Dynamic Fair Division Problem with General Valuations

Bo Li[1], Wenyang Li[2], Yingkai Li[3]
[1] Stony Brook University
[2] Melbourne University
[3] Northwestern University
boli2@cs.stonybrook.edu, wenl6@student.unimelb.edu.au, yingkai.li@u.northwestern.edu

Abstract

In this paper, we focus on how to dynamically allocate a divisible resource fairly among n players who arrive and depart over time. The players may have general heterogeneous valuations over the resource. It is known that exact envy-free and proportional allocations may not exist in the dynamic setting [Walsh, 2011]. Thus, we will study to what extent we can guarantee the fairness in the dynamic setting. We first design two algorithms which are $O(\log n)$-proportional and $O(n)$-envy-free for the setting with general valuations. Then by constructing the adversary instances such that all dynamic algorithms must be at least $\Omega(1)$-proportional and $\Omega(\frac{n}{\log n})$-envy-free, we show that the bounds are tight up to a logarithmic factor. Moreover, we introduce a setting where the players' valuations are uniform on the resource but with different demands, which generalizes the setting of [Friedman *et al.*, 2015]. We prove an $O(\log n)$ upper bound and a tight lower bound for this case.

1 Introduction

Initiated by the work of [Steinhaus, 1948], the fair division problem has been widely studied in the literature of economics, mathematics and computer science [Dubins and Spanier, 1961; Stromquist, 1980; Alon, 1987; Brams and Taylor, 1995; 1996; Robertson and Webb, 1998; Aziz and Mackenzie, 2016]. It mainly considers the problem of fairly allocating a divisible resource among a group of players who have different preferences over the resource.

To capture fairness, many solution concepts have been proposed, such as *proportionality* and *envy-freeness* [Brams and Taylor, 1996; Neyman, 1946; Varian, 1974; Dubins and Spanier, 1961]. An allocation is proportional if each player's valuation for his received resource is at least $\frac{1}{n}$ fraction of his valuation for the whole resource, where n is the number of players. An allocation is envy-free if no player values another player's allocation more than his own [Steinhaus, 1948]. One of the difficulties in finding allocations which are proportional or envy-free is that the resource may not be uniformly structured, such as time and land, and different players may hold different valuations over the same part of the resource.

Most of the previous studies have been focused on the static fair division problem, which assumes that all players arrive simultaneously. Recently, the dynamic fair division problem has been considered in [Walsh, 2011; Friedman *et al.*, 2015; 2017]. A real-life application of the dynamic fair division problem is to fairly allocate the resource on a server among different jobs. In this case, different jobs arrive and depart at different time, and each job has different values for the resource allocated to it. The objective here is to balance the resource allocated to different jobs. Unlike the static setting where an envy-free allocation is guaranteed to exist [Brams and Taylor, 1995; Su, 1999], [Walsh, 2011] shows that envy-free or proportional allocations may not exist when the players arrive one by one over time. Therefore, a weak version of envy-freeness is discussed in [Walsh, 2011], where the only constraint is that each player does not envy the allocations of the players who will arrive after him. However, this may not be sufficient to capture fairness in reality because the players may still observe the allocations of the previous players and envy their allocations. For example, an allocation which allocates the whole resource to the first player is considered to be envy-free in [Walsh, 2011], which is not plausible. Thus it is natural to explore to what extent we can approximate the envy-freeness and proportionality in the dynamic setting. This problem has been discussed in the literature, but they only considered the case where the players have uniform valuations. In [Friedman *et al.*, 2015], there is a single resource, and in [Friedman *et al.*, 2017], there are multiple resources and the players have different demands for each resource. The latter one seems a general problem, but it can still be reduced to the problem with a single divisible resource while each player has piecewise linear valuations. We prove the results for settings with valuations more general than [Friedman *et al.*, 2015; 2017].

Our Models and Results. In this paper, we aim at designing algorithms for the dynamic fair division (DFD) problem so that the maximum "dissatisfaction" among all players is minimized. We allow the players to have arbitrary heterogeneous yet additive valuations, which is a general class of valuations considered in almost all the fair division literature [Procaccia, 2016]. In our model, different players arrive and depart at different time, and the decision maker needs to decide each player's allocation upon his arrival. We say an allo-

cation is ξ-envy-free if at any time, each player's valuation on his allocation is at least $\frac{1}{\xi}$ fraction of his value on any other player's allocation. An allocation is σ-proportional if at any time, each player's value on his own allocation is at least $\frac{1}{\sigma n}$ fraction of his value on the whole resource. We refer to ξ and σ as the *approximation ratio* of the allocation.

The algorithms proposed in this paper are τ-*recallable algorithms* with $\tau \in \mathbb{Z}^+$, which means when a new player arrives, the decision maker can recall the resource from at most τ previous players and reallocate it [Friedman *et al.*, 2015]. If reallocation is not permitted, it is easy to see that the approximation ratio is unbounded, because for any dynamic algorithm, the adversary can always choose the valuation for the second player such that he is only interested in the allocation of the first player. In this paper, we will focus on the 1-recallable algorithms and at the end of each section, we will briefly discuss the τ-recallable algorithms for any constant τ, since they basically share the same idea and the same bound.

Our first contribution is to design two dynamic 1-recallable algorithms for the general DFD problem which are $O(n)$-envy-free and $O(\log n)$-proportional. Then we prove that for any dynamic 1-recallable algorithm, there exists an adversary instance such that the approximation ratio of the algorithm is at least $\Omega(\frac{n}{\log n})$, even if all the players' valuations are *piecewise uniform* [Chen *et al.*, 2013]. Since the $\Omega(1)$ lower bound for proportionality is already proved in [Friedman *et al.*, 2015], all our bounds are tight up to a logarithmic factor.

Because of the strong lower bound in the general setting, no one can hope to improve the fairness for settings with valuations more general than the piecewise uniform valuations. Thus, it is natural to discover the set of valuations with better dynamic performance. In [Friedman *et al.*, 2015], the authors show that if the players' valuations are uniform, there exist $O(1)$-envy-free algorithms. In this paper, we generalize their model, by allowing different players to have different demands over the resource. When the players have uniform valuations with different demands, an allocation is *fair* if (1) it meets all the players' demands or (2) each player gets at least his demand divided by total demand [Ghodsi *et al.*, 2011]. As before, in the static setting, a fair allocation always exists. However, in the dynamic setting, the adversary can manipulate the future players to violate the fairness. We say an allocation is η-*fair* if each player gets $\frac{1}{\eta}$ fraction of his allocated resource in a fair allocation. We design an $O(\log n)$-fair 1-recallable algorithm and we prove that the bound is tight by constructing an adversary instance such that no 1-recallable algorithm can be better than $\Omega(\log n)$-fair.

Additional Related Work. The fair division problem with multiple indivisible resources is also a problem widely studied in the literature [Budish, 2011; Kurokawa *et al.*, 2018; Amanatidis *et al.*, 2017; Brandt *et al.*, 2016; Endriss, 2017]. Since the envy-free allocation cannot be guaranteed in this setting, the notion of fairness is captured by different relaxed versions, such as envy-free up to one item (EF1), which can be found in polynomial time [Budish, 2011]. A stronger fairness concept, envy-free up to any item (EFX), is proposed in [Caragiannis *et al.*, 2016]. It is shown in [Plaut and Rough-

garde, 2018] that such an allocation exists for identical valuations, but the existence of EFX allocations is still unknown for general valuations. However, the authors show that there exists an allocation which is a 2 approximation to the EFX allocation. Another solution concept adopted for indivisible resources is the maxmin share, which can be approximated within a factor of $2/3$ in polynomial time [Kurokawa *et al.*, 2018]. The approximation ratio of this problem is improved to $3/4$ in [Ghodsi *et al.*, 2017]. The authors in [Ghodsi *et al.*, 2017] also prove that the maximin fair allocation can be approximated within a factor of 3 when the valuations of the players are submodular, a factor of 5 when the valuations of the players are XOS, and a logarithmic factor when the valuations of the players are subadditive.

2 Preliminaries

Fair Division Problem. The fair division problem is to divide and allocate a divisible and heterogeneous resource fairly among n players, where the resource is represented by the real number interval $[0, 1]$ and the player set is represented by $N = \{1, \cdots, n\}$. Each player i has a valuation v_i, mapping any subset of the resource to a nonnegative value, representing i's preference over the resource. Formally, for any set $I \subseteq [0, 1]$, player i's value for taking set I is $v_i(I) \in \mathbb{R}_+ \cup \{0\}$. The valuation profile is denoted by $v = \{v_1, \ldots, v_n\}$ and the set of all possible valuation profiles is denoted by $\mathcal{V}$. In this paper, besides the most general valuations, we also consider two special ones.

One widely studied valuation is *piecewise uniform* valuation [Chen *et al.*, 2013]. A player has piecewise uniform valuation if and only if he has uniform valuation over a subset of the resource (which may not be a continuous interval). That is, for player i, the valuation v_i is piecewise uniform if and only if there exists $I_i \subseteq [0, 1]$ such that $v_i(I) = |I \cap I_i|$ for all $I \subseteq [0, 1]$. By normalizing each player's valuation to range $[0, 1]$, we have that for any $I \subseteq [0, 1]$, $v_i(I) = \frac{|I \cap I_i|}{|I_i|}$.

Another valuation studied in this paper is called *uniform with demand*. In this model, each player has uniform valuation over the resource and he only cares the size of the resources he gets rather than which part of the resource is allocated to him. In general, different players may have different demand over the resource, and when the allocated resource exceeds the player's demand, his valuation for the allocated resource will not increase. For player i, we denote his demand as $d_i \in (0, 1]$, and his valuation v_i is uniform with demand if and only if for any $I \subseteq [0, 1]$, $v_i(I) = \min\{|I|, d_i\}$. By normalizing each player's valuation to range $[0, 1]$, we have that for any $I \subseteq [0, 1]$, $v_i(I) = \min\{\frac{|I|}{d_i}, 1\}$. In our paper, we adopt the normalized form of the valuations, but all the results hold for the valuations without normalization.

An *allocation* is denoted by a partition of $[0, 1]$, $A = (A_0, A_1, \cdots, A_n)$ where A_0 is the unallocated resource and A_i is the resource allocated to player i for any $i \in N$. The set of all such possible partitions of $[0, 1]$ is denoted by $\mathcal{U}$. A fair division algorithm (or an allocation rule) $\mathcal{A}$ is a mapping from a valuation profile to an allocation, i.e., $\mathcal{A} : \mathcal{V} \to \mathcal{U}$.

Two widely adopted fairness solution concept are *proportionality* and *envy-freeness*. An allocation A is propor-

tional if and only if each player gets at least $\frac{1}{n}$ fraction of the whole resource, and an allocation A is envy-free if and only if each player will not envy any other player's allocation. In this paper, we care more about the relaxed version of them. Formally, we say an allocation A is σ-proportional if $v_i(A_i) \geq \frac{1}{\sigma n} v_i([0,1])$ for any player i, and an allocation A is ξ-envy-free if $v_i(A_i) \geq \frac{1}{\xi} v_i(A_j)$ for any player i and j.

When players' valuations are uniform with demand, a fairness solution concept stronger than proportionality is required here because for a proportional allocation A^P, it only guarantees that for each player i, $v_i(A_i^P) \geq \frac{v_i([0,1])}{n} = \frac{1}{n}$. However, in the static setting there exists an allocation A such that each player i gets $\frac{d_i}{\max\{d,1\}}$ fraction of the whole resource, where $d = \sum_{j \in N} d_j$ is the total demand of the players, and player i's value is guaranteed to be $v_i(A_i) = \frac{1}{\max\{d,1\}}$. Since the total demand d might be significantly smaller than n, $v_i(A_i)$ might be significantly larger than $v_i(A_i^P)$. Therefore, a stronger notion of fairness is defined as follows for uniform with demand valuations [Ghodsi et al., 2011]. Formally, an allocation A is *fair* if $v_i(A_i) \geq \frac{1}{\max\{d,1\}}$, and A is η-fair if $v_i(A_i) \geq \frac{1}{\eta \cdot \max\{d,1\}}$.

Dynamic Fair Division Problem. Now we extend the fair division problem to the dynamic setting, where player i arrives at time t_i and departs at time t_i^d, with $t_i^d > t_i$. Without loss of generality, we have $0 < t_1 \leq t_2 \leq \cdots \leq t_n$, where n is the maximum number of players that could possibly arrive and is given to the algorithm as an input. However, the algorithm does not know the exact number of arriving players, or the players' valuations until their arrival. The algorithm needs to allocate the resource to each player when he arrives without knowing future events, including his departure time. In this paper, we will design our algorithms and state our results for the arrival only model, where $t_1^d = t_2^d = \cdots = t_n^d > t_n$. However, all our lower bounds and upper bounds can be directly applied for settings where players have arbitrary departure time with the same approximation ratio. See the full version [Li et al., 2018] for more details.

Formally, the allocation of the algorithm at time t_i is denoted by $A^i = (A_0^i, A_1^i, \cdots, A_i^i)$ and the total output from time 0 to time t_n is denoted by $A = (A^i)_{i \in \{0,1,\cdots,n\}}$ where $A^0 = \{A_0^0\}$ and $A_0^0 = [0,1]$. An algorithm is called τ-recallable if for any $i \in [n-1]$, there exists $s \leq \min\{\tau, i\}$ and a set of players $S = \{i_1, \cdots, i_s\} \subseteq [i]$ such that $A_j^{i+1} = A_j^i$ for all $j \in [i] \backslash S$, and $A_j^{i+1} \subseteq A_j^i$ for all $j \in S$.

In the following we extend the definition for fairness to the dynamic setting. An allocation A is σ-proportional if

$$\forall i, \forall j \leq i, v_j(A_j^i) \geq \frac{1}{\sigma \cdot i} v_j([0,1]);$$

and an allocation A is ξ-envy-free if

$$\forall i, \forall j, j' \leq i, v_j(A_j^i) \geq \frac{1}{\xi} v_j(A_{j'}^i).$$

For the uniform with demand valuations in the dynamic setting, an allocation A is η-fair if

$$\forall i, \forall j \leq i, v_j(A_j^i) \geq \frac{1}{\eta \cdot \max\{\sum_{l \leq i} d_l, 1\}}.$$

In this paper, we say an algorithm $\mathcal{A}$ is σ-proportional, ξ-envy-free or η-fair for valuation space $\mathcal{V}$ if for any valuation profile $v \in \mathcal{V}$, $\mathcal{A}(v)$ is σ-proportional, ξ-envy-free or η-fair. Note that, since we do not assume that the resource must be allocated to any player, a trivial envy-free algorithm is that everyone gets nothing. Such an algorithm is called *empty* in this paper.

Remark. In this paper, we only consider deterministic algorithms because there exists a trivial randomized algorithm which is proportional and envy-free in expectation in the dynamic setting [Chen et al., 2013]. This randomized algorithm is also fair for uniform with demand valuations.

3 Dynamic Fair Division (DFD) Problem

In this section, we consider the players' values to be additive. First we state the results for proportionality. In Theorem 6.1 of [Friedman et al., 2015], no 1-recallable algorithm can be better than $(2 \ln 2)$-proportional even if the players' valuations are uniform, i.e., $v_i(I) = |I|$, for any $i \in [n]$. In the following, we design the 1-recallable algorithm $\mathcal{A}_{DFD}^1$, defined in Algorithm 1, which is $O(\log n)$-proportional for the DFD problem with general valuations. Roughly speaking, when player i arrives, algorithm $\mathcal{A}_{DFD}^1$ divides the previous players' allocated resource into $2i(3 + \ln i)$ subsets and lets player i choose his favorite bundle of resource from one player, where the size of the bundle depends on the current approximation ratio of any player j who arrived before player i. The performance of algorithm $\mathcal{A}_{DFD}^1$ is formally analyzed in Theorem 1.

Algorithm 1 1-Recallable Algorithm $\mathcal{A}_{DFD}^1$

Input: A sequence of players $N = \{1, \cdots, n\}$ arriving and departing along time.
1: Initially, $A_j^i = \emptyset$ for all $1 \leq i \leq n$ and $0 \leq j \leq i$.
2: When the first player arrives, setting $A_1^1 = [0,1]$.
3: **for** any arriving player $i > 1$ **do**
4:   $\sigma = \lfloor 2i(3 + \ln i) \rfloor$.
5:   **for** $0 < j < i$ **do**
6:     Partition A_j^{i-1} into σ sets $\{A_{j,1}^{i-1}, \cdots, A_{j,\sigma}^{i-1}\}$ with $v_j(A_{j,1}^{i-1}) = \cdots = v_j(A_{j,\sigma}^{i-1})$.
7:     $\sigma_j = \lceil \frac{v_j([0,1])}{v_j(A_j^{i-1})} \rceil$.
8:   **end for**
9:   $(j^*, S^*) = \arg \max\limits_{\substack{j < i, S \subseteq [\sigma], \\ |S| = \sigma - \sigma_j}} \{\sum_{k \in S} v_i(A_{j,k}^{i-1})\}$.
10:   $A_i^i = \cup_{k \in S^*} A_{j^*,k}^{i-1}$.
11:   $A_{j^*}^i = \cup_{k \notin S^*} A_{j^*,k}^{i-1}$.
12:   $A_j^i = A_j^{i-1}, \forall j \in [i-1] - \{j^*\}$.
13: **end for**
Output: Allocation $A = (A_j^i)_{0 \leq i \leq n, 0 \leq j \leq i}$.

Theorem 1. *For any $n \geq 1$, 1-recallable algorithm $\mathcal{A}_{DFD}^1$ is $O(\log n)$-proportional for the DFD problem with n players.*

Proof. In the following, we prove the theorem by induction. When there is only 1 player, algorithm $\mathcal{A}^1_{DFD}$ is proportional since the player gets the whole resource. Assuming that when there are $i-1$ players, algorithm $\mathcal{A}^1_{DFD}$ is $2(3+\ln(i-1))$-proportional, we show that when player i arrives at time t_i, algorithm $\mathcal{A}^1_{DFD}$ is $2(3+\ln i)$-proportional.

First, we show that for any $j < i$, player j is $2(3+\ln i)$-proportional at time t_i. Note that at least σ_j elements from $\{A^{i-1}_{j,1}, \cdots, A^{i-1}_{j,\lfloor 2i(3+\ln i)\rfloor}\}$ is allocated to player j and player j has the same value for them. Thus,

$$\frac{v_j(A^i_j)}{v_j([0,1])} \geq \frac{\sigma_j \cdot v_j(A^{i-1}_j)}{\lfloor 2i(3+\ln i)\rfloor \cdot v_j([0,1])} \geq \frac{1}{2i(3+\ln i)}. \quad (1)$$

Next, note that for player i, the worst case happens when his value for the resource allocated to player j is $\frac{\rho_j \cdot v_i([0,1])}{\sum_{j\leq i-1}\rho_j}$, where $\rho_j = \frac{\lfloor 2i(3+\ln i)\rfloor}{\lfloor 2i(3+\ln i)\rfloor - \sigma_j}$. In any other cases, by choosing his favorite bundle, player i's value will increase. Thus,

$$\frac{v_i([0,1])}{v_i(A^i_i)} \leq \sum_{1\leq j\leq i-1} \frac{\lfloor 2i(3+\ln i)\rfloor}{\lfloor 2i(3+\ln i)\rfloor - \sigma_j}$$
$$\leq 2\sum_{1\leq x\leq\lceil\frac{i-1}{2}\rceil} \frac{\lfloor 2i(3+\ln i)\rfloor}{\lfloor 2i(3+\ln i)\rfloor - \lceil 2(i-x)(3+\ln(i-x))\rceil}$$
$$\leq 2\sum_{1\leq x\leq\lceil\frac{i-1}{2}\rceil} \frac{2i(3+\ln i)}{2i(3+\ln i) - 2(i-x)(3+\ln i) - 2}$$
$$= 2i\sum_{1\leq x\leq\lceil\frac{i-1}{2}\rceil} \frac{1}{x - \frac{2}{3+\ln i}} < 2i(3+\ln i)$$

The second inequality holds by carefully analyzing the value σ_j for each player j who arrived before player i. Note that at any time t_j, the algorithm can only make the assignment such that the approximation ratio of the new arriving player and the player whose resource is recalled is $2j(3+\ln j)$, while the approximation ratio for all other players remain unchanged. Thus when player i arrives, there exist at least $i+1-2x$ players with approximation ratio less than $2(i-x)(3+\ln(i-x))$. By properly eliminating the rounding in the term, the third inequality holds. Therefore, by induction, algorithm $\mathcal{A}^1_{DFD}$ is $O(\log n)$-proportional and Theorem 1 holds. □

Next we show how to design the $O(n)$-envy-free algorithm $\mathcal{A}^2_{DFD}$, which is quite similar to algorithm $\mathcal{A}^1_{DFD}$. Briefly speaking, $\mathcal{A}^2_{DFD}$ allocates the whole resource to the first player. Then, when player i arrives, for each player $j < i$, $\mathcal{A}^2_{DFD}$ uniformly divides the resource allocated to player j into i equal subsets and let player i chooses his favorite set. The formal description is omitted here due to the space limit.

Theorem 2. *For any $n \geq 1$, 1-recallable algorithm $\mathcal{A}^2_{DFD}$ is $O(n)$-envy-free for the DFD problem with n players.*

Proof. Again we prove by induction. When player 1 arrives, A^1_1 is exactly $[0,1]$. That is, $\mathcal{A}^2_{DFD}$ is envy-free. Assume that for player $i \geq 2$, $\mathcal{A}^2_{DFD}$ is $(i-1)$-envy-free at time t_{i-1}. Now let us prove that all i players are i-envy-free at time t_i by analyzing their value at time t_i.

First, we prove that player i is i-envy-free at time t_i. Assume that the allocation of player j, A^{i-1}_j, is partitioned into i sets $\{A^{i-1}_{j,1}, \cdots, A^{i-1}_{j,i}\}$. Since player i is allocated with his favorite set $A^{i-1}_{j^*,k^*} \in \{A^{i-1}_{j,k}\}_{j<i,k\leq i}$, for any $j < i$,

$$v_i(A^i_i) \geq v_i(A^{i-1}_{j^*,k^*}) \geq \frac{1}{i}v_i(A^{i-1}_j) \geq \frac{1}{i}v_i(A^i_j). \quad (2)$$

Next, we show that for any $j \in [i-1]$, player j is i-envy-free at time t_i. By induction hypothesis, for any $j, j' \in [i-1]$,

$$v_j(A^{i-1}_j) \geq \frac{1}{i-1}v_j(A^{i-1}_{j'}).$$

Note that A^{i-1}_j is partitioned into i sets with equal values, and at most one of them is allocated to player i. Hence, for any $j, j' \in [i-1]$,

$$v_j(A^i_j) \geq \frac{i-1}{i}v_j(A^{i-1}_j) \geq \frac{1}{i}v_j(A^{i-1}_{j'}) \geq \frac{1}{i}v_j(A^i_{j'}). \quad (3)$$

Since the allocation of player i at time t_i is also a subset of player j^*'s allocation at time $i-1$, according to Inequality 3, we have

$$v_j(A^i_j) \geq \frac{1}{i}v_j(A^{i-1}_{j^*}) \geq \frac{1}{i}v_j(A^i_i). \quad (4)$$

Combining Inequalities 2, 3 and 4, all players are i-envy-free at time t_i. By induction, algorithm $\mathcal{A}^2_{DFD}$ is n-envy-free. Thus finishes the proof of Theorem 2. □

We have designed an 1-recallable algorithm which is $O(n)$-envy-free for the DFD problem with general valuations. Next we show that this bound is almost tight by proving that all 1-recallable algorithms must be at least $\Omega(\frac{n}{\log n})$-envy-free, even if the players' valuations are piecewise uniform (DFDPU). Formally, we have the following theorem.

Theorem 3. *For any 1-recallable algorithm $\mathcal{A}$ (except empty algorithm) for the DFDPU problem, $\mathcal{A}$ is $\Omega(\frac{n}{\log n})$-envy-free with n players.*

Proof. Assume there exists an algorithm $\mathcal{A}$ which is ξ-envy-free and not empty. Consider the following adversary instance where for player 1, $v_1(I) = |I|$ and for any player $i \geq 2$, $v_i(I) = \frac{|I \cap A^{i-1}_1|}{|A^{i-1}_1|}$. That is, the first player wants the whole resource and each new arriving player only wants the resource allocated to the first player by $\mathcal{A}$ at time t_{i-1}. Since $\mathcal{A}$ is ξ-envy-free and not empty, we must have $A^i_1 \neq \emptyset$ for any $i \leq n$, and the adversary instance is well defined.

First we consider the envy-freeness for player $i \geq 2$. Since the resource that player i values is exactly the resource allocated to player 1 at time t_{i-1}, to ensure that player i is ξ-envy-free at time t_i, the algorithm must recall the resource from player 1's and reallocate at least $\frac{1}{1+\xi}$ fraction of the resource to player i. Therefore,

$$v_1(A^i_1) \leq (1 - \frac{1}{1+\xi})v_1(A^{i-1}_1),$$

for any $i \geq 2$. Accordingly,

$$v_1(A^n_1) \leq (1 - \frac{1}{1+\xi})^{n-2}v_1(A^2_1). \quad (5)$$

When all n players have arrived, to ensure that player 1 is ξ-envy-free, we have

$$v_1(A_1^n) \geq \frac{1}{\xi} v_1(A_2^n) = \frac{1}{\xi} v_1(A_2^2)$$

$$= \frac{1}{\xi |A_1^1|} v_2(A_2^2) \geq \frac{1}{\xi^2 |A_1^1|} v_2(A_1^2) = \frac{1}{\xi^2} v_1(A_1^2). \quad (6)$$

The first and the second inequality holds due to the ξ-envy-freeness of player 1 at time t_n and player 2 at time t_2. The first equality holds because player 1 is always the player who needs to be recalled. Therefore, $A_2^i = A_2^2$ for any $2 \leq i \leq n$. The second and the last equality holds because by the construction of the valuations of player 1 and player 2, $v_1(I) = \frac{1}{|A_1^1|} v_2(I)$ for any $I \subseteq A_1^1$.

Combining inequalities 5 and 6, we have

$$(1 - \frac{1}{1+\xi})^{n-2} \geq \frac{1}{\xi^2}.$$

By solving the above inequality, $\xi = \Omega(\frac{n}{\log n})$ and Theorem 3 holds. □

Remark. Given a valuation profile v, we say an allocation A is *non-wasteful* if for all player i, we only allocate the resource that he values to him. That is, for any $i \leq j \leq n$ and $I \subseteq A_i^j$, $v_i(I) > 0$. We say an algorithm $\mathcal{A}$ is *non-wasteful* for valuation space $\mathcal{V}$ if for any $v \in \mathcal{V}$, $\mathcal{A}(v)$ is non-wasteful. Then if we consider the τ-recallable algorithm where τ can be any constant, the adversary instance in the proof of Theorem 3 still shows that no non-wasteful algorithm can be $o(\frac{n}{\log n})$-envy-free even for piecewise uniform valuations.

4 When the Valuation Function is Uniform with Demand (UD)

In this section, we study the dynamic fair division problem with uniform with demand valuations. Since in this model, the players do not care which part of the resource is allocated to him, we regard the allocation A_j^i in $A = (A_j^i)_{0 \leq i \leq n, 0 \leq j \leq i}$ as the size of the allocated resource.

For the UD problem, a simple greedy algorithm is 2-envy-free and 2-proportional. However, our main focus in this section is its performance under the more demanding fairness solution concept. First we consider the following special case where the smallest demand is within a constant fraction of the largest demand. This case will provide enough intuition about solving the general case. In order to make our algorithm clear, we explicitly write out all the parameters in $\mathcal{A}_{UD}^S(d, c, \eta)$, where the algorithm has those extra parameters as input. Here parameter d is the minimum demand in the adversary instance and c is the ratio between the largest and smallest demand. Note that $cd \leq 1$. Also, η is an extra parameter representing the total amount of resource can be allocated to the arriving players. Algorithm $\mathcal{A}_{UD}^S$ is formally defined in Algorithm 2. The general idea of the algorithm is to treat each arriving player's demand as the maximum demand, and the total demand as the sum of the minimum demand. Then the algorithm only allocates the resource to each arriving player such that the approximation ratio is not violated. The performance of the algorithm is analyzed in Lemma 1.

Algorithm 2 1-Recallable Algorithm $\mathcal{A}_{UD}^S(d, c, \eta)$

Input: A sequence of players $N = \{1, \cdots, n\}$ arriving and departing along time.
1: Initially, $A_j^i = 0$ for all $0 \leq i \leq n, 0 \leq j \leq i$.
2: **for** any arriving player $i \geq 1$ **do**
3:      Let $j^* = \arg \max_{1 \leq j \leq i-1} \{A_j^{i-1}\}$.
4:      Set $A_i^i = A_{j^*}^i = \frac{d}{2\eta \ln 3 \cdot \max\{d \cdot i, 1\}}$.
5:      $A_j^i = A_j^{i-1}, \forall j \in [i-1] - \{j^*\}$.
6:      $A_0^i = A_0^0 - \sum_{j \in [i]} A_j^i$.
7: **end for**
Output: Allocation $A = (A_j^i)_{0 \leq i \leq n, 0 \leq j \leq i}$.

Lemma 1. *For any $c, d, \eta > 0$, algorithm $\mathcal{A}_{UD}^S$ is $(2c\eta \ln 3)$-fair and at most $\frac{1}{\eta}$ fraction of the resource is allocated.*

Proof. We first show that algorithm $\mathcal{A}_{UD}^S$ is well defined by showing that the reallocation in Step 4 is always feasible and the total allocated resource is no more than $\frac{1}{\eta}$. First note that $\frac{d}{2\eta \ln 3 \cdot \max\{i \cdot d, 1\}}$ is non-increasing with respect to i. Next we divide the analysis into three cases.

Case 1: $i \leq \lfloor \frac{1}{d} \rfloor$. When $i \leq \lfloor \frac{1}{d} \rfloor$, $A_j^i = \frac{d}{2\eta \ln 3}$ for all $j \leq i$. Therefore, $\sum_{j \in [i]} A_j^i = \frac{id}{2\eta \ln 3} \leq \frac{1}{2\eta \ln 3} < \frac{1}{\eta}$. In this case, no resource is recalled and the total allocation never exceed $\frac{1}{\eta}$.

Case 2: $\lfloor \frac{1}{d} \rfloor < i \leq 2\lfloor \frac{1}{d} \rfloor$. In this case, the first $\lfloor \frac{1}{d} \rfloor$ players still satisfy Case 1, but when the $(\lfloor \frac{1}{d} \rfloor + l)$-th player arrives, player l's allocation will be recalled, where $1 \leq l \leq i - \lfloor \frac{1}{d} \rfloor$. Then both player l and $(\lfloor \frac{1}{d} \rfloor + l)$ will be allocated with $\frac{1}{2\eta \ln 3 \cdot (\lfloor \frac{1}{d} \rfloor + l)}$. The total allocated resource is

$$(2\lfloor \frac{1}{d} \rfloor - i) \frac{d}{2\eta \ln 3} + 2 \sum_{l=1}^{i - \lfloor \frac{1}{d} \rfloor} \frac{1}{2\eta \ln 3 \cdot (\lfloor \frac{1}{d} \rfloor + l)} \leq \frac{1}{\eta}.$$

The above inequality holds by analyzing the monotonicity.

Case 3: $i > 2\lfloor \frac{1}{d} \rfloor$. When $i > 2\lfloor \frac{1}{d} \rfloor$, all the resource allocated to the first $\lfloor \frac{i}{2} \rfloor$ players will be recalled. In Algorithm $\mathcal{A}_{UD}^S$, whenever the resource is recalled from a player, the algorithm will create two sets of resource with the same size, and allocate them to both the recalled player and the new arriving player. Thus the total allocated resource in this case is at most twice of the resource allocated to players from $\lfloor \frac{i}{2} \rfloor$ to i, that is,

$$2 \sum_{l=\lfloor \frac{i}{2} \rfloor}^{i} \frac{1}{2\eta \ln 3 \cdot l} \leq \frac{1}{\eta \ln 3} \ln \frac{i}{\lfloor \frac{i}{2} \rfloor} \leq \frac{1}{\eta}.$$

Finally, we analyze the performance of $\mathcal{A}_{UD}^S$. At anytime t_i, $1 \leq i \leq n$, since player i's demand $d_i \leq c \cdot d$, his value for allocation A_i^i is $v_i(A_i^i) \geq \frac{1}{2c\eta \ln 3 \cdot \max\{i \cdot d, 1\}}$. When player i arrives, the total demand $\sum_{l \leq i} d_l \geq d \cdot i$, and the value of player i at time t_i is

$$v_i(A_i^i) \geq \frac{1}{2c\eta \ln 3 \cdot \max\{\sum_{l \leq i} d_l, 1\}}.$$

Similarly, the value of any player $j < i$ satisfies the above inquality. Therefore, algorithm $\mathcal{A}_{UD}^S$ is $(2c\eta \ln 3)$-fair, and thus finishes the proof of Lemma 1. □

Next we design the algorithm $\mathcal{A}_{UD}$ for the valuations with arbitrary demands, which is formally defined in Algorithm 3. The idea for this algorithm is to classify the players with demands that differ up to a constant factor into the same group and run the algorithm $\mathcal{A}_{UD}^S$ for the players in each group. By analyzing the performance of algorithm $\mathcal{A}_{UD}$, we have the following theorem.

Algorithm 3 1-Recallable Algorithm $\mathcal{A}_{UD}$

Input: A sequence of players $N = \{1, \cdots, n\}$ arriving and departing along time. Each player $i \in N$ has demand d_i. Let $m = \lceil \log n \rceil$.
1: Divide the demand space $[0, 1]$ into $m + 1$ sets $(S_0, S_1, \ldots, S_m)$, where $S_0 = [0, 2^{-m}]$, $S_l = (2^{-l}, 2^{1-l}], l \in [m]$.
2: **for** player i with demand d_i **do**
3:     Find j such that $d_i \in S_j$.
4:     $A_i^i = \frac{2^{1-j}}{2\eta \ln 3 \max\{\sum_{l \le i} d_l, 1\}}$.
5:     Let $i^* = \arg\max_{i'}\{A_{i'}^{i-1}|d_{i'} \in S_j\}$.
6:     $A_{i^*}^i = A_i^i$, and $A_{i'}^i = A_{i'}^{i-1}, \forall i' \neq i, i^*$.
7:     $A_0^i = 1 - \sum_{j \in [i]} A_j^i$.
8: **end for**
Output: Allocation $(A_j^i)_{0 \le i \le n, 0 \le j \le i}$.

Theorem 4. *For any $n \ge 1$, 1-recallable algorithm $\mathcal{A}_{UD}$ is $O(\log n)$-fair for the UD problem with n players.*

Proof. It is clear that $\mathcal{A}_{UD}$ classifies all players in $m + 1$ sets. Let N_j denote the set of players with demand in S_j. Then except the players in N_0, the demands of the players in the same set differ up to a constant factor 2.

First we find the parameter η such that algorithm $\mathcal{A}_{UD}$ is feasible. For players in N_0, since the maximum demand is at most $\frac{1}{n}$ and there are at most n players in total, the resource allocated to those players is bounded by $\frac{1}{2\eta \ln 3}$. For player i in N_j, where $j \in [m]$, the total demand when player i arrives is at least the total demand of the players in N_j who arrived before player i. Using the similar argument in Lemma 1 and applying the case studies, the total resource allocated to players in N_j is at most $\frac{1}{\eta}$. By setting $\eta = 1 + m = 1 + \lceil \log n \rceil$, algorithm $\mathcal{A}_{UD}$ is always feasible. Then, similar to Lemma 1, it is easy to get that algorithm $\mathcal{A}_{UD}$ is $(4\eta \ln 3)$-fair, which is $O(\log n)$-fair. Therefore, Theorem 4 holds. □

Next we construct an adversary instance for the UD problem to show that the bound of our algorithm is tight. Formally, we have the following theorem.

Theorem 5. *For any 1-recallable algorithm $\mathcal{A}$ for the UD problem, $\mathcal{A}$ is $\Omega(\log n)$-fair with n players.*

Proof. Let $\mathcal{A}$ be any 1-recallable algorithm which is η-fair. Here we consider the following adversary instance. In order to make the instance clear, we describe the arriving play-

ers by $m = \frac{\log n}{3}$ rounds (n is a large enough integer such that m is also a large enough integer). Formally, within stage $j \le m$, $n_j = \frac{n}{2 \times 4^{j-1}}$ players arrive one by one and each of them has demand $\frac{8^j}{n}$. That is, in the first round, $\frac{n}{2}$ players arrive one by one and each of them has demand $\frac{8}{n}$. For each of the following rounds, the number of the arriving players decreases by a multiplicative factor of 4, but the demand of each player increases by a multiplicative factor of 8. Note that in the last round the demand of each player is $\frac{8^m}{n} = 1$. Also, the total number of players in the designed instance is $\sum_{j \le m} n_j < n$. Thus this instance is well defined, and it is sufficient to prove that all 1-recallable algorithm is $\Omega(\log n)$-fair for this instance.

In each stage $j \le m$, the total number of players that would arrive in the future is less than

$$u = \left(\frac{n}{2 \times 4^j}\right)/\left(1 - \frac{1}{4}\right) = \frac{n}{6 \times 4^{j-1}}.$$

It is easy to see that $u = \frac{1}{3}n_j$. Therefore, at least $\frac{2}{3}n_j = \frac{n}{3 \times 4^{j-1}}$ players who arrive during the jth round won't get recalled by in the future. Then we can lower bound the resource allocated to those players. First note that the total demand before stage j is $\sum_{j' < j} \frac{n}{2 \times 4^{j'-1}} \times \frac{8^{j'}}{n} = \sum_{j' < j} \frac{8^{j'}}{2 \times 4^{j'-1}}$. The total demand when the ith player in stage j arrives is

$$D_i = \sum_{j' < j} \frac{8^{j'}}{2 \times 4^{j'-1}} + \frac{8^j \cdot i}{n}.$$

Since algorithm $\mathcal{A}$ is an 1-recallable algorithm, for any integer $x \le \frac{n_j+1}{2}$, there exists at least $n_j + 1 - 2x$ players whose allocated resource is at least $\frac{8^j}{n\eta \cdot D_{n_j-x}}$. Thus the total resource allocated to the players in stage j is

$$L_j \ge \frac{2}{\eta} \sum_{i=1}^{\frac{n}{6 \times 4^{j-1}}} \frac{8^j}{n \cdot D_{n_j-i}} \ge \frac{2 \ln \frac{6}{5}}{\eta},$$

when we set $n \to \infty$. The computation detail of the above inequality can be found in the full version of this paper [Li *et al.*, 2018]. Thus the total allocated resource is

$$1 \ge \sum_{j \le \frac{\log n}{3}} L_j \ge \frac{2 \ln \frac{6}{5} \log n}{3\eta}.$$

Therefore, $\eta \ge \frac{2}{3} \ln \frac{6}{5} \log n$ and Theorem 5 holds. □

Remark If we consider τ-recallable algorithms here, where τ is a constant, by designing the similar adversary instance, except that the number of players and the demand change in the ratio of $\frac{1}{4\tau}$ and 8τ respectively between each stage, we can still obtain the $\Omega(\log n)$ lower bound.

5 Conclusion and Future Direction

In this paper, we study the dynamic fair division problem with a divisible resource. We design $O(\log n)$-proportional and $O(n)$-envy-free algorithms for players with general valuations. We also show that those bounds are tight up to a

logarithmic factor. However, for the valuations that are uniform with demand, we design an $O(\log n)$-fair algorithm and we prove that this ratio is tight.

There are a lot of future directions that are worthy exploration. An immediate open question is whether we can find algorithms with better approximation ratio for τ-recallable algorithms without the non-wasteful assumption when $\tau \geq 2$. Another particularly interesting question is to consider the problem via a game theoretic view in the dynamic setting. Moreover, the dynamic version of allocating multiple indivisible resources is also worthy of effort.

Acknowledgments

References

[Alon, 1987] Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.

[Amanatidis *et al.*, 2017] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):52, 2017.

[Aziz and Mackenzie, 2016] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *57th Annual Symposium on Foundations of Computer Science (FOCS'16)*, pages 416–427, 2016.

[Brams and Taylor, 1995] Steven J Brams and Alan D Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

[Brams and Taylor, 1996] Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[Brandt *et al.*, 2016] Felix Brandt, Vincent Conitzer, Ulle Endriss, Ariel D Procaccia, and Jérôme Lang. *Handbook of computational social choice*. Cambridge University Press, 2016.

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

[Caragiannis *et al.*, 2016] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. In *17th ACM Conference on Economics and Computation (EC'16)*, pages 305–322, 2016.

[Chen *et al.*, 2013] Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284–297, 2013.

[Dubins and Spanier, 1961] Lester E Dubins and Edwin H Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, 1961.

[Endriss, 2017] Ulle Endriss. *Trends in Computational Social Choice*. Lulu. com, 2017.

[Friedman *et al.*, 2015] Eric Friedman, Christos-Alexandros Psomas, and Shai Vardi. Dynamic fair division with minimal disruptions. In *16th ACM Conference on Economics and Computation (EC'15)*, pages 697–713, 2015.

[Friedman *et al.*, 2017] Eric Friedman, Christos-Alexandros Psomas, and Shai Vardi. Controlled dynamic fair division. In *18th ACM Conference on Economics and Computation (EC'17)*, pages 461–478, 2017.

[Ghodsi *et al.*, 2011] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *8th USENIX conference on Networked systems design and implementation (NSDI'11)*, pages 323–336, 2011.

[Ghodsi *et al.*, 2017] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvement and generalization. *arXiv:1704.00222*, 2017.

[Kurokawa *et al.*, 2018] David Kurokawa, Ariel D Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM (JACM)*, 65(2):8, 2018.

[Li *et al.*, 2018] Bo Li, Wenyang Li, and Yingkai Li. Dynamic fair division problem with general valuations. *arXiv:1702.01416*, 2018.

[Neyman, 1946] Jerzy Neyman. Un theoreme d'existence. *C.R. Acad. Sci. Paris*, 222:843–845, 1946.

[Plaut and Roughgarde, 2018] Benjamin Plaut and Tim Roughgarde. Almost envy-freeness with general valuations. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'18)*, pages 2584–2603, 2018.

[Procaccia, 2016] Ariel D Procaccia. Cake cutting algorithms. In *Handbook of Computational Social Choice, chapter 13*. Cambridge University Press, 2016.

[Robertson and Webb, 1998] Jack Robertson and William Webb. Cake-cutting algorithms: Be fair if you can. *AK Peters/CRC Press*, 1998.

[Steinhaus, 1948] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[Stromquist, 1980] Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.

[Su, 1999] Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American mathematical monthly*, 106(10):930–942, 1999.

[Varian, 1974] Hal R Varian. Equity, envy, and efficiency. *Journal of economic theory*, 9(1):63–91, 1974.

[Walsh, 2011] Toby Walsh. Online cake cutting. *Algorithmic Decision Theory*, pages 292–305, 2011.