Lemma 4.4 ([8]). *For additive identical valuations, Algorithm 6 is EF1 and can be implemented in polynomial time. Moreover, throughout Algorithm 6, A_i does not change.*

Now we give the performance of Algorithm 7.

Theorem 4.5. *Suppose that the valuations are additive identical and there exist $0 < L \le R$ such that $v(g) \in [L, R]$ for any $g \in M$. Then Algorithm 7 is EF1 and requires $O\big((R/L) \cdot n^2 T\big)$ adjustments.*

Before proving Theorem 4.5, we give some useful lemmas. Observe that without the contiguity constraint, leximin allocations are EF1 for additive identical valuations. Since otherwise by selecting agents i, j such that agent i envies agent j even up to one item and moving an item with a positive valuation of agent j to agent i, we reach a contradiction. Nevertheless, it is possible that none of the contiguous leximin allocations is EF1. For instance, suppose that there are 3 agents and 5 items with valuations 1, 3, 1, 1, 1. In this instance, the only contiguous leximin allocation is

---
[3]For completeness, we show how to use Dynamic Programming to efficiently compute the contiguous leximinmin allocation in Appendix C.

$(\{g_1\}, \{g_2\}, \{g_3, g_4, g_5\})$ and agent 1 envies agent 3 even up to one item.

In a contiguous leximin allocation $A = (A_1, \ldots, A_n)$, an essential observation is that the difference between the valuations of two adjacent blocks is upper bounded by R, where R is the maximum valuation of items. As a result, let $i = \arg\min_{j \in N} v(A_j)$ be the agent with the minimum valuation, then $v(A_j) - v(A_i)$ is upper bounded by $|i - j|R$. We show in the following lemma that such inequality always holds throughout Algorithm 6

Lemma 4.6. *For additive identical valuations, suppose that the valuation of each item is at most R. Let $i = \arg\min_{j \in N} v(A_j)$ and $B = v(A_i)$ where A is the initial contiguous leximinmin allocation. Then throughout Algorithm 6, $v(A_j) \le |i - j|R + B$ always holds for any $j \in N$.*

PROOF. Due to symmetry, we assume that $i = 1$ and we will prove that

$$v(A_j) \le (j-1)R + B \qquad (3)$$

always holds for any $j \in N$. Note that by Lemma 4.4, A_i does not change throughout the algorithm. Thus (3) always holds for $j = 1$. Now supposing for induction that (3) always holds for $j = k-1$ where $k > 1$, we will show that (3) holds for $j = k$ all the time. Suppose for contradiction that $v(A_k) > (k-1)R + B$ at some time. During iteration k, we will keep moving the leftmost item in A_k to A_{k-1} until agent i does not envy agent k up to one item. Note that agent i does not envy agent k up to one item only if $v(A_k) \le v(A_i) + R = B + R$, since otherwise for any $g \in A_k$, $v(A_k - g) \ge v(A_k) - R > v(A_i)$. Thus the total valuation of the items being moved from A_k to A_{k-1} during iteration k is at least

$$v(A_k) - (v(A_i) + R) > (k-1)R + B - (B + R) = (k-2)R,$$

which, combining the fact that $v(A_{k-1}) \ge B$, contradicts the induction assumption that (3) always holds for $j = k-1$. □

The following lemma describes the monotonicity of the separating points between two adjacent blocks in the initial contiguous leximinmin allocations in Algorithm 6. This is essentially why we choose leximinmin allocations as initial allocations. Recall that $P_i(A)$ is the index of the last item in bundle A_i.

Lemma 4.7. *For additive identical valuations, let X^t be the initial contiguous leximinmin allocation fixed by Algorithm 6 in round t. Then for every $t < T$ and $j \in N$, we have $P_j(X^t) \le P_j(X^{t+1})$.*

Now it is sufficient to prove Theorem 4.5.

PROOF OF THEOREM 4.5. Lemma 4.4 implies that Algorithm 7 is EF1. It remains to prove the required number of adjustments. We will use Lemma 4.6 to show that for any $t \in [T]$, the number of adjustments required to transform X^t to A^t is upper bounded by $O(n^2 R/L)$. Combining the monotonicity of $P_j(X^t)$ given by Lemma 4.7, we can establish the desired upper bound by triangle inequality.

Fix round t and let $i = \arg\min_{j \in N} v(X_j^t)$. By Lemma 4.6, we know that $v(A_j) \le v(A_i) + |i - j|R$ throughout Algorithm 6. Since by Lemma 4.4, A_i never changes and the valuation of each item is at least L, the number of items being moved from A_j toward A_i during the iteration j is at most $(v(A_j) - v(A_i))/L \le |i-j|R/L$. Note that for

two contiguous allocations A' and A'', the number of adjustments that are required to transform A' to A'' is $\sum_{j=1}^{n-1} |P_j(A') - P_j(A'')|$. Thus the number of adjustments required by Algorithm 6 to transform the initial contiguous leximinmin allocation X^t to the final EF1 allocation A^t is

$$\sum_{j=1}^{n-1} |P_j(X^t) - P_j(A^t)| \leq \sum_{j \neq i} |i - j| R/L \leq O(n^2 R/L).$$

Therefore, the number of adjustments required by Algorithm 7 is

$$\sum_{t=1}^{T-1} \sum_{j=1}^{n-1} |P_j(A^t) - P_j(A^{t+1})|$$

$$\leq \sum_{t=1}^{T-1} \sum_{j=1}^{n-1} \Big(|P_j(A^t) - P_j(X^t)| + |P_j(X^t) - P_j(X^{t+1})| $$

$$+ |P_j(X^{t+1}) - P_j(A^{t+1})| \Big)$$

$$\leq \sum_{t=1}^{T-1} \sum_{j=1}^{n-1} |P_j(X^t) - P_j(X^{t+1})| + O\left((R/L) \cdot n^2 T \right)$$

$$= \sum_{t=1}^{T-1} \sum_{j=1}^{n-1} (P_j(X^{t+1}) - P_j(X^t)) + O\left((R/L) \cdot n^2 T \right)$$

$$= \sum_{j=1}^{n-1} (P_j(X^T) - P_j(X^1)) + O\left((R/L) \cdot n^2 T \right)$$

$$\leq O(nT) + O\left((R/L) \cdot n^2 T \right) = O\left((R/L) \cdot n^2 T \right),$$

where the first equality is due to the monotonicity of $P_j(X^t)$ for any $j \in N$ given by Lemma 4.7. □

4.3 Lower Bounds

We demonstrate that our upper bound to achieve PROPa for identical valuations is asymptotically tight. Moreover, by showing that EF1 implies PROPa, the same lower bound is also applied to EF1.

Theorem 4.8. *For identical valuations, every PROPa algorithm requires at least $\Omega(nT)$ adjustments.*

PROOF. Suppose that there are n agents with identical valuations and each agent has a valuation of 1 for every item. The proportional constraint factor for M_T that should be satisfied by each block is

$$\left\lceil \frac{T}{n} - \frac{n-1}{n} \right\rceil = \left\lfloor \frac{T}{n} \right\rfloor.$$

Denote the number of items that have arrived as T. We first prove that, for a large enough t, g_t must come to every agent's block at least once as T grows larger. In other words, for every agent i, there exists T_i such that in round T_i, g_t belongs to agent i. Then we use this property to show the desired lower bound.

We will prove that, for $t \geq n^2$, g_t must come to every agent's block at least once as T grows larger. Fix $t \geq n^2$. Note that in round $T = t$, g_t must belong to agent n, since agent n's block cannot be empty. Now we show that for any $i < n$, agent i will obtain g_t at some time. Suppose $t = ki - r$, where $0 \leq r < i$ and $k \geq n$. In round $T = kn > t$, each agent should obtain exactly k items. Then g_t belongs to agent i since $k(i-1) < t \leq ki$.

Now it is sufficient to give the lower bound. By the above arguments, for any $t \geq n^2$, $T \geq t$ and $i \geq nt/(T-n)$, we know that g_t must have been obtained by agents $i, i+1, \ldots, n$ previously. Thus the number of adjustments made on g_t throughout T rounds is at least $n - \lceil nt/(T-n) \rceil$. Summing over all $t \geq n^2$, we have

$$\sum_{t=n^2}^{T} \left(n - \left\lceil \frac{nt}{T-n} \right\rceil \right) \geq \sum_{t=n^2}^{T} \left(n - \frac{nt}{T-n} - 1 \right)$$

$$\geq (n-1)(T-n^2) - \sum_{t=1}^{T} \frac{nt}{T-n}$$

$$= (n-1)(T-n^2) - \frac{nT(T+1)}{2(T-n)}$$

$$= \Omega(nT),$$

where the last equality is due to $T \gg n$. Therefore, the number of adjustments required by any PROPa algorithm is at least $\Omega(nT)$. □

It is worth noting that the proportionality condition (2) can be implied by EF1 even for nonidentical valuations. Thus the $O(nT)$ lower bound also works for EF1.

Proposition 4.9. *For agents with nonidentical valuations, if an allocation is EF1, then it also satisfies (2) for all $i \in N$.*

PROOF. Suppose that there are n agents with valuations $v_1, \ldots, v_n$ and allocation $A = (A_1, \ldots, A_n)$ is EF1. Fix $i \in N$. By the definition of EF1, for any $j \in N$, there exists $g_j \in A_j$ such that

$$v_i(A_i) \geq v_i(A_j - g_j).$$

Summing over all $j \neq i$, we obtain

$$(n-1)v_i(A_i) \geq \sum_{j \neq i} v_i(A_j - g_j)$$

Adding $v_i(A_i)$ to both sides,

$$nv_i(A_i) \geq v_i(M) - \sum_{j \neq i} v_i(g_j).$$

Therefore,

$$v_i(A_i) \geq \frac{1}{n} \cdot v_i(M) - \frac{1}{n} \sum_{j \neq i} v_i(g_j)$$

$$\geq \frac{1}{n} \cdot v_i(M) - \frac{n-1}{n} \cdot v_i^{\max}(M)$$

as desired. □

Corollary 4.10. *For identical valuations, every EF1 algorithm requires at least $\Omega(nT)$ adjustments.*

5 CONTIGUOUS SETTING WITH NONIDENTICAL VALUATIONS

In this section, for nonidentical valuations with contiguity requirement, we show that it is hopeless to make any significant improvement to the trivial $O(T^2)$ upper bound for both PROPa and EF1. All the omitted proofs in this section can be found in Appendix E.

Theorem 5.1. *For nonidentical valuations, every PROPa algorithm requires at least $\Omega(T^2/n)$ adjustments.*
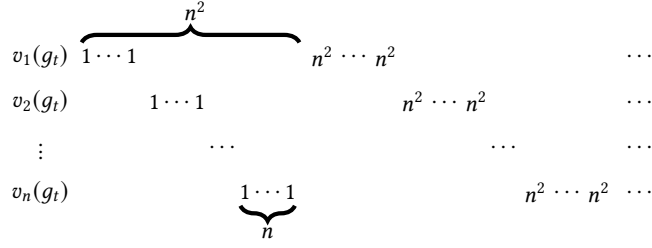
Figure 1: Figure illustrating the instance in the proof of Theorem 5.1. Each period of length n^2 is divided into n blocks of length n, and agent i is only interested in the items in the i-th block in each period c with n^{2c} valuation for each of these items. The valuation 0 is omitted.

Proof. Suppose that there are T items and n agents with valuations $v_1, \ldots, v_n$. For any $t \in [T]$, let

$$v_i(g_t) = \begin{cases} n^{2c}, & cn^2 + (i-1)n < t \le cn^2 + in \text{ for some } c \ge 0, \\ 0, & \text{otherwise,} \end{cases}$$

for all $i \in N$. As illustrated in Figure 1, each period of length n^2 is divided into n blocks of length n, and agent i is only interested in the items in the i-th block in each period c with n^{2c} valuation for each of these items. It suffices to show that, for every $k \ge 2n$, the first $nk - n^2$ items belong to one agent in round nk and another agent in round $nk + n$. Since it follows that the number of adjustments is at least $\sum_{k=2n}^{T/n-1}(nk - n^2) = \Theta(T^2/n)$, where we assume $T \gg n$.

We only give the proof for k such that k is a multiple of n, and the proof can be easily generalized to any $k \ge 2n$. Now we show that for any $c \ge 2$ and $k = cn$, the first $nk - n^2 = (c-1)n^2$ items belong to agent 1 in round nk and belong to agent 2 in round $nk + n$. Note that in round nk, the proportional constraint factor for all agents is

$$\frac{1}{n}\left(n \sum_{j=0}^{c-1} n^{2j}\right) - \frac{n-1}{n} \cdot n^{2(c-1)} = n^{2c-3} + \sum_{j=0}^{c-2} n^{2j},$$

and the total valuation of the first $nk - n^2$ items for each agent is

$$n \sum_{j=0}^{c-2} n^{2j} = \frac{n^{2(c-1)} - 1}{n+1} + \sum_{j=0}^{c-2} n^{2j} < n^{2c-3} + \sum_{j=0}^{c-2} n^{2j}.$$

Define $G_i = \{g_t \mid (c-1)n^2 + (i-1)n < t \le (c-1)n^2 + in\}$ as the set of items that arrive during period c and agent i is interested in. To satisfy the proportional constraint, each agent i should get at least one of the items in G_i. Due to the contiguity requirement, for every $i \in N$, agent i must get the i-th block which should contain at least one item in G_i. As a result, the first $nk - n^2 = (c-1)n^2$ items belong to agent 1. Similarly, we can show that in round $nk + n$, the first $nk - n^2 + n$ items belong to agent 2 and we are done. □

When it comes to EF1, the lower bound is even stronger, indicating that we cannot make any improvement to the trivial $O(T^2)$ upper bound. The proof of Theorem 5.2 is analogous to the proof of Theorem 5.1 except that the length of a period is reduced from n^2 to $O(n)$.

Theorem 5.2. *For nonidentical valuations, every EF1 algorithm requires at least $\Omega(T^2)$ adjustments.*

Note that in the hard instance given to prove Theorem 5.1, the valuations of items are unbounded. Nevertheless, even for 2 agents with binary valuations, i.e. $v_i(g) \in \{0, 1\}$ for all i and g, the $\Omega(T^2)$ lower bound still pertains[4].

Theorem 5.3. *For 2 agents with binary valuations, every EF1 algorithm requires at least $\Omega(T^2)$ adjustments.*

6 CONCLUSION AND FUTURE WORK
In this paper, we study the problem of dynamically allocating indivisible items in a fair manner with few adjustments in both the contiguous and noncontiguous settings. We present some directions for future work.

- Even though we establish tight upper and lower bounds to achieve PROPa in the contiguous setting, there are still large gaps between the upper and lower bounds to achieve EF1 in both settings. The first open problem is to tighten these bounds.
- If the types of items are drawn from certain distributions rather than adversely, can we show a better upper bound in expectation or asymptotically in both settings[5]?
- It would be interesting to investigate other fairness notions like EQ1 [23].
- Another promising direction, like always asked in the offline setting [15], is to achieve fairness and Pareto optimality simultaneously. This has been shown in some other online models [3, 27].

REFERENCES
[1] Hannaneh Akrami, Rojin Rezvan, and Masoud Seddighin. 2022. An EF2X Allocation Protocol for Restricted Additive Valuations. In *IJCAI*. ijcai.org, 17–23.
[2] Martin Aleksandrov and Toby Walsh. 2017. Expected Outcomes and Manipulations in Online Fair Division. In *KI (Lecture Notes in Computer Science, Vol. 10505)*. Springer, 29–43.
[3] Martin Aleksandrov and Toby Walsh. 2019. Strategy-Proofness, Envy-Freeness and Pareto Efficiency in Online Fair Division with Additive Utilities. In *PRICAI (1) (Lecture Notes in Computer Science, Vol. 11670)*. Springer, 527–541.
[4] Martin Aleksandrov and Toby Walsh. 2020. Online Fair Division: A Survey. In *AAAI*. AAAI Press, 13557–13562.
[5] Yushi Bai and Paul Gölz. 2022. Envy-Free and Pareto-Optimal Allocations for Agents with Asymmetric Random Valuations. In *IJCAI*. ijcai.org, 53–59.
[6] Gerdus Benadè, Daniel Halpern, and Alexandros Psomas. [n.d.]. Dynamic Fair Division with Partial Information. ([n.d.]).
[7] Gerdus Benade, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. 2018. How to Make Envy Vanish Over Time. In *EC*. ACM, 593–610.
[8] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. 2022. Almost envy-free allocations with connected bundles. *Games Econ. Behav.* 131 (2022), 197–221.
[9] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. In *IJCAI*. ijcai.org, 135–141.

---
[4]For 2 agents with binary valuations, PROPa is equivalent to EF1.
[5]Similar problems are presented in [26].

[10] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. 2019. Chore division on a graph. *Auton. Agents Multi Agent Syst.* 33, 5 (2019), 540–563.

[11] Johannes Brustle, Jack Dippel, Vishnu V. Narayan, Mashbat Suzuki, and Adrian Vetta. 2020. One Dollar Each Eliminates Envy. In *EC*. ACM, 23–39.

[12] Eric Budish. 2010. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. In *BQGT*. ACM, 74:1.

[13] Eric J. Friedman, Christos-Alexandros Psomas, and Shai Vardi. 2015. Dynamic Fair Division with Minimal Disruptions. In *EC*. ACM, 697–713.

[14] Eric J. Friedman, Christos-Alexandros Psomas, and Shai Vardi. 2017. Controlled Dynamic Fair Division. In *EC*. ACM, 461–478.

[15] Jugal Garg and Aniket Murhekar. 2022. Computing Pareto-Optimal and Almost Envy-Free Allocations of Indivisible Goods. *CoRR* abs/2204.14229 (2022).

[16] Jonathan R. Goldman and Ariel D. Procaccia. 2014. Spliddit: unleashing fair division algorithms. *SIGecom Exch.* 13, 2 (2014), 41–46.

[17] Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. 2019. Achieving a Fairer Future by Changing the Past. In *IJCAI*. ijcai.org, 343–349.

[18] Felix Höhne and Rob van Stee. 2021. Allocating contiguous blocks of indivisible chores fairly. *Inf. Comput.* 281 (2021), 104739.

[19] Ayumi Igarashi. 2022. How to cut a discrete cake fairly. arXiv:2209.01348 [cs.GT]

[20] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *EC*. ACM, 125–131.

[21] Pasin Manurangsi and Warut Suksompong. 2020. When Do Envy-Free Allocations Exist? *SIAM J. Discret. Math.* 34, 3 (2020), 1505–1521.

[22] Pasin Manurangsi and Warut Suksompong. 2021. Closing Gaps in Asymptotic Fair Division. *SIAM J. Discret. Math.* 35, 2 (2021), 668–706.

[23] Neeldhara Misra, Chinmay Sonar, P. R. Vaidyanathan, and Rohit Vaish. 2021. Equitable Division of a Path. *CoRR* abs/2101.09794 (2021).

[24] Hoon Oh, Ariel D. Procaccia, and Warut Suksompong. 2021. Fairly Allocating Many Goods with Few Queries. *SIAM J. Discret. Math.* 35, 2 (2021), 788–813.

[25] Roman Seidl. 2018. Handbook of Computational Social Choice *by Brandt Felix, Vincent Conitzer, Ulle Endriss, Jerome Lang, Ariel Procaccia. J. Artif. Soc. Soc. Simul.* 21, 2 (2018).

[26] Warut Suksompong. 2019. Fairly allocating contiguous blocks of indivisible items. *Discret. Appl. Math.* 260 (2019), 227–236.

[27] David Zeng and Alexandros Psomas. 2020. Fairness-Efficiency Tradeoffs in Dynamic Fair Division. In *EC*. ACM, 911–912.

A   TWO AGENTS AND MIXED MANNA

In this section, we allow valuations to be negative, which is often referred to as *mixed manna*. For two agents, we show that $O(T)$ adjustments are sufficient to achieve EF1 in Algorithm 8. The algorithm is the extension of the Envy Balancing Algorithm for two agents with nonnegative valuations given by [17], which assumes that we know the information of all items in advance and requires no adjustments.

The idea of Algorithm 8 is based on the envy-cycle elimination algorithm, which is proposed by [20] to efficiently compute an EF1 allocation. Recall that an allocation A is *envy-free (EF)* if $v_i(A_i) \geq v_i(A_j)$ for any $i, j \in N$. In Algorithm 8, we maintain two disjoint allocations G and C satisfying that G is EF and C is EF1. Since the combination of an EF allocation and an EF1 allocation is also EF1, the returned allocation, which is the combination of G and C, is also EF1. To this end, we only assign the new item to an agent in C while maintaining the EF1 property of C (Line 3–Line 6). Whenever both agents envy each other in C, we swap their bundles in C to make C an EF allocation (Line 7–Line 8). Once C becomes EF, we merge C into G (Line 9–Line 11).

Algorithm 8 Envy Balancing Algorithm for Mixed Manna

Require: v_1, v_2
1: $G \leftarrow (\emptyset, \emptyset), C \leftarrow (\emptyset, \emptyset)$
2: **for** $t = 1, \ldots, T$ **do**
3:     **if** (a_1 is unenvied in $C \wedge v_1(g_t) > 0) \vee (v_1(g_t) > 0 \wedge v_2(g_t) \leq 0) \vee (a_1$ does not envy a_2 in $C \wedge v_1(g_t) \leq 0 \wedge v_2(g_t) \leq 0)$ **then**
4:         $C \leftarrow (C_1 \cup \{g_t\}, C_2)$
5:     **else**
6:         $C \leftarrow (C_1, C_2 \cup \{g_t\})$
7:     **if** a_1 and a_2 envy each other in C **then**
8:         $C \leftarrow (C_2, C_1)$
9:     **if** C is envy-free **then**
10:         $G \leftarrow (G_1 \cup C_1, G_2 \cup C_2)$
11:         $C \leftarrow (\emptyset, \emptyset)$
12:     $A^t = (C_1 \cup G_1, C_2 \cup G_2)$
13: **return** $[A^1, A^2, \ldots, A^T]$

Theorem A.1. *Algorithm 8 is an EF1 algorithm for two agents and mixed manna that requires $O(T)$ adjustments.*

PROOF. We first prove that every returned allocation is EF1. Since the combination of an EF allocation and an EF1 allocation is EF1, it suffices to prove that G is always EF and C is always EF1. Notice that we merge C into G only if C is EF, and the combination of two EF allocations is EF. Thus G is always EF.

Now we prove that C is always EF1. We only need to show that, if C is EF1 at the beginning of the current iteration, it is also EF1 after the iteration. If C is EF1, at least one of the agents is unenvied in C since otherwise the bundles in C would have been swapped in the previous iteration, which makes C an EF allocation. Thus there is also at least one agent who does not envy the other agent in C. After g_t arrives, according to Line 3, if it is a good for an agent and a chore for the other agent, i.e., $v_1(g_t) > 0 \wedge v_2(g_t) \leq 0$ or $v_1(g_t) \leq 0 \wedge v_2(g_t) > 0$, it will be assigned to the agent who views it as a good, which ensures that both $v_1(C_1) - v_1(C_2)$ and $v_2(C_2) - v_2(C_1)$ don't decrease after the assignment of g_t. In this case, the EF1 property of C is maintained. Next, if the new item is a good for both agents, i.e., $v_1(g_t) > 0 \wedge v_2(g_t) > 0$, the agent who it is assigned to is unenvied by the other agent in C. Thus by definition, C is EF1 after the assignment of g_t. Finally, if the new item is a chore for both agents, i.e., $v_1(g_t) \leq 0 \wedge v_2(g_t) \leq 0$, the agent who it is assigned to does not envy the other agent in C. By definition, C is EF1 after the assignment of g_t.

It remains to prove the number of adjustments. Note that adjustments happen only when both agents envy each other in C, and, in this case, their bundles in C are swapped, which contributes $|C_1 \cup C_2|$ adjustments. After the swap, C becomes EF and is merged into G. Since no adjustments are made on G, each item is adjusted at most once. Therefore, the total number of adjustments is $O(T)$.   □

B   THE INCENTIVE ISSUE

It is natural to ask about the incentive issue. That is, apart from the objective of fairness, we also hope that each agent is better off after the adjustments to make them happy with the new allocation. It is not difficult to see that all the algorithms given in this paper in the noncontiguous setting satisfy the incentive property. Furthermore, we prove in Lemma B.1 that if we re-run the round-robin algorithm with a deterministic tie-breaking rule upon the arrival of a new item, the valuation of each agent will not decrease in the new allocation. This immediately leads to the incentive property of the algorithms given by [17] which are based on the round-robin algorithm.

Now we prove the incentive property of the round-robin algorithm. Recall that in the round-robin algorithm, agents pick their favorite items alternately in each round. Suppose that there are n agents and m items $g_1, \ldots, g_m$. In the outcome of the round-robin algorithm, we use C_i^j to denote the index of the item obtained by agent i in round j. In particular, $C_i^j = 0$ if agent i does not obtain any item in round j, and

let $v_i(0) = 0$ for all $i \in N$. Note that there are $\lceil m/n \rceil$ rounds in total. When a new item g_{m+1} arrives, we run the round-robin algorithm again to allocate all the $m + 1$ items. Let $\widetilde{C}_i^j$ denote the new outcome analogously.

Lemma B.1. *For every* $1 \leq j \leq \lceil (m + 1)/n \rceil$ *and* $i \in N$, *we have* $v_i(C_i^j) \leq v_i(\widetilde{C}_i^j)$.

PROOF. Without loss of generality, we assume that the agent with a smaller index always picks an item before the agent with a larger index in each round. Define

$$M_i^j = \{g_1, \ldots, g_m\} \setminus \{C_k^\ell \mid \text{either } \ell < j \text{ or } \ell = j, k < i\}$$

as the set of remaining items before agent i pick an item in round j, and

$$\widetilde{M}_i^j = \{g_1, \ldots, g_{m+1}\} \setminus \{\widetilde{C}_k^\ell \mid \text{either } \ell < j \text{ or } \ell = j, k < i\}$$

analogously. Since

$$C_i^j = \arg\max_{g \in M_i^j} v_i(g), \quad \widetilde{C}_i^j = \arg\max_{g \in \widetilde{M}_i^j} v_i(g),$$

it suffices to show that $M_i^j \subseteq \widetilde{M}_i^j$, which trivially holds when $j = 0$.

Assume that $i < n$ and the case of $i = n$ is treated similarly. Suppose for induction that $M_i^j \subseteq \widetilde{M}_i^j$ is satisfied. We show that $M_{i+1}^j \subseteq \widetilde{M}_{i+1}^j$. Since $|M_i^j| + 1 = |\widetilde{M}_i^j|$, let g be the only item in $\widetilde{M}_i^j \setminus M_i^j$. On one hand, if $\widetilde{C}_i^j = g$, then $\widetilde{M}_{i+1}^j = M_{i+1}^j \cup \{C_i^j\}$. On the other hand, if $\widetilde{C}_i^j \neq g$, we have $\widetilde{C}_i^j = C_i^j$ and thus $\widetilde{M}_{i+1}^j = M_{i+1}^j \cup \{g\}$. Therefore, $M_{i+1}^j \subseteq \widetilde{M}_{i+1}^j$ holds.   □

By contrast, with the contiguity constraint, suppose that the order of agents stays the same throughout the entire period, i.e., agent i obtains the i-th block, which is the default assumption for identical valuations in this work. In this case, the incentive property is not compatible with EF1, even for two agents with additive identical valuations. For instance, there are three items with valuations of 1, 3 and 2, respectively. In round 2, we have $A^2 = (\{g_1\}, \{g_2\})$. However, in round 3, the only contiguous allocation that satisfies EF1 is $A^3 = (\{g_1, g_2\}, \{g_3\})$, in which the valuation of agent 2 is less than the previous round.

C EFFICIENT COMPUTATION OF CONTIGUOUS LEXIMINMIN ALLOCATIONS FOR IDENTICAL VALUATIONS

In this section, we discuss how to efficiently compute a contiguous leximinmin allocation. Recall that the contiguous leximinmin allocation is the contiguous leximin allocation A with lexicographically minimum $P(A)$ among all leximin allocations, where $P(A) = (\ell_0, \cdots, \ell_n)$ such that $\ell_0 = 0$ and ℓ_i is the index of the last item in A_i. Given an instance with n agents and m items, we will apply Dynamic Programming to compute the contiguous leximinmin allocation in $O(n^2 m^2)$ time.

Suppose that there are n agents with general identical valuations $v_1, \ldots, v_n$ and m items $g_1, \ldots, g_m$ arranged on a line. Recall that $M_{l,r} = \{g_{l+1}, \ldots, g_r\}$ is the set of items whose indexes are in range $[l+1, r]$. In particular, let $M_{l,r} = \emptyset$ if $l \geq r$. For an allocation $A = (A_1, \ldots, A_k)$, define $Q(A) = (v(A_{\sigma(1)}), v(A_{\sigma(2)}), \ldots, v(A_{\sigma(k)}))$ as the tuple obtained by sorting the valuations of all agents in nondecreasing order, where σ is a permutation with length k satisfying $v(A_{\sigma(i)}) \leq v(A_{\sigma(i+1)})$ for any $i < k$. Given allocations A' and A'', we say that A' is better than A'' if either $Q(A')$ is lexicographically larger than $Q(A'')$ or $Q(A') = Q(A'')$ and $P(A')$ is lexicographically smaller than $P(A'')$. By definition, the contiguous leximinmin allocation is the best allocation among all contiguous allocations.

For any $i \in [n]$ and $j \in [m] \cup \{0\}$, let $A[i, j]$ be the leximinmin allocation for M_j with i agents. Observe that if $A = (A_1, \ldots, A_i)$ is a leximinmin allocation, for any $k \in [i]$, $A' = (A_1, \ldots, A_k)$ is the leximinmin allocation for the first $|A_1| + \cdots + |A_k|$ items with k agents. Thus if the last block of $A[i, j]$ is $M_{k,j}$ for some $0 \leq k \leq j$, then $A[i, j] = A[i - 1, k] \circ M_{k,j}$, where we use $A \circ x$ to denote the tuple obtained by appending x to the end of tuple A. As a result, the recurrence relation of $A[i, j]$ is

$$A[i, j] = \begin{cases} (M_j), & i = 1, \\ \text{the best allocation in } \{A[i - 1, k] \circ M_{k,j} \mid 0 \leq k \leq j\}, & \text{otherwise.} \end{cases}$$

In the implementation, we only need to maintain $P(A[i, j])$ and $Q(A[i, j])$, which are the necessary information to compare two allocations, rather than recording the exact allocation. We use $f[i, j]$ to denote $P(A[i, j])$ and $h[i, j]$ to denote $Q(A[i, j])$. By the recurrence relation of $A[i, j]$, we have $f[i, j] = f[i - 1, k] \circ j$, where the last block of $A[i, j]$ is $M_{k,j}$. Analogously, $h[i, j]$ can be derived by inserting $v(M_{k,j})$ to the appropriate position in $h[i - 1, k]$ to ensure the monotonicity of the elements in $h[i, j]$.

Now we analyze the time complexity of the Dynamic Programming approach that we discussed above. For each pair $(i, j) \in [n] \times ([m] \cup \{0\})$, we need to enumerate all allocations in $\{A[i - 1, k] \circ M_{k,j} \mid 0 \leq k \leq j\}$ and make comparisons among them to compute $A[i, j]$. Thus the total number of comparisons to be made is $O(nm^2)$. Moreover, each comparison can be finished in $O(n)$ time. Therefore, the running time of the algorithm is $O(n^2 m^2)$.
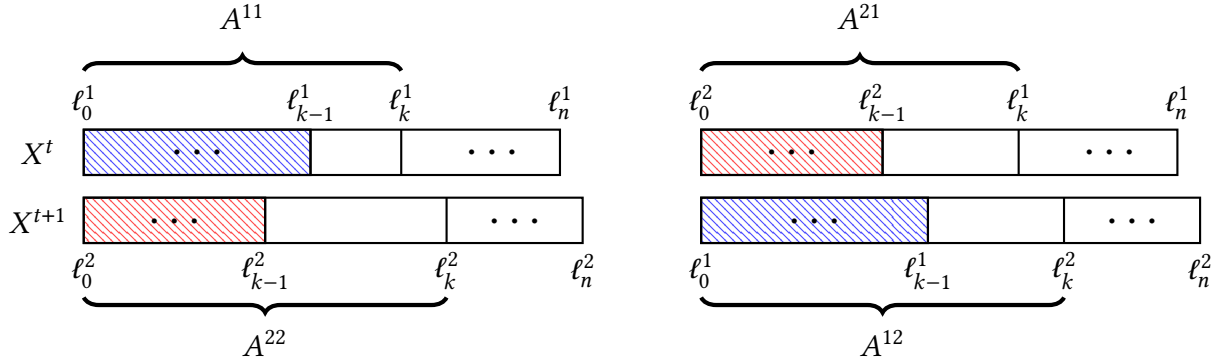
Figure 2: Figure illustrating the proof of Lemma 4.7

D MISSING PROOFS OF SECTION 4

Theorem 4.2. *Algorithm 4 is a PROPa algorithm for identical valuations that requires $O(nT)$ adjustments.*

PROOF. Let $B_t^{\max} = \max_{i \in [t]} B_i$. For a contiguous allocation $A = (A_1, \ldots, A_n)$, define $\ell(A_i)$ as the index of the last item in A_i. In particular, let $\ell(A_0) = 0$. We abuse the notation and let $\ell(A) = (\ell(A_1), \ldots, \ell(A_{n-1}))$. For two contiguous allocations A' and A'', we say that $\ell(A') \prec \ell(A'')$ if $\ell(A')$ is lexicographically smaller than $\ell(A'')$. We say that $\ell(A') \preceq \ell(A'')$ if $\ell(A') \prec \ell(A'')$ or $\ell(A') = \ell(A'')$. For any $t \in [T]$, define X^t as the allocation with lexicographically minimum $\ell(A)$ among all contiguous allocations $A = (A_1, \ldots, A_n)$ for M_t satisfying constraint factor $B_t^{\max}$, i.e., $\min_{i \in N} v(A_i) \geq B_t^{\max}$. We first show that X^t is well-defined for all $t \in [T]$, and $\ell(X_i^t) \preceq \ell(X_i^{t+1})$ for all $t < T$ and $i < n$. Then we prove that $A^t = X^t$ for all $t \in [T]$.

We first prove that X^t is well-defined for all $t \in [T]$, which is equivalent to showing the existence of the contiguous allocation for M_t that satisfies constraint factor $B_t^{\max}$. The existence of X^t trivially holds for $t = 0$. We assume for induction that X^t exists and show that X^{t+1} also exists. If $B_{t+1}^{\max} = B_t^{\max}$, it is easy to show that by adding g_{t+1} to the last block of X^t, the corresponding allocation for M_{t+1} satisfies constraint factor $B_t^{\max} = B_{t+1}^{\max}$ and we are done. If $B_{t+1}^{\max} > B_t^{\max}$, since in this case we must have $B_{t+1}^{\max} = B_{t+1}$, by Lemma 4.1, the allocation that satisfies constraint factor B_{t+1} exists. Therefore, X^t is well-defined for all $t \in [T]$.

Next, we show that for all $t < T$ and $i < n$, $\ell(X_i^t) \preceq \ell(X_i^{t+1})$. Fix $t < T$. Suppose for contradiction that there exists $i < n$ such that $\ell(X_i^t) > \ell(X_i^{t+1})$ and $\ell(X_j^t) \leq \ell(X_j^{t+1})$ for all $j < i$. By the definition of $\ell(X_i^t)$, we have $X_i^{t+1} \subsetneq X_i^t$. Since $B_1^{\max}, \ldots, B_T^{\max}$ are nondecreasing, we have $v(X_i^{t+1}) \geq B^{t+1} \geq B^t$ due to the definition of X_i^{t+1}. In allocation X^t, by moving the items in $X_i^t \setminus X_i^{t+1}$ from X_i^t to X_{i+1}^t, we obtain a new allocation X for M_t with $\ell(X) \prec \ell(X^t)$ that satisfies the constraint factor $B_t^{\max}$, which contradicts the lexicographic minimality of X^t. Thus $\ell(X_i^t) \preceq \ell(X_i^{t+1})$ for all $t < T$ and $i < n$.

Now it suffices to prove that for all $t \in [T]$, $A^t = X^t$, which satisfies constraint factor $B_t^{\max} \geq B_t$ by definition. First, $A^t = X^t$ trivially holds for $t = 0$. Supposing for induction that $A^{t-1} = X^{t-1}$ where $t < T$, we will show that $A^t = X^t$.

If $B_t^{\max} = B_{t-1}^{\max}$, then $p_1, \ldots, p_{n-1}$ will not change in round t since $A^{t-1} = X^{t-1}$ satisfies the constraint factor $B_{t-1}^{\max}$, which indicates that $\ell(A^t) = \ell(A^{t-1})$. By the lexicographic minimality of X^t, we have $\ell(X^t) \preceq \ell(A^t)$. Thus,

$$\ell(X^{t-1}) \preceq \ell(X^t) \preceq \ell(A^t) = \ell(A^{t-1}) = \ell(X^{t-1}),$$

by which we conclude that $A^t = X^t$.

Suppose $B_t^{\max} > B_{t-1}^{\max}$, which implies that $B_t = B_t^{\max}$. At the beginning of round t, we have

$$p_i = \ell(A_i^{t-1}) = \ell(X_i^{t-1}) \leq \ell(X_i^t),$$

for all $i < n$. It is trivial that $\ell(A_0^t) = \ell(X_0^t)$. Assuming for induction that $\ell(A_{i-1}^t) = \ell(X_{i-1}^t)$ where $i < n$, we will show that $\ell(A_i^t) = \ell(X_i^t)$. On one hand, $\ell(A_i^t) \leq \ell(X_i^t)$ must hold, since $v(X_i^t) \geq B_t$ and the condition of Line 5 will be violated when $p_i = \ell(X_i^t)$. On the other hand, $\ell(A_i^t) \geq \ell(X_i^t)$ must hold, since otherwise we have $A_i^t \subsetneq X_i^t$ and $v(A_i^t) \geq B_t$, and by moving the items in $X_i^t \setminus A_i^t$ from X_i^t to X_{i+1}^t, we obtain a new allocation X for M_t with $\ell(X) \prec \ell(X^t)$ that satisfies constraint factor $B_t = B_t^{\max}$, which contradicts the lexicographic minimality of X^t. Therefore, $A^t = X^t$ for all $t \in [T]$.

Finally, we prove the number of adjustments required by Algorithm 4. For an item g_t that has arrived, the agent i that g_t belongs to satisfies $p_{i-1} < t \leq p_i$. Since p_i is nondecreasing for each $i \in N$, the number of adjustments made on any item g_t is at most $n - 1$. As a result, the total number of adjustments is at most $(n-1)T = O(nT)$. □

Lemma 4.6. *For additive identical valuations, suppose that the valuation of each item is at most R. Let $i = \arg\min_{j \in N} v(A_j)$ and $B = v(A_i)$ where A is the initial contiguous leximinmin allocation. Then throughout Algorithm 6, $v(A_j) \leq |i - j|R + B$ always holds for any $j \in N$.*

PROOF. Let's first specify some conventions. Since for any feasible tuple $O = (\ell_0, \ldots, \ell_n)$, there exists the only allocation A satisfying $P(A) = O$, from now on we will use a tuple to refer to the corresponding allocation. For an allocation $A = (A_1, \ldots, A_k)$ and $O = P(A)$, define $Q(O) = (v(A_{\sigma(1)}), v(A_{\sigma(2)}), \ldots, v(A_{\sigma(k)}))$ as the tuple obtained by sorting the valuations of all agents in nondecreasing order, where σ is a permutation with length k satisfying $v(A_{\sigma(i)}) \leq v(A_{\sigma(i+1)})$ for any $i < k$. For two tuples O_1 and O_2, we say $O_1 \leq O_2$ if O_1 is lexicographically not larger than O_2. Without loss of generality, we assume that $v(g) > 0$ for all $g \in M$.

Fix round t. We will prove that $P_j(X^t) \leq P_j(X^{t+1})$ for every $j \in N$. Let $P_j(X^t) = (\ell_0^1, \ell_1^1, \ldots, \ell_n^1)$ and $P_j(X^{t+1}) = (\ell_0^2, \ell_1^2, \ldots, \ell_n^2)$, as illustrated in Figure 2. Since both X^t and X^{t+1} are leximinmin, for any $j \in [n]$ and $k \in \{1, 2\}$, $(\ell_0^k, \ldots, \ell_j^k)$ must be a leximinmin allocation for $M_{\ell_j^k}$ with j agents. Thus if $\ell_j^1 = \ell_j^2$ for some $j \in N$, then $\ell_k^1 = \ell_k^2$ for all $k \leq j$.

Suppose for contradiction that there exists $j \in N$ such that $\ell_j^1 > \ell_j^2$. Let k be the first index larger than j such that $\ell_k^1 < \ell_k^2$. Since $\ell_n^1 = t < t + 1 = \ell_n^2$, such k must exist and, in the same time, satisfies that $\ell_{k-1}^1 > \ell_{k-1}^2$. For $p, q \in \{1, 2\}$, define A^{pq} as the allocation for $M_{\ell_k^q}$ such that $P(A^{pq}) = (\ell_0^p, \ell_1^p, \ldots, \ell_{k-1}^p, \ell_k^q)$. Since both A^{11} and A^{22} are leximinmin, we have

$$Q(A^{21}) \leq Q(A^{11}),$$

and

$$Q(A^{12}) \leq Q(A^{22}). \tag{4}$$

It suffices to show that $Q(A^{11}) \leq Q(A^{21})$ and $Q(A^{22}) \leq Q(A^{12})$. Since in this case, if

$$(\ell_0^2, \ell_1^2, \ldots, \ell_{k-1}^2) \leq (\ell_0^1, \ell_1^1, \ldots, \ell_{k-1}^1),$$

we can replace the first k elements of $P(X^t) = (\ell_0^1, \ldots, \ell_n^1)$ with $\ell_0^2, \ldots, \ell_{k-1}^2$ to obtain another leximin allocation A with $P(A) \leq P(X^t)$, which contradicts the fact that X^t is leximinmin. By contrast, if

$$(\ell_0^1, \ell_1^1, \ldots, \ell_{k-1}^1) \leq (\ell_0^2, \ell_1^2, \ldots, \ell_{k-1}^2),$$

we can replace the first k elements of $P(X^{t+1}) = (\ell_0^2, \ldots, \ell_n^2)$ with $\ell_0^1, \ldots, \ell_{k-1}^1$ to obtain another leximin allocation A with $P(A) \leq P(X^{t+1})$, which contradicts the fact that X^{t+1} is leximinmin.

Due to the symmetry, we only prove $Q(A^{11}) \leq Q(A^{21})$. Since $\ell_k^1 < \ell_k^2$ and $\ell_{k-1}^1 > \ell_{k-1}^2$, we have

$$v(A_k^{11}) < v(A_k^{12}) < v(A_k^{22}), \tag{5}$$

and thus

$$v(A_k^{11}) = v(A_k^{12}) - v(M_{\ell_k^2} \setminus M_{\ell_k^1}) < v(A_k^{22}) - v(M_{\ell_k^2} \setminus M_{\ell_k^1}) = v(A_k^{21}). \tag{6}$$

Note that all but the last blocks in A^{12} are the same as those in A^{11}, and all but the last blocks in A^{22} are the same as those in A^{21}. Recall that when deciding the lexicographic order of two tuples, we first compare the first elements of two tuples; if they are equal, then compare the second ones, and so on. If (4) is decided only by the blocks in A^{12} and A^{22} with valuations smaller than $v(A_k^{11})$, then $Q(A^{11}) \leq Q(A^{21})$ holds because the blocks with valuations smaller than $v(A_k^{11})$ in A^{11} and A^{21} are the same with those in A^{12} and A^{22}. On the other hand, if (4) is decided by the blocks with valuations at least $v(A_k^{11})$, (4) implies that the number of blocks with valuations $v(A_k^{11})$ in A^{12} is not smaller than the number of blocks with valuations $v(A_k^{11})$ in A^{22}. By (5) and (6), the number of blocks with valuations $v(A_k^{11})$ in A^{11} is strictly larger than the number of blocks with valuations $v(A_k^{11})$ in A^{21}. Therefore, $Q(A^{11}) \leq Q(A^{21})$ is established. □
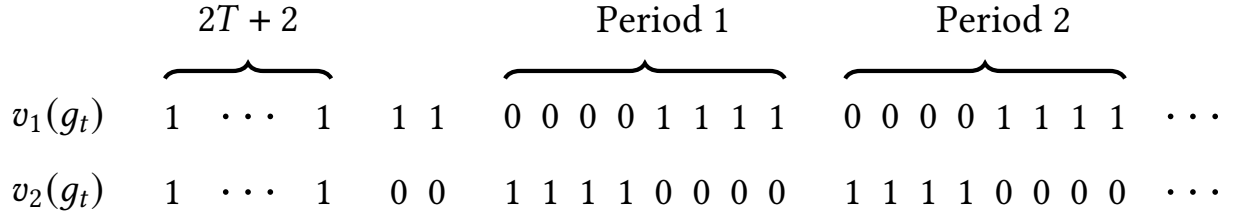
E   MISSING PROOFS OF SECTION 5

Theorem 5.2. *For nonidentical valuations, every EF1 algorithm requires at least $\Omega(T^2)$ adjustments.*

PROOF. Suppose that there are T items and n agents with valuation $v_1, \ldots, v_n$. For any $t \in [T]$, let

$$v_i(g_t) = \begin{cases} n^{2c}, & 3cn + 3(i-1) < t \leq 3cn + 3i \text{ for some } c \geq 0, \\ 0, & \text{otherwise,} \end{cases}$$

for all $i \in N$. That is, each period of length $3n$ is divided into n blocks of length 3, and agent i is only interested in the items in the i-th block in each period c with n^{2c} valuation for each of these items. It suffices to show that, for every $k \geq 2n$, the first $3k - 3n$ items belong to one agent in round $3k$ and another agent in round $3k + 3$. Since it follows that the number of adjustments required is at least $\sum_{k=2n}^{T/3-1} 3(k - n) = \Theta(T^2)$, where we assume $T \gg n$.

We only give the proof for k such that k is a multiple of n, and the proof can be easily generalized to any $k \geq 2n$. Now we prove that, for any $c \geq 2$ and $k = cn$, the first $3k - 3n = (c - 1) \cdot 3n$ items belong to agent 1 in round $3k$ and belong to agent 2 in round $3k + 3$. Define

$$\overbrace{\qquad 2T+2 \qquad} \qquad\qquad \overbrace{\qquad \text{Period 1} \qquad} \qquad \overbrace{\qquad \text{Period 2} \qquad}$$

$$v_1(g_t) \quad 1 \ \cdots \ 1 \quad 1\ 1 \quad 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \quad 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \ \cdots$$

$$v_2(g_t) \quad 1 \ \cdots \ 1 \quad 0\ 0 \quad 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0 \quad 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0 \ \cdots$$

Figure 3: Figure illustrating the instance in the proof of Theorem 5.3

$G_i = \{g_t \mid (c-1)\cdot 3n + 3(i-1) < t \le (c-1)\cdot 3n + 3i\}$ as the set of items that arrive during period c and agent i is interested in. In round $3k$, if agent i gets none of the items in G_i, there must exist another agent j that obtains at least two of them due to the contiguity requirement. Furthermore, the total valuation of the first $3k - 3n = (c-1)\cdot 3n$ items for agent i is

$$3\sum_{j=0}^{c-2} n^{2j} = 3\cdot \frac{n^{2(c-1)}-1}{n^2-1} < n^{2(c-1)} = v_i(g), \quad \forall g\in G_i.$$

Thus if agent i gets none of the items in G_i, he must envy agent j up to one item. As a result, to satisfy EF1, each agent i should get at least one of the items in G_i. Due to the contiguity requirement, for every $i \in N$, agent i must get the i-th block which contains at least one item in G_i. In this case, the first $3k - 3n = (c-1)\cdot 3n$ items belong to agent 1. Similarly, we can show that in round $3k+3$, the first $3k - 3n + 3$ items belong to agent 2 and we are done. □

Theorem 5.3. *For 2 agents with binary valuations, every EF1 algorithm requires at least $\Omega(T^2)$ adjustments.*

PROOF. We describe an instance with $4T + 4$ items where $T > 0$ is a multiple of 4. The instance is illustrated in Figure 3. We say that an item g is type 0 if $v_1(g) = v_2(g) = 1$, type 1 if $v_1(g) = 1, v_2(g) = 0$, and type 2 if $v_1(g) = 0, v_2(g) = 1$. The first $2T + 2$ items are type 0. The following 2 items are type 1. The remaining $2T$ items are divided into periods with a length of 8. In each period, the first 4 items are type 2 and the last 4 items are type 1. For any $0 \le k < T/4$, we show that the first block must belong to agent 2 in round $2T + 8k + 4$ and must belong to agent 1 in round $2T + 8k + 8$.

Given an allocation, we say that an item is non-wasteful if it has a valuation of 1 for the agent that obtains it. In round $2T + 8k + 4$, the numbers of non-wasteful items required by agent 1 and agent 2 are $T + 2k + 2$ and $T + 2k + 1$, respectively. Thus an EF1 allocation should contain at least $2T + 4k + 3$ non-wasteful items. Suppose for contradiction that the second block belongs to agent 2. Denote G as the set of the last $8k + 2$ items. Since $v_2(G) = 4k < T + 2k + 1$, all items in G must belong to agent 2, and thus there are $|G| - v_2(G) = 4k + 2$ wasteful items in G. As a result, the number of non-wasteful items is at most

$$(2T + 8k + 4) - (4k + 2) = 4T + 4k + 2 < 2T + 4k + 3,$$

which leads to a contradiction. Therefore, the first block must belong to agent 2 in round $2T + 8k + 4$.

The proof for round $2T + 8k + 8$ is analogous. In round $2T + 8k + 8$, the numbers of non-wasteful items required by agent 1 and agent 2 are $T + 2k + 2$ and $T + 2k + 3$, respectively. Thus an EF1 allocation should contain at least $2T + 4k + 5$ non-wasteful items. Suppose for contradiction that the second block belongs to agent 1. Denote G as the set of the last $8k + 6$ items. Since $v_1(G) = 4k + 2 < T + 2k + 3$, all items in G must belong to agent 1, and thus there are $|G| - v_1(G) = 4k + 4$ wasteful items in G. As a result, the number of non-wasteful items is at most

$$(2T + 8k + 8) - (4k + 4) = 2T + 4k + 4 < 2T + 4k + 5,$$

which leads to a contradiction. Therefore, the first block must belong to agent 1 in round $2T + 8k + 8$.

Finally, we prove the lower bound of the number of adjustments. Start from round $2T + 4$ and the agent who obtains the first block alternates after every 4 rounds. Since the first block must contain the first T items, the number of adjustments is at least $T^2/2 = \Theta(T^2)$. □

Unfortunately, the instance given above only provides a $\Omega(nT)$ lower bound after being generalized to any number of agents, which is directly implied by Theorem 4.8.