Fair Randomized Allocations under Lexicographic Valuations

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

C Surya Panchapakesan

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Dr. Rohit Vaish

Certificate

This is to certify that this dissertation entitled "Fair Randomized Allocations under Lexicographic Valuations" towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by C Surya Panchapakesan at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi under the supervision of Dr. Rohit Vaish, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Delhi , during the academic year 2024-25.

Rohit Vaish

Dr. Rohit Vaish

Committee:

Dr. Rohit Vaish

Prof. Soumen Maity

This thesis is dedicated to Amma, Appa and Avanti

Declaration

I hereby declare that the matter embodied in the report entitled "Fair Randomized Allocations under Lexicographic Valuations " are the results of the work carried out by me at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi, under the supervision of Dr. Rohit Vaish and the same has not been submitted elsewhere for any other degree.

C Surya Panchapakesan

Acknowledgments

First and foremost, I owe my sincerest gratitude to my supervisor, Dr. Rohit Vaish for his exceptional mentoring. Beyond being an outstanding researcher, his patience and encouragement have meant the world to me. I have learnt a great deal from our interactions - both academically and beyond. He has definitely raised the bar for what to expect from a supervisor, and I hope to carry forward everything that I've learnt from him throughout my research career. I also extend my sincere thanks to Vignesh and Jatin for their insightful discussions and ideas, which greatly improved the contents of this thesis.

My gratitude also goes to Prof. Soumen Maity for introducing me to Theory CS as well as for providing my first teaching experience. I would also like to mention Prof. M.S. Madhusudhan, under whom I undertook my first ever research project. Our discussions over the past five years have profoundly shaped my perspective on science and pursuing a career in research. The interactions I've had with all my mentors have been incredibly rewarding, and I hope to pay it forward some day.

The past year has been quite the bumpy road, full of untimely twists and turns - and I am forever grateful to Ancy Aunty, Joby Uncle, and Jason for being my home away from home. Their warmth and support made all the difference. All my friends at IISER have been nothing short of family, standing by me through every high and low over the last five years. The memories we've shared are irreplaceable, and I am truly lucky to have had them by my side.

Finally, to the unsung heroes - my parents and my sister. Their unwavering support and belief in me have been my greatest strength. My sister, in particular, has been one of the reasons behind my love for teaching and mathematics — she was the first person I ever taught math to, and has since then, ensured I never lost touch with that passion.

Abstract

In this thesis, we study the problem of *fairly* dividing a set of indivisible items among a group of agents. Envy-free (EF) allocations might fail to exist in this setting, motivating the study of relaxed notions such as EF1 and EFX. While EF1 allocations can always be computed efficiently, the existence of EFX (even for $n \geq 4$ additive agents) remains one of fair division's largest unresolved problems to date. However, upon restricting our domain to lexicographic valuations - a subclass of additive functions - EFX allocations are known to always exist.

When randomization is allowed, it is possible to achieve EF in expectation (ex-ante) - simply choosing an agent uniformly at random and assigning all the goods to them. However, preserving ex-ante fairness while ensuring that every deterministic allocation in the support also satisfies strong fairness guarantees (ex-post) is a far more non-trivial problem. In this thesis, we detail our attempts to compute randomized allocations that are simultaneously ex-ante EF and ex-post EFX, for agents with lexicographic valuations. Our main result is a polynomial time algorithm which computes an ex-ante $6/7$ -EF and ex-post EFX+PO allocation for lexicographic goods. For the chores setting, we provide an algorithm that gives ex-ante EF and ex-post EFX, but fails ex-post PO. Finally, we show that by relaxing ex-ante EF to ex-ante PROP, it is possible to obtain ex-post EFX + PO for both goods and chores. We also discuss some of our alternative approaches and study their guarantees and limitations.

Contents

Introduction

The central problem in fair division is to partition or allocate a set of resources among a set of agents who have diverse and heterogeneous preferences over the resources. This is a very natural problem that arises in numerous real-world scenarios such as rent division, land partitioning, divorce settlements, division of inheritance, electronic frequency allocation among others. Roughly speaking, the goal is to allocate resources in a *fair* and *efficient* manner and has been widely studied by economists, mathematicians and computer scientists over the past century.

Most of the classic work on the problem has been devoted to the fair division of *divisible* resources - items that could be assigned fractionally among agents. This line of research dates back to Steinhaus (1948) [18], who introduced the well-known cake-cutting problem. The quintessential fairness notion is *envy-freeness* (EF) [1], where every agent prefers their own assignment over that of any other. As a very simplistic model, consider a one-dimensional cake represented by the $[0, 1]$ interval, where each agent has a valuation density function that dictates their utilities for different sub-intervals. The $[0, 1]$ interval needs to be partitioned into n *pieces* such that every agent prefers their own piece of cake to any of the others'. While the existence of such a partition was long established [4], it was only in 2016 that a bounded time algorithm was discovered [3], with a staggering upper bound of $O(n^{n^{n^{n^{n^n}}}})$. On the other hand, the best known lower bound for envy free cake cutting is $\Omega(n^2)$. Closing this gap is one of the largest open problems in the divisible setting [27].

On the other hand, when the resources are *indivisible* (each item has to be assigned wholly to some agent), the nature of the problem becomes that of combinatorial optimization. However, in this setting, EF allocations need not even exist - take the case where there are two agents, but just one item. Furthermore, computing if an instance admits an envy-free allocation is shown to be NP-complete.

Naturally, one proceeds to consider relaxed notions of fairness such as *envy-freeness upto one good* (EF1) [5]. An allocation is said to be EF1 if every agent prefers their own bundle over any other agent's, after removing atmost one good from the other agent's bundle. It is known that (unlike EF) EF1 allocations always exist for monotone valuations and can be found in polynomial time [8]. Caragiannis *et al.* [6] define a stronger notion, called *envy-freeness upto any good* (EFX) where envy between agents can be eliminated by the removal of *any* single good from the envied bundle. Given that EF allocations could fail to exist, EFX is widely regarded as the next best fairness guarantee in the indivisible setting. However, it has proven to be a notoriously challenging problem - even for just four agents with additive valuations, the existence of EFX allocations remains elusive.

In some special cases, such as when the marginal gain of any good is either 0 or 1 [10], or when agents can be grouped into less than three types, and agents in a single type share a common valuation function [17], EFX is known to exist. Another special case where EFX is known to exist, is when the agents have *lexicographic preferences* over the items. In fact, not only do EFX allocations always exist, they can be computed in polynomial time.

Another line of work in fair division that has taken significant attention in the past few years, is that of "Best of Both Worlds" fairness (BoBW). Consider a real-life scenario where all the goods are replenished in a periodic manner and have to be allocated among the agents repeatedly. Deterministic algorithms that achieve fair allocations, upon repeated allocation could be systematically rigged in order to discriminate a specific agent or a group. One would require some randomness over the allocations to counter this fundamental bias present in deterministic allocations. A randomized allocation can be thought of as a probability distribution over deterministic allocations. A randomized allocation $\mathcal{A}$ is said to be *ex-ante* EF if no agent envies any other agent in expectation. For some property $\langle Q \rangle$, $\mathcal{A}$ is said to be *ex-post* $\langle Q \rangle$, if every allocation in the support of $\mathcal{A}$ satisfies fairness notion $\langle Q \rangle$.

It is straightforward to obtain just *ex-ante* EF: pick an agent uniformly at random and assign all the goods to that agent. In expectation, every agent receives the same bundle. However, this results in significant envy *ex post*, as a single agent receives everything while all others get nothing. Aziz *et al.* [13] show that for additive agents, an allocation that is *simultaneously* ex-ante EF and ex-post EF1 can be computed in poly. time. This approach of computing randomized allocations with simultaneous fairness guarantees has since been termed as the "Best of Both Worlds" (BoBW) approach.

Achieving randomized allocation for additive agents with stronger ex-post guarantees such as ex-post EFX, while preserving ex-ante EF is still an open question. We now know that EFX allocations always exist when the agent have lexicographic valuations. This motivates our central question: *Given any instance with lexicographic valuations, can we efficiently compute a randomized allocation that is ex-ante EF and ex-post EFX? Additionally, could we achieve ex-post PO as well?*

Organization of the thesis and Original Contributions

Although this thesis focuses on a specific problem, we begin by introducing fundamental questions in fair division for readers unfamiliar with the field and highlight some key results. The focus of this thesis is on *discrete* fair division as mentioned earlier.

Chapter 1 provides a summary of computational complexity and basic fair division concepts and notations, which shall be used throughout the thesis.

Chapter 2 is a review chapter which serves as all the background and motivation for our problem. It is divided into 4 sections:

- Section 2.1 highlights some fundamental algorithms for achieving different fairness notions such as EF1, EFX and PROP etc. for instances across different valuation classes (additive and monotone) and problem settings (e.g. goods-only, chores-only).
- Section 2.2 talks in detail about lexicographic preferences. We study this subclass of additive functions, and compare which properties and results are strengthened from the additive setting. Finally, it also includes an algorithm for characterizing all EFX+PO allocations for lexicographic goods instances. Additionally, we briefly discuss lexicographic chores and mixed instances, identifying key differences in the computational hardness of problems across the different settings.
- Section 2.3 is about the introductory work on BoBW fairness and techniques related to fractional allocations. We first examine the Simultaneous Eating algorithm, a mecha-

nism with strong fairness guarantees such as SD-envy-freeness, ex-ante EF, and ordinal efficiency. Next we study the Birkhoff -von Neumann decomposition - a very useful algorithm to decompose any bi-stochastic matrix into a convex combination of permutation matrices. Finally, we talk about the *PS-Lottery* algorithm which computes an allocation that is ex-ante EF and ex-post EF1 for additive agents.

Chapter 3 presents some original progress towards BoBW guarantees for lexicographic agents. Our main result is the *Uniform tailed Simultaneous Eating* algorithm, which computes a ex-ante $^6/_7$ -EF and ex-post (EFX+PO) allocation for lexicographic goods. We then go on to formally prove our result and also describe a weaker algorithm that guarantees $^1/_2$ ex-ante EF and ex-post (EFX+PO). For the chores setting, we provide an algorithm that gives ex-ante EF and ex-post EFX, but fails ex-post PO. We briefly discuss some of our alternative approaches and their limitations. In the final section, we show that by relaxing ex-ante EF to ex-ante PROP, it is possible to obtain ex-post (EFX + PO) for both goods and chores.

The Sections 3.1, 3.2, 3.3, 3.4 comprise entirely of original work. In all of the remaining sections, we present known results whose proofs have be rewritten or modified.

Chapter 1

Preliminaries

1.1 Basics of Computational Complexity

All the definitions and notations used here are taken from the textbook by Eva Tardos and Jon Kleinberg [22].

Running Times

Let $T(n)$ be the worst case running time of an algorithm of input size n. Given another non-negative function $f(n)$:

1. Asymptotic Upper bound: $T(n)$ is $O(f(n))$ if there is some positive constant c such that $T(n) \leq c \cdot f(n)$ for all sufficiently large n.
2. Asymptotic Lower bound: $T(n)$ is $\Omega(f(n))$ if there is some positive constant c such that $T(n) \geq c \cdot f(n)$ for all sufficiently large n.
3. Asymptotic Tight bound: $T(n)$ is $\Theta(f(n))$ if it is both $\Omega(f(n))$ and $O(f(n))$.

An algorithm is said to run in polynomial time if $T(n)$ is $O(f(n))$ for some polynomial $f(n)$.

Definition 1.1.1 (Reducibility)**.** *Consider two decision problems X and Y. If X is polynomial time reducible to Y ($X \leq_p Y$) then Y is atleast as hard as X.*

In other words, every instance of X can be converted to an equivalent instance of Y using a *reduction*: an algorithm that runs in polynomial time. If X can't be solved efficiently (that is, solved in polynomial time), so can't Y.

Definition 1.1.2 (P)**.** *The class of decision problems that can be solved in polynomial time.*

Definition 1.1.3 (NP)**.** *The class of decision problems whose solutions can be verified in polynomial time. In other words, it is the set of all such decision problems for which there exists a polynomial-time algorithm (known as an efficient certifier) that takes as input an instance of the problem, and a certificate for the problem instance (whose length is polynomial in the size of the problem instance), and verifies the certificate.*

Definition 1.1.4 (NP-Complete and NP-hard)**.** *A decision problem $\mathcal{P}$ is said to be* NP-complete *if (1) $\mathcal{P} \in$ NP and, (2) for any decision problem $P_1 \in$ NP, $P_1 \leq_p \mathcal{P}$. When the first condition is relaxed, $\mathcal{P}$ is said to be* NP-hard.

1.2 Fair Division

We concern ourselves with the basic definitions and notations in the literature of fair division of *indivisible* items, which would be used throughout this thesis.

Definition 1.2.1. *A fair division instance is a tuple $\langle N, M, \mathcal{V} \rangle$, where $N = \{1, ..., n\}$ of n agents, a set $M = \{1, ..., m\}$ of m indivisible items, and a valuation profile $\mathcal{V} = (v_1, .., v_n)$. Each valuation function $v_i : 2^M \longrightarrow \mathbb{R}$ maps every $S \subseteq M$ to a real valued $v_i(S)$, interpreted as the utility i derives from S.*

We shall refer to any $X \subseteq M$ as a 'bundle'. By convention, $v_i(\{\emptyset\}) = 0$ for all agents i.

Definition 1.2.2 (Allocation)**.** *A* ***integral*** *allocation $A = (A_1, A_2, \ldots, A_n)$ is an $n-$partition of M, where A_i is the bundle assigned to agent i. Allocation A is a complete allocation if $\bigcup_{i \in N} A_i = M$, and is termed partial if $\bigcup_{i \in N} A_i \subset M$.*

When the items are divisible, we define a fractional allocation as follows:

Definition 1.2.3. *A* ***fractional*** *allocation A is specified by a $N \times M$ matrix where $A_{i,g}$ is the fraction of item g assigned to agent i. Allocation A is a complete allocation if $\sum_{i \in N} A_{i,g} = 1$ for every $g \in M$, and is termed partial otherwise.*

In this thesis, an allocation automatically refers to a complete allocation unless explicitly mentioned otherwise.

In a *goods instance*, every item (called a *good*) in M is assigned a non-negative value by all agents. Informally, this means that every agent prefers receiving some good over nothing.

Conversely, in a *chores instance*, each item is assigned a non-positive value. Here, the items can be thought of as tasks or chores, where $v_i(c)$ is the negative utility (*disutility*) agent i incurs from c. Consequently, one can define *mixed* instances as well.

Valuation Functions

Several kinds of valuation functions are studied in the literature, to describe agents' preferences over bundles. For a goods instance, the valuation function $v(\cdot)$ is :

1. Monotone: Given bundles $X, Y \subseteq M$ such that $X \subseteq Y$, then $v(X) \leq v(Y)$.

2. Additive: For any $X \subseteq M$, $v(X) = \sum_{x \in X} v(x)$ i.e. the utility of a bundle is exactly the sum of the utilities of its elements.

3. Submodular: For any $X \subseteq Y \subseteq M$ and a good $g \in M \setminus Y$, $v(X \cup \{g\}) - v(X) \geq v(Y \cup \{g\}) - v(Y)$.

4. Sub-Additive: For any $X, Y \subseteq M$, it holds that $v(X \cup Y) \leq v(X) + v(Y)$

5. (XOS) Fractionally Subadditive: for any $X \subseteq M$ there exist valuations $v_1, v_2, \ldots, v_k$ such that:
$$v(X) = \max_{j \in [k]} v_j(X),$$
where each $v_j(X) = \sum_{x \in X} v_j(x)$ is an additive function.

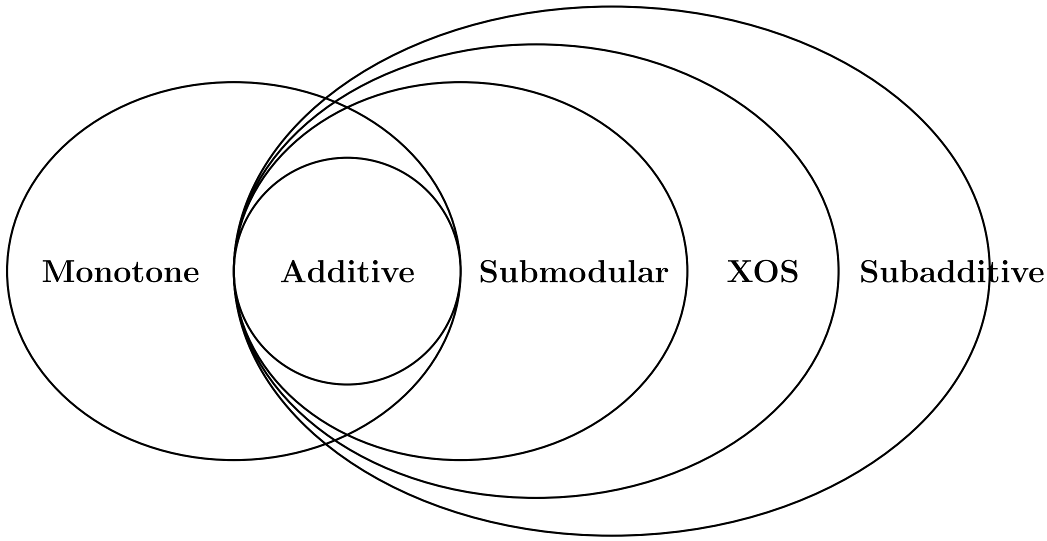

Figure 1.1: Relations between different valuation classes. Although only additivity implies monotonicity, most works dealing with beyond-additive valuations implicitly assume monotonicity as well.

Economic Efficiency of an Allocation

Definition 1.2.4 (Pareto Optimality)**.** *An allocation A Pareto dominates an allocation B if for all agents i, $v_i(A_i) \geq v_i(B_i)$ and for some agent k, $v_k(A_k) > v_k(B_k)$. An allocation A is Pareto Optimal (PO) if no other allocation Pareto dominates it.*

Definition 1.2.5 (fPO)**.** *An allocation X is **fractionally** Pareto Optimal (fPO) if there exists no fractional allocation that Pareto dominates X.*

Observe that fPO $\Rightarrow$ PO but PO $\not\Rightarrow$ fPO. Consider an instance with just 2 agents and 2 goods.

Goods	**Alice's Value**	**Bob's Value**
A	3	4
B	2	1

Table 1.1: PO $\not\Rightarrow$ fPO

In example 1.1, the allocation X where Alice gets A and Bob gets B is PO. However, the fractional allocation where Alice gets half of A and whole of B, and Bob gets half of A, dominates X. Thus, X is PO but not fPO.

Fairness Notions

There are multiple well studied notions to describe how *fair* an allocation is. We define some of these notions for the *goods-only* instance.

Definition 1.2.6 (PROP)**.** *An allocation X is **proportional** (PROP) if for all $i \in N$, $v_i(X_i) \geq \frac{v_i(M)}{n}$.*

Definition 1.2.7 (EF for goods)**.** *An allocation X is said to be **envy-free** (EF) if for every pair of agents $i, j \in N$, it holds that $v_i(X_i) \geq v_i(X_j)$*

Definition 1.2.8 (EF1 for goods)**.** *An allocation X is **envy-free upto 1 good** (EF1) if for every pair of agents $i, j \in N$, $\exists\ g \in X_j$ such that $v_i(X_i) \geq v_i(X_j \setminus \{g\})$*

Definition 1.2.9 (EFX for goods)**.** *An allocation X is **envy-free upto any good** (EFX) if for every pair of agents $i, j \in N$, for all $g \in X_j$, it holds that $v_i(X_i) \geq v_i(X_j \setminus \{g\})$*

We can also define an approximate notion of EF, where each agent compares their bundle with other agents after some scaling.

Definition 1.2.10 (α-EF for goods)**.** *An allocation X is said to be α-EF ($\alpha \in [0, 1]$) if for every pair of agents $i, j \in N$, it holds that $v_i(X_i) \geq \alpha \cdot v_i(X_j)$*

An allocation is simply referred to as EF when it is 1-EF. Similarly, we can define corresponding notions of α-EF1 and α-EFX.

In the chores setting, the fairness notions are defined a bit differently.

Definition 1.2.11 (EF1 for chores)**.** *An allocation X is **envy-free upto 1 chore** (EF1) if for every pair of agents $i, j \in N$, $\exists\ c \in X_i$ such that $v_i(X_i \setminus \{c\}) \geq v_i(X_j)$*

Definition 1.2.12 (EFX for chores)**.** *An allocation X is **envy-free upto any chore** (EFX) if for every pair of agents $i, j \in N$, for all $c \in X_i$, it holds that $v_i(X_i \setminus \{c\}) \geq v_i(X_j)$*

When defining approximate fairness (such as α -EF) for chores, note that $\alpha \geq 1$. Also, note that EF $\Rightarrow$ EFX $\Rightarrow$ EF1 $\Rightarrow$ . . . $\Rightarrow$ EFk (similar to EF1, we can define EFk for $k \in \mathbb{N}$, $k > 1$)

Randomness in Allocations

Definition 1.2.13 (Randomized allocations)**.** *A randomized allocation* $Z = (Z_1, Z_2, ..., Z_n)$ *is a random variable* Z *whose outcomes are deterministic allocations.* Z_i *is the random variable denoting the random bundle assigned to agent* i*.*

Definition 1.2.14 (Ex-Ante EF)**.** *A randomized allocation* Z *is **ex-ante*** EF *if no agent envies any other agent in expectation. Formally,* $\mathbb{E}[v_i(Z_i)] \geq \mathbb{E}[v_i(Z_j)]$*. For* $\alpha \in [0, 1]$*, allocation* Z *is ex-ante* α-EF *if* $\mathbb{E}[v_i(Z_i)] \geq \alpha \cdot \mathbb{E}[v_i(Z_j)]$*.*

Definition 1.2.15 (Ex-Post Fairness)**.** *Given a property* $\langle Q \rangle$*, a randomized allocation* Z *is said to be **ex-post*** $\langle Q \rangle$ *if every deterministic allocation in the support of* Z *satisfies* $\langle Q \rangle$*.*

Definition 1.2.16 (Stochastic Dominance)**.** *For non-negative random variables* X *and* Y*, we say* X *stochastically dominates* Y *(written as* $X \succeq_{sd} Y$*) if* $\Pr[X \geq t] \geq \Pr[Y \geq t]$ *for all* $t \geq 0$*.*

1.2.1 Notation

1. We shall write $X \succeq_i Y$ ($X \succ_i Y$) to denote that agent i weakly (strongly) prefers bundle X over bundle Y.
2. We shall denote $X \setminus \{g\}$ for bundle X and some good g as $X \setminus g$ or $X - g$ for convenience.
3. Similarly, $X \cup \{g\}$ is denoted as $X \cup g$ or $X + g$.

Chapter 2

Review: Background and Motivation

2.1 Fair Division of Indivisible Resources

Consider the discrete fair division setting $\langle N,\ M, \mathcal{V}\rangle$, where the goal is to allocate the items of M to agents in N in a *fair* and *efficient* manner. There are two predominant fairness notions: envy-freeness (EF) and proportionality (PROP), both of which have been well studied in the divisible setting as well. An allocation is said to be PROP if every agent receives their proportional share ($1/n$ of the total) -making it a *threshold* based fairness criteria. EF is a stronger, comparison based notion: every agent i values their own bundle (weakly) more than any other agent's bundle.

Proposition 2.1.1. *Let A be an allocation of an **additive** instance $\langle N, M, \mathcal{V}\rangle$. If A is* EF*, then A satisfies* PROP*. However, the converse is not true.*

Proof. $\implies$ Pick any agent i. If A is EF, then $v_i(A_i) \geq v_i(A_j)$ for all $j \in N$.

$$\therefore\ n \cdot v_i(A_i) \geq \sum_{j \in N} v_i(A_j) = v_i(M)$$

Thus, A satisfies PROP. We provide a counterexample to show why the converse does not hold.

	g_1	g_2	g_3
a	4	7	1
b	5	5	5
c	1	10	1

Table 2.1: The allocation $A_a = \{g_1\}$, $A_b = \{g_3\}$, $A_c = \{g_2\}$ is PROP but not EF

Consider the allocation where g_1 is given to a g_2 is given to c and g_3 to b. Clearly, it is PROP. However, a envies agent c. □

However, see that PROP and EF are equivalent notions when there are two agents. It is easy to see that EF and PROP need not always exist: When there is a single good but two agents, the agent who doesn't receive the good will envy the other. Furthermore, deciding whether an instance admits an EF allocation is NP-Complete, even for additive agents. We can show this via a reduction from the partition problem. This naturally motivates the study of relaxations of envy-freeness, such as EF1, EFX etc. While there are similar relaxations of PROP, we shall not cover them in this work.

2.1.1 Algorithms for EF1

The first relaxation we consider is *envy-freeness upto 1 good* (EF1) [5]. In simple terms, EF1 means that if an agent envies another, then the envy could be eliminated by removing exactly one good from the envied bundle. Unlike EF, EF1 always exists and can be computed in poly. time. There are several algorithms to compute EF1 allocations, the simplest of which is Round Robin: that gives EF1 when the agent valuations are additive.

Every agent, acc. to some ordering, picks their favourite good from the set of the remaining goods during their turns, until all goods are allocated. We can sort M in decreasing order of preferences in $O(m\,\log m)$ for every agent. Then, the algorithm runs in m steps each taking $O(1)$ time.

Algorithm 1: *Round Robin*

Input: An additive goods instance $\langle N, M, \mathcal{V} \rangle$

Output: A complete allocation A

$A \leftarrow (\emptyset, ..., \emptyset)$

Label the agents as 1, 2, ..., n

$M' \leftarrow M$

for j *from* 1 *to* m **do**

  $i \leftarrow j \mod n$

  $g \leftarrow \arg\max_{h \in M'} v_i(h)$

  $A_i \leftarrow A_i + g$

  $M' \leftarrow M' \setminus g$

return A

Theorem 2.1.1. *For additive instances, Round Robin returns an allocation A that is EF1.*

Proof. See that the algorithm has $T = \lceil m/n \rceil$ rounds, and the bundle of any 2 agents differ atmost by 1 good. Let g_{k_t} denote the good picked by agent k in round t.
Pick any $i, j \in N$. We shall look at i's envy towards j. If $i < j$ in the ordering, then: for each round t that i picks, $v_i(g_{i_t}) \geq v_i(g_{j_t})$. Since i goes before j, i picks atleast as many times as j and hence, i doesn't envy j.
If $i > j$ in the ordering, then: for each round $t \in [T-1]$, $v_i(g_{i_t}) \geq v_i(g_{j_{t+1}})$. Thus, summing this over all $t \in [T-1]$, $v_i(A_i) \geq v_i(A_i \setminus g_{i_1})$ This is exactly the condition for EF1. Hence, A is EF1. □

We shall see that EF1 can always be computed efficiently even when there is very little structure on agent valuations: monotonicity. Amusingly, the concept of EF1 was implicitly introduced by [8], but formally defined much later by Budish [5].

Description: At every step, we allocate a good to an unenvied agent. At some point in the process, if there is no unenvied agent, then we resolve a cycle in the envy graph of the partial allocation at that step. The envy graph $G_A = (N, E)$ is a directed graph where $(i, j) \in E \iff v_i(A_i) < v_i(A_j)$. Thus, an unenvied agent is a source vertex in G_A. Resolving a cycle C means that every agent receives the bundle they point to in C i.e the bundle that they envy in C.

Algorithm 2: *Envy Cycle Elimination Algorithm*

Input: A monotone goods instance $\langle N, M, \mathcal{V}\rangle$

Output: A complete allocation A

```
1  A ← (∅, ...., ∅)
2  M' ← M
3  while there exists an unallocated good g do
4      G_A ← envy graph of the partial allocation A
5      while no agent is unenvied do
6          C = (j_1, j_2 ... j_c, j_1) ← arbitrary cycle in G_A
7          p(j_k) ← the predecessor of agent j_k in C
8          for k in 1 to c do
9              A_{p(j_k)} ← A_{j_k}
10     i ← arbitrary unenvied agent
11     A_i ← A_i + g
12     M' ← M' \ g
13 return A
```

Theorem 2.1.2. *The Envy Cycle Elimination algorithm computes an allocation A that is EF1, in polynomial time.*

Proof. To see why A is EF1 : the first n goods are given to different agents. Thus, the partial allocation is trivially EF1. During any step of the algorithm, we can do one of two things to agent i: assign a good g or resolve some envy cycle it is part of.

When agent i is assigned a good g, it is because it is an unenvied agent. Thus, if any other agent envies i after this step, then simply removing g eliminates this envy. Thus, A continues to remain EF1 after this step. When an envy cycle is resolved, every agent in the cycle becomes strictly better off. The agents outside C are unaffected. Since no bundles are tampered with, but simply transferred completely to another agent, the envy any agent has towards any other agent is still EF1. Hence, the final allocation A is EF1.

Lastly, it needs to be shown that the algorithm terminates in polynomial time. Infact, it is not immediately obvious why the algorithm should even terminate - there could be an instance where we are stuck in step 5: resolving a cycle could create another cycle, and resolving that creates another, and this might loop around. However, we show that the number of edges in G_A strictly reduces with every cycle resolution. See that when a cycle C is resolved, every agent in the cycle becomes strictly better off and thus the edges in C are removed. Consider edges in $E(G_A) \setminus E(C)$: edges with endpoints that are agents not in C remain unaffected. Any outgoing edge from an agent in C either stays the same, or disappears. The cardinality of edges from outside into C remain the same (but the edges might themselves change) as the bundles are simply passed around and not modified. Therefore, $|E(G_A)|$ decreases by atleast 2 (as $|C| \geq 2$). After atmost $\frac{n(n-1)}{2}$ cycle resolutions, we find a source. Hence, we would need to resolve $O(mn^2)$ cycles. Detecting a cycle can be done in $O(n^2)$ time and thus, the algorithm terminates in poly. time. □

Note that this proof relies only on monotonicity—that an agent's utility weakly increases when they receive a good—rather than additivity.

If one wants to achieve Pareto Optimality along with EF1 the question becomes way harder. One might expect the *Round Robin algorithm*1 might achieve PO, but for $m > n$ it fails (Refer 2.3 for a counterexample). Barman *et al.* [40] show a method to compute such allocations in pseudo-polynomial time, but poly. time computation of EF1+PO allocations remains one of the largest open questions in fair division.

EF1 for Chores

When the items are all chores, we can think of them as items with disutilities (or negative valuations). For the chores scenario, an allocation A is EF1 means that if agent i envies another agent, then there exists some chore in i's own bundle whose removal eliminates the envy. See that for additive agents, *Round Robin* itself achieves EF1 for the chores case as well.

Theorem 2.1.3. *For any additive chores instance, Round Robin returns an EF1 allocation.*

Proof. Let c_{k_t} denote the good picked by agent k in round t.
Pick any $i, j \in N$. We shall look at i's envy towards j. If $i < j$ in the ordering, then: for each round t that i picks, $v_i(c_{i_t}) \geq v_i(c_{j_t})$. However, i could receive one more chore than j, in which case, removal of that extra chore eliminates i's potential envy.

If $i > j$ in the ordering, then: for each round $t \in [T-1]$, $v_i(g_{i_t}) \geq v_i(g_{j_{t+1}})$. See that the last round in which j picks is atleast the last round when i picks. Thus, summing this over all $t \in [T-1]$, $v_i(A_i \setminus c_{i_t}) \geq v_i(A_i \setminus c_{i_1}) \geq v_i(A_i)$.
Hence, A is EF1. □

Algorithm 3: *TT Cycle Elimination*

Input: A monotone chores instance $\langle N, M, \mathcal{V} \rangle$

Output: A complete allocation A

$A \leftarrow (\emptyset, ..., \emptyset)$
$M' \leftarrow M$
while *there exists an unallocated chore* g **do**
  $G_A \leftarrow$ envy graph of the partial allocation A
  while *no agent is unenvious* **do**
    $C = (j_1, j_2 ... j_c, j_1) \leftarrow$ a top trading envy cycle in G_A
    $p(j_k) \leftarrow$ the predecessor of agent j_k in C
    for k *in* 1 *to* c **do**
      $A_{p(j_k)} \leftarrow A_{j_k}$
  $i \leftarrow$ arbitrary unenvious agent
  $A_i \leftarrow A_i + c$
  $M' \leftarrow M' \setminus c$
return A

The *Envy cycle Algorithm* however, does not extend to monotone chores. This is because, when the items are chores, a greedy strategy would involve giving additional goods only to unenvious agents i.e a sink vertex in the envy graph. Sricharan *et al.* [24] gave the *Top Trading Cycle Elimination* where at every step, either a chore is assigned to a sink vertex, but now instead of arbitrarily resolving envy cycles, we resolve a Top Trading Envy Cycle - where each agent points to the bundle they envy the most. In the original algorithm, while an agent does become better off after cycle resolution, the receiver doesn't have any control over which chores he receives. An agent might exchange a bundle with a single heavy chore (where the rest are insignificant, preserving EF1 in the last step) for a better bundle but contains multiple chores with significant dis-utility, because of which the allocation would no longer remain EF1.

See that the absence of a source vertex, means that every vertex has an incoming edge $\implies$ there exists a cycle. Similarly, the absence of a sink vertex, means every vertex has an outward edge $\implies$ following the *most envied* outward edge gives us the top trading cycle required. The formal proof of correctness is omitted here.

2.1.2 Known Results and Challenges for EFX

An allocation is *envy-free upto any good* (EFX), meaning that if agent i envies some other agent, removing *any* good from the envied bundle eliminates the envy. One can also think of it as removing the "worst" good from the envied bundle. It is stronger than EF1, where *some* particular good is removed from the other bundle to eliminate envy.

However, unlike EF1, both the existence and efficient computation of EFX is still an unresolved open problem (barring a few special cases). For identical agents, [7] show that EFX always exists even for monotone valuations. We first see a straightforward method when the agents are identical and the valuations are additive. We initially sort all the goods in decreasing order of agent preferences. We then assign goods according to this order- at every step of the Algorithm, we assign the current good to the poorest agent. See that this is simply the *Envy Cycle Algorithm*, but instantiated with special order of goods.

Algorithm 4: *EFX for additive identical agents*

Input: A additive goods instance $\langle N, M, \mathcal{V} \rangle$ with all agents identical

Output: A complete allocation A

$A \leftarrow (\emptyset, ..., \emptyset)$

$g_1 \succ g_2 \succ \cdots \succ g_m$: label the goods such that g_k is the k^{th} favourite for every agent

for k *from* 1 *to* m*:* **do**

    $i \leftarrow \arg\min_j v_j(A_j)$ `/* Ties can be broken arbitrarily */`

    $A_i \leftarrow A_i + g_k$

return A

Theorem 2.1.4. *Algorithm 4 returns an* EFX *allocation.*

Proof. Fix any agent i. See that the first n goods must go to distinct agents. The partial allocation A is trivially EFX, since every agent is given a singleton bundle. We show that A

remains EFX throughout the algorithm. Let g be assigned to agent j. i is the poorest agent and thus is unenvied by anyone. Any envy after this step is due to g, and thus removing it eliminates envy. Therefore, $v_i(A_i) \geq v_i(A_j \setminus g)$.
Moreover, g is the *worst* good in A_i. Thus, $v_i(A_i) \geq v_i(A_j \setminus g) \geq v_i(A_j \setminus h)$ for any $h \in A_i$. Hence, A continues to remain EFX. □

The same arguments show why this algorithm works when it is an *ordered* instance: Every agent has the same ordinal preference over bundles, but might have different numerical valuations.

Another variation of the *Envy Cycle Algorithm* is where rather than assigning an arbitrary good, we assign the unenvied agent's top available good. It is easy to see that this returns an EFX allocation for $n = 2$, even for non-identical additive agents.

Proposition 2.1.2. *The above mentioned variant of Envy Cycle Algorithm returns a* $1/2$ *-EFX allocation for an additive instance with* $n > 2$ *(non-identical) agents.*

Proof. Fix some agent i. Once again, the first n goods go to distinct agents. The partial allocation A is EFX at this stage. If at some step, an agent j is assigned its favourite available good g, then it is because j was unenvied.

$$\begin{aligned} v_i(A_i) &\geq v_i(A_j \setminus g) \\ \text{Also, } v_i(A_i) &\geq v_i(g) \\ \text{Therefore, } v_i(A_i) &\geq \frac{v_i(A_j)}{2} \end{aligned}$$

Hence, A is $1/2$ -EFX.

□

In recent years, EFX has garnered significant attention. Bu *et al.* [10] showed efficient computation of EFX allocations when the marginals of goods are either 0 or 1. Mahara [42] demonstrated the existence of EFX allocations when agents belong to at most two types. The breakthrough result of Chaudhury *et al.* [16] proved that EFX exists for 3 agents, and this year, [17] extended this result to 3 types of agents. However, for the general case of $n \geq 4$ additive agents, the problem still remains elusive.

In the next section, we explore a subclass of additive functions - lexicographic valuations. These valuations are particularly compelling, because in this restricted domain, EFX are guaranteed to exist and can be computed in poly time.

2.2 Lexicographic Preferences

In the previous section, we introduced the notion of envy-freeness up to any good (EFX) as the strongest known relaxation of EF for indivisible resources. However, its existence and efficient computation remain open problems in general, with only a few special cases where EFX allocations are known to exist. [16, 17, 10, 25, 26]

In this section, we analyze the setting where agents have lexicographic valuations, a structured subclass of additive valuations. The material covered in this chapter is primarily based on the work of Hosseini *et al.* [9, 11]. We show that in this domain, EFX allocations always exist and are compatible with Pareto optimality (PO). Moreover, for any given instance, the set of all EFX+PO allocations can be characterized. We begin by examining the case of goods-only instances and later discuss the challenges that arise in chores and mixed instances, even within this restricted domain.

To understand this restricted setting, first we deviate from the usual framework of numerical preferences, and look at it from an ordinal point of view. Consider an underlying *preference profile* $\succ = (\succ_1, \succ_2 \cdots \succ_n)$ denoting the ordinal preferences of each agent $i \in N$ over the set of goods M. We shall assume that there are no ties between goods according to any agent, i.e the preference profile is strict. The agents' preferences over different bundles are given by the lexicographic extension of their preferences over individual goods.

For example, if agent j ranks the goods as $\succ_j$: $g_1 \succ g_2 \succ \cdots \succ g_m$, then it prefers any bundle with g_1 over any bundle that doesn't, subject to which it prefers a bundle that has g_2 over any bundle that without g_2, and so on.

Definition 2.2.1 (Lexicographic Preferences)**.** *Given any pair of bundles $A, B \subseteq M$, we have $A \succ_i B$ if and only if $\exists\, g \in A \setminus B$ such that $\{h \in B : h \succ_i g\} \subseteq A$.*

Consider an instance with 3 goods such that $\succ_i$: $g_a \succ g_b \succ g_c$. Then, the agent i prefers the bundles as : $\{g_a, g_b, g_c\} \succ \{g_a, g_b\} \succ \{g_a, g_c\} \succ \{g_a\} \succ \{g_b, g_c\} \succ \{g_b\} \succ \{g_c\} \succ \{\emptyset\}$

Many real-world scenarios involve lexicographic preferences. Say 3 people are dividing an inheritance consisting of a house, a bike, and a chair. Agents will usually prioritize any allocation in which they receive the house over one where they do not. Given this, they would then prefer an allocation where they receive the bike over one where they do not. Lexicographic preferences offer a succinct representation of preferences in combinatorial domains. The numerical valuations we consider are additive functions that remain consistent with lexicographic ordinal preferences. The utility derived by an agent from a good g is strictly more than the combined utility of all the goods that it prefers less than g. One natural instantiation of such valuations arises when the items' values follow a geometric progression, such as the powers of 2.

Theorem 2.2.1. *A lexicographic goods instance $\langle N, M, \mathcal{V} \rangle$, admits an EF allocation if and only if every agent's top-ranked good is distinct.*

Proof. $\Longrightarrow$ Let A be an allocation that is EF. Thus, for all agents i, j, $A_i \succ_i A_j$. This implies that each agent strictly prefers the highest-ranked good in their own bundle over the highest-ranked good in any other agent's bundle. This condition is satisfied only if every agent receives their top-ranked good. Thus, every agent's top-ranked good is distinct.
$\Longleftarrow$ Conversely, if every agent's top ranked good is different, then assign each agent its top ranked good. Any completion of this partial allocation remains EF.

□

While the problem of determining *envy-freeness* (EF) is NP-complete in the additive setting, it can be solved in polynomial time for this subclass. However, as we will see later, it remains NP-complete in the case of chores and mixed instances.

Remark 1. *Lexicographic preferences $\Longrightarrow$ every agent derives strictly positive utility from every good, meaning no good is valued at zero by any agent, as every agent prefers any good to the empty bundle. See that, on relaxing this assumption, it is possible to construct instances where no allocation simultaneously satisfies both EFX and PO.*

We present an example where no allocation is EFX + PO:

	g_1	g_2	g_3
a	2	1	0
b	2	0	1

Table 2.2: Agents a and b both have the same top good g_1, but have different approval sets on the remainder of goods.

In this Example 2.2, any PO allocation should assign g_2 to a and g_3 to b. Now, if g_1 is assigned to either of the 2 agents, EFX is violated, as the agent receiving g_1 will be envied and will have a bundle containing more than one good. Thus, no PO allocation is also simultaneously EFX.

Proposition 2.2.1. *In an instance where goods are valued at 0 by some agent, an allocation A is PO only if every good is assigned to some agent for which it has strictly positive utility.*

2.2.1 EFX and Pareto Optimality for Goods

Lemma 2.2.1. *For a lexicographic goods instance, an allocation A is* EFX $\iff$ *every envied agent in A gets exactly one good.*

Proof. $\implies$ Say A is EFX and i is an agent envied by some j. Thus, $A_i \succ_j A_i$, meaning, i has some good g that j strictly prefers over its favorite good in A_j. If $|A_i| > 1$, then there exists some $g' \neq g \in A_i$. See that j would continue to envy i even after removing g' from A_i, thus violating EFX.

$\impliedby$ If every *envied* agent has exactly 1 good, then trivially A is EFX. □

Algorithm 5:

Input: A lexicographic goods instance $\langle N, M, \mathcal{V} \rangle$
Output: A complete allocation A

```
A ← (∅, ...., ∅)
Initialize M' ← M for i in 1 to n do
    g_i ← arg max_{h∈M'} v_i(h)
    A_i ← A_i + g_i M' ← M \ g_i
A_n ← A_n ∪ M'
return A
```

Theorem 2.2.2. *Algorithm 5 returns an allocation A that is* EFX + PO *in poly. time.*

Proof. Let us denote the last agent as i. Observe that i will not be envied by anyone else, since everyone else picks their good before i. Since every subsequent goods assigned to i are also less preferred by other agents to what they have. Thus, because the valuations are lexicographic, every agent prefers their singleton over the i's bundle. Thus, every envied agent (if any at all) is assigned a singleton. Thus, A is EFX.
To show PO, we directly invoke Proposition 2.2.2. □

Definition 2.2.2 (Sequencibility). *A picking sequence is an ordered tuple $\langle a_1, a_2, ..., a_k \rangle$ where each position denotes an agent. The sequence specifies an order in which agents take turns to pick their most preferred available item. An allocation A is sequencible $\iff$ A can be simulated via a picking sequence.*

Proposition 2.2.2. *For a lexicographic goods instance, an allocation A is* PO $\iff$ *A is a sequencible allocation.*

Proof. $\implies$ Let A be a PO. Consider a directed graph where the vertex set is N (i, j) is an edge $\iff$ j receives i's top good. If no agent receives their top good, then every vertex must have at least one outgoing edge, implying the existence of a cycle. Resolving this cycle by cyclicly exchanging the goods in the opposite direction, improves the utility of every agent in the cycle, while all other agents remain unaffected. This new allocation Pareto dominates A, contradicting the assumption that A was PO. Thus, at least one agent must receive their top good. We append this agent to the tail of the picking sequence, and recursively do this procedure for the remaining goods. Note that this is a valid procedure, at every step of this procedure, the *sub-allocation* remains PO.

$\impliedby$ Let A be sequencible, but not PO. Let B be another allocation that Pareto dominates A. Then, there must exist some agent i such that $A_i \setminus B_i \neq \emptyset$. WLOG, let i be the first such agent in our picking sequence. Let $g \in A_i \setminus B_i$ be the good picked by i in the allocation A.

Since $B_i \succ_i A_i$, there exists $g' \in B_i \setminus A_i$ such that $g' \succ_i g$ (by lexicographic property). By our choice of i, all agents before i in the picking sequence have the same bundle in both B and A. Thus, during i's turn to pick, both g and g' are available. But this is a contradiction, as i picked g in A. Thus, A is Pareto Optimal.

$\square$

Note that in Algorithm 5, the allocation A is PO as it can be simulated via the picking sequence $1,\ 2,\ 3\,, ...,\ \underbrace{n,\ n\, ..., n}_{m-n+1 \text{ times}}$.

However, when the valuations are relaxed to the additive domain, the equivalence doesn't hold both ways anymore. While PO allocations are still sequencible (for this direction, in the previous proof, we didn't invoke lexicographic property anywhere), all sequencible allocations need not satisfy PO.

	g_1	g_2	g_3	g_4	g_5
A	10	6	3	2	1
B	5	4	3	2	1

Table 2.3: Round Robin fails PO

We illustrate with Example 2.3. This is an additive instance, where the round robin algorithm fails PO. Let X be the allocation returned by round robin where $X_A = \{g_1,\ g_3,\ g_5\}$ and $X_b = \{g_2,\ g_4\}$. Consider another allocation Y where $Y_A = \{g_1,\ g_2\}$ and $Y_B = \{g_3,\ g_4,\ g_5\}$. Agent A is strictly better off in Y (it's utility increases from 14 to 16), while agent B remains the same (utility is 6 under both allocations). Hence, X is Pareto dominated by Y.

We now exploit Proposition 2.2.2 and Lemma 2.2.1 to give an algorithm that efficiently characterizes all EFX + PO allocations.

Each algorithm in the family (Algorithm 6) is parameterized by an apriori permutation σ and a picking sequence π which is determined after Phase 1. Phase 1 involves assigning exactly 1 good to each agent. Any agent who receives a good in Phase 2 is unenvied. There must be atleast 1 unenvied agent ($U \neq \emptyset$): in particular, the last agent according to σ is an unenvied agent, since every other agent prefers the good they picked in Phase 1 over the good that the last agent received. Furthermore, note that Algorithm 5 is a special case of Algorithm 6, where the last agent simply gets all the remaining goods. Infact, we now show that not only every algorithm in this family returns an EFX + PO allocation, but given some instance, *every* EFX + PO allocation can be computed by some algorithm in this family.

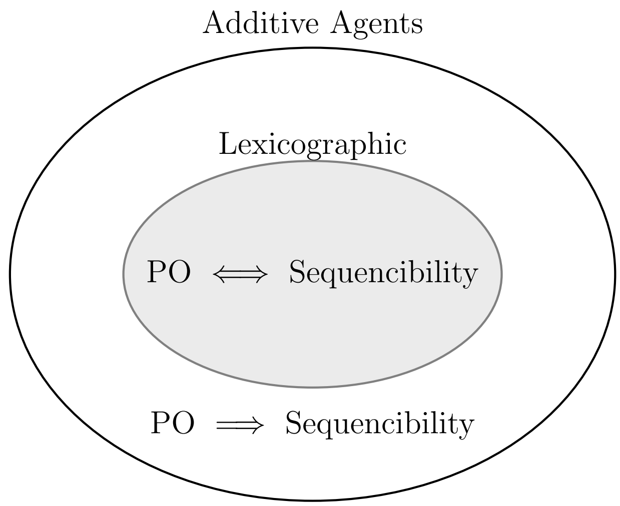

Figure 2.1: Figure 2.1 shows the relation between PO and Sequencibility for Additive and Lexicographic valuations.

:

Algorithm 6: EFX+PO Algorithm

Input: A lexicographic goods instance $\langle N, M, \mathcal{V} \rangle$

Parameter: A permutation $\sigma : N \to N$

Output: A complete allocation A

`/* Phase 1: construct a matching */`

$M \leftarrow M'$ **for** *each agent i in the order of σ* **do**

- $g_i \leftarrow \arg\max_{h \in M'} v_i(h)$
- $A_i \leftarrow A_i + g_i$
- $M' \leftarrow M' \setminus g_i$

`/* Phase 2: Allocate remaining goods to unenvied agents */`

$U \leftarrow \{k \in N : k$ is not envied by any agent after Phase 1$\}$

$\pi \leftarrow$ arbitrary picking sequence of length $m - n$ comprising agents in U

for *each agent j in the order of π* **do**

- $g_j \leftarrow \arg\max_{h \in M'} v_i(h)$
- $A_j \leftarrow A_j + g_j$
- $M' \leftarrow M' \setminus g_j$

return A

Theorem 2.2.3. *For any permutation $\sigma : N \to N$, the allocation computed by Algorithm 6 satisfies EFX and PO. Conversely, any EFX + PO allocation can be computed by Algorithm 6 for some instantiation of σ.*

Proof. $\Longrightarrow$ We will first show that Algorithm 6 returns an EFX + PO allocation.
To show PO, see that A is sequencible - the picking sequence is simply $\langle \sigma, \pi \rangle$. (using 2.2.2) At the end of Phase 1, the partial allocation is trivially EFX. In Phase 2, only an unenvied agent receives additional goods. No new envy is created in Phase 2, as every agent has picked their most preferred available good in Phase 1. If an agent i remains unenvied at the end of Phase 1, then every other agent strictly prefers their allocated good over i's selection. Anymore goods that i receives in Phase 2 are strictly less preferred, and because of lexicographic preferences, i remains unenvied. Thus, A is EFX.
$\Longleftarrow$ Let B be a given EFX + PO allocation. Any PO allocation can be induced by a picking sequence, and therefore any EFX + PO allocation is also sequencible. We claim that the first n positions of $\mathcal{P}$ (some sequence that simulates B) belongs to different agents, and the following $m - n$ positions are of agents that are unenvied. If we can show this, we have our parameters to instantiate Algorithm 6 with, such that $B = A$.
We partition N based on agents' allocations in B: let S be the set of agents receiving a singleton bundle under B, and $N \setminus S$ be the remaining agents. We claim that some agent in S must receive their most preferred good in M. To see this, we use a similar argument as in Proposition 2.2.2. If no agent in S receives their top good, then a cycle must exist *within* S. This is because the top good of any agent in S if assigned to some agent j in $N \setminus S$, would create envy towards j. However, since B is an EFX allocation, an envied agent can only receive a singleton bundle. Resolving this cycle (within S) would lead to a Pareto improvement over B, contradicting its Pareto optimality. Thus, some agent in S receives their most preferred good in M, and we push this agent to the first position in $\mathcal{P}$.
We recursively apply this argument, establishing that the first $|S| \leq n$ positions of $\mathcal{P}$ are distinct. It remains to show that the next $n - |S|$ positions are also distinct. Since all remaining agents belong to $N \setminus S$, they are unenvied by all agents, and in particular, by other agents in $N \setminus S$ as well. Therefore, each agent in $N \setminus S$ receives their most preferred good from the remaining set of goods, meaning any permutation of these agents simulates B, forming the next $n - |S|$ positions of $\mathcal{P}$.

Hence, the first n-prefix of $\mathcal{P}$ defines our required σ, while the $m - n$ suffix of $\mathcal{P}$ is our

π, which we use to instantiate Algorithm 6 to obtain A.

□

While there are strong existence and computational results for EFX and PO, extending these results to fPO while preserving EFX is not always possible.

Proposition 2.2.3. *There exists a lexicographic goods instance that admits no allocation that is* EFX+ fPO.

Proof. Consider an instance with 3 goods and 2 agents and valuations as follows:

	a	b	c
v_1	1	$(1+\epsilon)^{10}$	$1+\epsilon$
v_2	$1+\epsilon$	$(1+\epsilon)^{10}$	1

Table 2.4: Both agents v_1 and v_2 have the same top good b. However, v_1 prefers a over c (vice versa for v_2)

For $1/9 < \epsilon < 1$, the instance is lexicographic. There are only 2 EFX allocations: either only v_1 or v_2 gets item b and the remaining two go to the other agents. WLOG, consider the EFX allocation A where $A_{v_1} = \{b\}, A_{v_2} = \{a, c\}$. Here, $v_1(A_{v_1}) = (1+\epsilon)^{10}$ and $v_2(A_{v_2}) = 2+\epsilon$. Now, consider the fractional allocation:

	a	b	c
v_1	0	$1-(1+\epsilon)^{-9}$	1
v_2	1	$(1+\epsilon)^{-9}$	0

Table 2.5: Both agents are allocated none of their least favourite goods. One agent is given a very tiny portion of b while the other gets the rest of b. Good a is completely given to v_2, and c to v_1.

Now, in this new fractional allocation B, $v_1(B_{v_1}) = (1+\epsilon)^{10}$ and $v_2(A_{v_2}) = 2+2\epsilon$. Thus, B strictly improves v_2 while v_1 remains the same in utility, $\implies$ A is not fPO. Similarly, we can construct a fractional allocation that dominates the other EFX allocation (by symmetry). Thus, the instance admits no EFX + fPO allocations. □

2.2.2 The Chores Problem

We now shift our focus to the setting where the valuations are still lexicographic, but every item is a *bad* (or) *chore*. Every agent has a strict preference profile (similar to the *goods* case) over the set of chores. Informally, if agent i has a preference $\succ_i$: $a \succ b \succ \cdots \succ l \succ m$, then it prefers any bundle *without* m over any bundle containing m, subject to which it prefers a bundle *without* l over any bundle that contains l, and so on. We call the highest ranked chore (here it is a) as its '*least hated*' chore. We define it formally as:

Definition 2.2.3. *Given any pair of bundles $A, B \subseteq M$, bundle $A \succ_i B \iff$ there exists $c^* \in B \setminus A$ such that $\{c \in A : c^* \succ_i c\} \subseteq B$*

So, if there are 3 chores such that $\succ_i$: $c_a \succ c_b \succ c_c$, then the ranking over bundles is given by $\{\emptyset\} \succ \{c_a\} \succ \{c_b\} \succ \{c_a, c_b\} \succ \{c_c\} \succ \{c_a, c_c\} \succ \{c_b, c_c\} \succ \{c_a, c_b, c_c\}$. The cardinal valuations we consider are additive functions consistent with these lexiographic ordinal preferences. They are very similar to the goods setting, but take negative values. We again assume that no chore is valued at 0 by any agent, for the same reasons as in the previous case.

Theorem 2.2.4. *Determining whether a chores instance with lexicographic preferences admits an envy-free (*EF*) allocation is* NP-complete*.*

Let us compare the chores and the goods settings. Recall that deciding EF in goods is poly. time computable. An allocation is EF *if and only if* every agent receives their top ranked good. Furthermore, any *completion* of a partial allocation with this property remains EF.

However, in the chores setting, an allocation A is EF only if, for every agent, their *worst* (least preferred) chore in their own bundle is strictly preferred over their *worst* chore in every other agent's bundles. Since envy relations in this setting are dictated by the worst chore assigned to each agent, a partial allocation that is EF does not necessarily remain EF upon completion. An agent i who is initially unenvious in a partial allocation may receive additional, more undesirable chores, thereby introducing envy at a later stage. While we omit the proof of Theorem 2.2.4, we note that it follows from a reduction from SAT [11]. Moving on to EFX, similar to the goods setting, we make a very simple observation about EFX allocation in chores.

Lemma 2.2.2. *An allocation A is EFX if and only if each envious agent in A gets exactly one chore.*

Proof. $\implies$ Say A is EFX and i is an envious agent. Let j be an agent that i envies. Thus, i has a chore c that it strictly hates over any chore in A_j. If $|A_i| > 1$, i would continue to envy j even after removal of some chore $c^* \neq c \in A_i$, thereby violating EFX.
$\impliedby$ If every envious agent receives a single chore, then EFX is trivially satisfied, as removing that chore would eliminate envy. ☐

In the previous section, we saw a family of algorithms to characterize all EFX + PO allocations in *goods-only* instances. Although EFX + PO allocations always exist in *chores-only* instances (Algorithm 7), an efficient characterization of all such allocations is still not known.

Algorithm 7:

Input: A lexicographic chores instance $\langle N, M, \mathcal{V} \rangle$

Output: A complete allocation A

$A \leftarrow (\emptyset, ..., \emptyset)$
Let us denote the agents as 1, 2, 3, .., n
$C = \{c_1, c_2, ..., c_{m-n+1}\} \leftarrow$ agent i's least hated $m - n + 1$ chores
$A_i \leftarrow A_i \cup C$ **for** *each agent j from* 2 *to* n **do**
  $c_j \leftarrow j$'s least hated *available* chore
  $A_j \leftarrow A_j + c_j$
return A

Theorem 2.2.5. *Algorithm 7 returns an allocation A that is* EFX + PO.

Proof. Fix any 2 agents i, j. Let's analyze i's envy towards j. If i is an agent that received a singleton (i.e $i \neq$ agent 1), then i's envy towards j is trivially EFX. If i is agent 1, because every other agent picks after i, i strictly dislikes the chore any other agent has over any chore in its own bundle, and therefore by lexicographic property, i is unenvious. Thus, A is EFX. To show PO, see that A is a sequencible allocation. Even in the chores setting, we shall show that sequencible allocations fully characterize the set of PO allocations. (Proposition 2.2.4) ☐

Proposition 2.2.4. *For a lexicographic chores instance, an allocation A is PO $\iff$ A is a sequencible allocation.*

Proof. $\implies$ The proof follows directly from the proof in the goods case (Proposition 2.2.2).

$\impliedby$ Let A be a sequencible allocation but not PO. Let B be some allocation that Pareto dominates A. Let $N' = \{i \in N : B_i \oplus A_i \neq \emptyset\}$ $N' \neq \emptyset$ as atleast one agent becomes strictly better off in B. For some $i \in N'$, since $B_i \succ_i A_i$, there exists atleast one chore $c_i \in A_i \setminus B_i$ such that c is strictly disliked over any chore in $B_i \setminus A_i$. WLOG, let us define c_i be the worst (least-preferred) such chore in $A_i \setminus B_i$ for every agent in N'.
Let c_i be assigned to some agent $j \neq i$ in allocation B ($\therefore c_i \in B_j \setminus A_j$). Thus $j \in N'$ as well, and j must also be strictly improved in B. Let us define $c_j \in A_j \setminus B_j$, note that j strictly prefers c_i over c_j.
Consider a graph with vertex set N' and (i, j) is a directed edge $\iff$ $c_i \in B_j$. Since for every vertex has an outgoing edge, there must exist a cycle. Let C be one such cycle involving k agents. We shall denote these k agents as $1, 2, ..., k$ such that for each $i \in [k]$, there is a directed edge $(i, i+1\,(mod\,k))$.
Recall that A is sequencible, and define p_i to be the position that i picked the chore c_i while simulating A. Note that agent $i+1\,(mod\,k)$ prefers c_i to $c_{i+1\,(mod\,k)}$, but didn't receive c_i in A, $\implies$ $p_i < p_{i+1\,(mod\,k)}$. But because they lie on a cycle, this means that $p_1 < p_2 < p_3 ... < p_k < p_1$ which is a contradiction. □

Thus, while there is a sharp contrast between computational results for fairness guarantees between the chores and the goods setting, Pareto Optimality is preserved across both.

2.2.3 A short note on Mixed Instances

A *mixed* instance is a generalization of the *chores-only* and *goods-only* settings, where every item could be either a good or a chore. One can also define an *objective* mixed instance, where each item is treated a good or as a chore by all agents. To understand lexicographic preferences over bundles, let us use an example. Let according to agent i, $\lhd_i : g_1 \lhd c_2 \lhd g_3$. Informally, this means that i prefers a bundle that contains g_1 over any bundle without it, subject to which, it prefers any bundle *without* c_2 over one that does, subject to which it prefers a bundle containing g_3 over one that doesn't.

Definition 2.2.4 (Lexicographic Mixed Instances)**.** *Given a pair of bundles* $X, Y \subseteq M$*, we have* $X \succ_i Y \iff$ *either (i)* there exists good $g \in X \setminus Y$ *such that* $\{$all goods $g' \in Y : g' \lhd_i g\} \subseteq X$ *or (ii) there exists a chore* $c \in Y \setminus X$ *such that* $\{$all chores $c' \in X : c \lhd c'\} \subseteq Y$

Recall that determining whether a lexicographic instance admits an EF allocation is NP-complete in the case of chores. By extension, the problem remains NP-complete in the mixed setting as well. However, a more striking observation is that positive fairness results in goods-only and chores-only models do not carry over to the mixed setting. In fact, EFX allocations may fail to exist even for *objective* mixed instances. Furthermore, deciding whether an objective mixed instance admits an EFX allocation is NP-complete [43]. However, if you enforce every agent's top item to be a good, an EFX+PO allocation always exists and can be computed efficiently.

Furthermore, we find that sequencibility, which characterizes PO allocations in the goods-only and chores-only domains, fails to do so in the mixed setting.

Proposition 2.2.5. *For a lexicographic mixed instance,* PO *implies sequencibility, but the converse does not hold true.*

Proof. The proof of PO $\implies$ sequencibility follows similar to the arguments used in Proposition 2.2.2 and 2.2.4. We now show a counter-example for the converse.

$$1 : g_1 \lhd g_2 \lhd c_3$$

$$2 : c_3 \lhd g_1 \lhd g_2$$

	g_1	g_2	c_3
Agent 1	100	10	-1
Agent 2	10	1	-100

Table 2.6:

Let A be the allocation simulated by the picking sequence $\langle 1, 2, 2\rangle$. $A_1 = \{g_1\}$ and $A_2 = \{g_2, c_3\}$. But, the allocation $B_1 = \{g_1, g_2, c_3\}$ and $B_2 = \emptyset$ dominates A. $\square$

2.3 The Best of Both Worlds framework

Consider a scenario where a set of M indivisble goods has to be repeatedly assigned to a set of N agents in a periodic fashion. One might employ the Round Robin Algorithm, to give a sufficiently desirable fair allocation. Say, the algorithm, the agent ordering used is σ. While every allocation is in-fact *fair* (EF1), the first agent in σ could gain an unfair advantage (over repeated allocation) over agents after it, and in particular over the last agent. Deterministic allocations can be used to disadvantage certain agents, thus motivating the need for randomization instead.

The *randomized allocation* Z can be thought of as a probability distribution over integral deterministic allocations. Formally, Z is specified by a set of q ordered pairs $\{(p_k, A^k)\}_{k\in[q]}$, where, for every $k \in [q]$, A^k is an integral allocation implemented with probability $p_k \in [0, 1]$, and $\sum_{k=1}^{q} p_k = 1$. Z is naturally associated with a *fractional* allocation $Z^f = \sum_k p_k \cdot A^k$ where $Z^f_{i,g}$ is the marginal probability that agent i receives good g.

Two perspectives arise when considering fairness in randomized allocations: *ex-ante* and *ex-post* fairness. A randomized allocation Z is *ex-ante* envy-free (EF) if no agent envies any other agent in expectation. For some property $\langle Q\rangle$, Z is *ex-post* $\langle Q\rangle$ if every integral allocation in the support of Z satisfies property $\langle Q\rangle$. Achieving *ex-ante* fairness is straightforward - one could simply assign all the goods to an agent chosen uniformly at random. However, this trivial strategy induces a lot of *ex-post* envy, as a single agent receieves all the goods and the others receive none. Since EF does not always exist in the indivisible setting, one can construct instances where no randomized allocation satisfies *ex-post* EF.

However, recall that, unlike EF, EF1 always exists for monotone valuations. This motivates a very natural question: can we retain a randomized allocation that is ex-post EF1 and simultaneously obtain exact EF ex-ante? If yes, then what it means is, given any instance, we can always randomize over a set of EF1 allocations such that the resulting allocation is ex-ante EF. Aziz *et al.* [13] answer this question positively, proving that when agents have additive valuations, one can efficiently compute a randomized allocation that is simultaneously ex-ante EF and ex-post EF1. This result has since sparked significant interest in studying simultaneous fairness guarantees [21, 20, 29, 30], an approach now widely known as the "Best of Both Worlds" framework.

In this section, we first examine a natural approach that fails achieve ex-ante EF and ex-post EF1. We then introduce two powerful tools used to study fractional allocations, that become of immediate relevance to our question. Finally, we present the algorithm from [13] and conclude with a discussion that sets the stage for the next chapter.

2.3.1 The Simultaneous Eating Mechanism

A natural approach to achieve ex-ante EF and ex-post EF1 is to identify a set of EF1 allocations and then randomize over them. Round Robin algorithm returns an allocation that is EF1, and one might hope that uniformly randomizing the agent ordering (among $n!$ permutations) might achieve ex-ante EF. Let's term this algorithm as *Randomized Round Robin*.

However, the intuition behind why this might fail is as follows: for two agents, i and j, while the probability of i going before or after j is the same, the deficit in utility accumulated by i in the latter case could still outweigh the advantage gained in the former.

Proposition 2.3.1. *There exists an additive instance where Randomized Round Robin fails ex-ante EF.*

Proof. Consider an instance with 3 agents and 3 goods as follows:

	g_1	g_2	g_3
a	$1+\epsilon$	1	$1-4\epsilon$
b	$1-4\epsilon$	$1+\epsilon$	1
c	$1+\epsilon$	$1-4\epsilon$	1

Table 2.7: *Randomized Robin* is not ex-ante EF

There are $3! = 6$ permutations. Let $A = \frac{1}{6}(A^1 + A^2 + A^3 + A^4 + A^5 + A^6)$ denote the fractional allocation implemented by *Randomized Round Robin*, where A^k is the EF1 allocation from instantiating round robin with the k^{th} permutation.

$\mathbb{E}[v_a(A_a)] = \dfrac{3v_a(g_1) + v_a(g_2) + 2v_a(g_3)}{6} = \dfrac{6-5\epsilon}{6}$ while $\mathbb{E}[v_a(A_b)] = \dfrac{5v_a(g_2) + v_a(g_3)}{6} = \dfrac{6-4\epsilon}{6}$. Thus, agent a envies agent b in expectation. □

However, *Randomized Round Robin* satisfies the weaker notion of ex-ante *proportionality* (PROP) .

Theorem 2.3.1. *For an additive instance, Randomized Round Robin returns a randomized allocation that is ex-ante* PROP+ *ex-post* EF1*.*

Proof. We only need to show the ex-ante guarantees, since every realization of this algorithm is trivially EF1. Let us label the goods as g_1, g_2..., g_m as per agent i's preferences, where g_1 is the most preferred good. Say i's position in some permutation is k for $1 \leq k \leq n$. If $m = n$, then i receives a good weakly better than g_k. For $m > n$,

$$v_i(A_i) \geq \sum_{\lambda=0}^{m/n} v_i(g_{\lambda n+k})$$

Thus, for the randomized allocation:

$$\mathbb{E}[v_i(A_i)]] \geq \sum_{k=0}^{n} \frac{1}{n} \left(\sum_{\lambda=0}^{m/n} v_i(g_{\lambda n+k}) \right) = \frac{v_i(M)}{n} \tag{2.1}$$

For the case where $m = \lfloor m/n \rfloor + r$ where $r > 0$, the exact same argument will work, where we sum over two cases separately : one where $k > r$ and $k < r$. This gives us the required result. ☐

Instead of starting with a method that guarantees ex-post EF1 and modifying it to achieve ex-ante EF, let us do the opposite way : We shall (*efficiently*) compute a fractional allocation that is EF, and then implement it over a set of EF1 integral allocations. To this end, we introduce the Simultaneous Eating mechanism [14], which offers strong guarantees of both fairness and efficiency.

Description of Simultaneous Eating (SE) : In this mechanism, let us treat each item as 1 unit of a divisible resource. Every agent is assigned an *eating speed*, which quantifies the rate of instant consumption of goods at any time t. In the standard version, all agents eat at the constant rate of 1 good per unit time. The algorithm works as follows: All agents start at time $t = 0$ and eat from their favourite available goods. Ties between goods can be settled arbitrarily. When a good is exhausted by a subset of agents, each of those agents proceed to eat their next favorite good that is available. The algorithm terminates when all goods have been eaten, and each agent is allocated the fraction of goods they have consumed. For example, if agent i is allocated 0.2 fraction of good g, then we interpret it as good g being allocated to agent i with a marginal probability of 0.2.

See that the standard protocol will take m/n time units to terminate. Although the mechanism involves agents eating in a continuous fashion, the protocol can be implemented in time $O(mn)$, by discretizing the time interval $[0, m/n]$, considering only the time points where an agent stops consuming a certain good and moves onto another [31].

A useful property of this protocol is that it computes a fractional allocation that is EF i.e the corresponding randomized allocation is ex-ante EF for additive agents. Furthermore, observe that the SE protocol only uses the ordinal preferences of agents, and thus, satisfies a stronger property called *Stochastic Dominance-envy-free* (SD-EF).

Definition 2.3.1 (SD-preference)**.** *Given fractional allocations X, Y for an additive instance, we say agent i* SD-prefers *X_i to Y_i, written as $X_i \succeq_i^{SD} Y_i$, if for every good $g \in M$, $\sum_{h\,:\,h \succeq_i g} X_{i,h} \geq \sum_{h\,:\,h \succeq_i g} Y_{i,h}$*

See that the above definition (in the additive setting) is equivalent to saying that the utility i receives under allocation X stochastically dominates the utility it receives under Y i.e $v_i(X_i) \succeq_{SD} v_i(Y_i)$.

Definition 2.3.2 (SD-EF)**.** *A fractional allocation X is* SD*-envy free (*SD-EF*) if for every pair of agents $i, j \in N$, $v_i(X_i) \succeq_{SD} v_i(X_j)$.*

In other words, if a fractional allocation X is SD-EF, then for any agent i, on fixing any prefix of goods (top k goods acc. to i, for some k), i weakly prefers its own bundle (restricted to this prefix) to any other agent's bundle (restricted to the prefix). The corresponding randomized allocation is said to be ex-ante SD-EF. We shall formally prove this.

Theorem 2.3.2. *Let X be the fractional allocation returned by the* Simultaneous Eating (SE) *mechanism. Then X is* SD-EF *and consequently, it is also* EF*.*

Proof. Intuitively, this claim holds because, at any given time in the mechanism, each agent consumes from its most preferred available good, and all agents consume at an identical rate. Consequently, over any infinitesimal time interval, no agent "falls behind" compared to another.

Consider the randomized allocation that the fractional allocation X underlies. Fix some agent i. It suffices to show that the random variable $v_i(X_i)$ stochastically dominates $v_i(X_j)$ for every $j \neq i \in N$. That is,

$$\forall\ j \neq i,\ \Pr[v_i(X_i) \geq t] \geq \Pr[v_i(X_j) \geq t] \text{ for any } t \geq 0$$

For all $t > 0$, define $G_t = \{g \in M : v_i(g) \geq t\}$. Observe that $\Pr[v_i(X_i) \geq t] = \sum_{g \in G_t} X_{i,g}$ and $\Pr[v_i(X_j) \geq t] = \sum_{g \in G_t} X_{j,g}$. Suppose, there exists some t such that $t_i = \sum_{g \in G_t} X_{i,g} < \sum_{g \in G_t} X_{j,g} = t_j$. This means that after t_1 time units, agent i started consuming a good it values less than t. However, since agent j was still consuming some item from G_t between times t_i and t_j, there is atleast one good in G_t that was not fully consumed after time t_i. This is a contradiction, as agent i switched to its next most preferred available good at t_i, which is valued at less than t. Thus, $\Pr[v_i(X_i) \geq t] \ \geq\ \Pr[v_i(X_j) \geq t]\ \forall\ t \geq 0$.
Consequently,

$$\mathbb{E}[v_i(X_i)] = \int_0^\infty \Pr[v_i(X_i) \geq t]\, dt \geq \int_0^\infty \Pr[v_i(X_j) \geq t] = \mathbb{E}[v_i(X_j)]$$

Thus, the fractional allocation X is SD-EF and therefore, EF. $\square$

The SE mechanism is frequently used a subroutine to construct partial allocations with desirable guarantees [20]. If the SE mechanism is let to run for $t \leq {}^m/_n$ time units, then we get a fractional partial allocation such that $\sum_{g \in M} X_{i,g} = t$ for every $i \in N$. Consider the special case where $t = 1$.

Definition 2.3.3. *[20] A partial allocation $X = (X_1, X_2..., X_n)$ is* weakly separated *if for every agent i, $v_i(X_i) \geq v_i(w)$ for any unallocated good $w \in M \setminus \bigcup_{i \in N} X_i$.*

Definition 2.3.4. *[20] Given a randomized partial allocation $X = (X_1, X_2, ..., X_n)$ we define the r.v $U = M \setminus \bigcup_{i \in N} X_i$. X is* strongly separated *if, for any integral allocation Y in the support of X, $v_i(Y_i) \geq v_i(w)$ for all $i \in N$ and every good w such that* $\Pr[w \in U] > 0$.

Observe that Strong Separation $\implies$ Weak Separation.

Theorem 2.3.3. *Let X be the fractional partial allocation computed by running the* SE mechanism *for $t = 1$ time unit. Then X is strongly separated.*

Proof. Let's fix any agent i. For all goods g, $\Pr[g \in U] = 1 - \Pr[g \in X_k$ for some $k \in N] = 1 - \sum_{k \in N} X_{k,g}$

Thus, $\Pr[g \in U] > 0 \iff \sum_{k \in N} X_{k,g} < 1$. In other words, for g to be unallocated in some realization of X, g must be a good **not** completely consumed during SE. Now, any good h that agent i receives under X, is a good i ate a positive fraction of. Clearly $v_i(h) \geq v_i(g)$, or otherwise i would have begun eating g before h and since g is not completely consumed, g would have been the final good i ate from. □

Theorem 2.3.4. *Let X be the fractional allocation returned by the* SE mechanism. *When $n = m$, X is ex-post PO. That is, every integral allocation in the support of the randomized allocation (induced by X) is PO.*

Proof. Observe that every agent gets exactly 1 good in every realization of X.

For the sake of contradiction, let us assume there exists some allocation A in the support of X such that A is not Pareto Optimal. Let B be an allocation that dominates A. Let $N' = \{i \in N : B_i \neq A_i\}$. Note that all agents in N' are strictly improved in B. Construct a directed graph $G(V, E)$ where $V = N'$ and $(i, j) \in E(G) \iff B_i = A_j$. That is, agent i points to the agent j, if j's good under A is the bundle that i receives under B. This implies i strictly prefers the good A_j over A_i and thus would have started eating A_j before A_i. Let's denote $f(g)$ as the finish time of good g: the time at which g was completely exhausted during SE. See that $f(A_j) < f(A_i)$. Moreover, note that G is a cycle (as every vertex has in-degree and out-degree = 1). Labelling the vertices along the cycle as $1, 2, ..n'$, this would imply that $f(A_2) < f(A_1) < f(A_{n'}) < \cdots < f(A_2)$, which is a contradiction. □

For a deeper comparison of the Simultaneous Eating and Random Permutation mechanisms, the reader may refer to the work of Bogomolnaia and Moulin [14].

2.3.2 Birkhoff - von Neumann Decomposition

We have now examined a mechanism that computes fractional allocations satisfying EF. The next challenge is to implement this fractional allocation over a set of EF1 integral allocations. To achieve this, we leverage a classic theorem of Birkhoff and von-Neumann [15, 32] about doubly stochastic matrices.

Definition 2.3.5. *A doubly stochastic matrix is a non-negative matrix $X = [x_{ij}]$ such that $\sum_j x_{i,j} = 1$ for all i and $\sum_i x_{i,j} = 1$ for all j, that is, each row and each column of X sums to* 1*.*

Any doubly stochastic matrix is a square matrix, as the total sum of its entries, computed row-wise or column-wise, must be equal. A permutation matrix is a doubly stochastic matrix where every entry is either a 0 or a 1. Thus, a permutation matrix contains exactly one 1 in every row and every column.

Theorem 2.3.5 (B-vN Decomposition)**.** *Given an $n \times n$ doubly stochastic matrix X, there exists an algorithm that decomposes X into a convex combination of permutation matrices in poly. time. Formally,*

$$X = \sum_{k=1}^{\ell} p_k \cdot A^k$$

such that for each $k \in [\ell] : A^k$ is a permutation matrix, $p_k \in [0, 1]$ and $\sum_{k=1}^{\ell} p_k = 1$

Proof. The proof of the theorem serves roughly as a guide for the description of Algorithm 8. We shall prove by induction on $\lambda =$ the number of non zero entries in X. Observe that $\lambda \geq n$ for any doubly stochastic matrix.

(Base Case) When $\lambda = n$, clearly, X is a permutation matrix itself. We now assume the induction hypothesis holds for all cases where the number of nonzero entries in X is less than a given $\lambda > n$ and show that it also holds for λ as well.

Let us construct a bipartite graph $G_X = (R \cup C,\, E_X)$ where $R = C = [n]$ and an edge $(i,\, j) \in E_X \iff X_{i,j} > 0$.

Lemma 2.3.1. *Given a doubly stochastic matrix X, its positivity graph G_X contains a perfect matching.*

Proof. We shall show the existence of perfect matching in G_X using Hall's theorem. Let S be any subset of rows of X (corresponding to a set of vertices in R). Define $N(S)$ as neighbourhood of S in G_X. Then, $N(S) = \{j \in [n] : X_{i,j} > 0 \text{ for some } i \in S\}$. Permute the rows S and columns $N(S)$ so that they form a contiguous submatrix X' of X. Looking at entries restricted to rows in S, any column $j \notin N(S)$ will only have zero entries. This implies, each row of X' sums to 1 and thus all entries of X' sum to a total of $|S|$. However, since each column sum of $X' \leq 1$ and therefore, $|S| \leq |N(S)|$.

This is exactly the condition specified by Hall's theorem, and thus G_X has a perfect matching. $\square$

G_X has a perfect matching, and this matching corresponds to a transversal in the matrix X. Let α_0 be the smallest entry of this transversal. Note that, $\alpha_0 > 0$. Let A_0 be a matrix where exactly the elements corresponding to the transversal are 1 and the rest are 0.
Then, $X - \alpha_0\, A_0$ is a non-negative matrix, with every row and column sum $= 1 = \alpha_0$. Hence, $X_1 = \dfrac{(X - \alpha_0\, A_0)}{1 - \alpha_0}$ is also a doubly stochastic matrix, but has $\leq \lambda - 1$ non-zero entries. Using our induction hypothesis, one can compute a convex decomposition of X_1:

$$X_1 = \alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_\ell A_\ell \quad \text{where} \sum_{i=1}^{\ell} \alpha_i = 1$$
$$\implies X = \alpha_0 A_0 + (1 - \alpha_0)(\alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_\ell A_\ell)$$

It is easy to see that the coefficients in the decomposition of X are all non-negative and sum to 1.

Running time of Algorithm 8 : For a doubly stochastic matrix X of dimension n, we can construct the positivity graph G_X in $O(n^2)$. To compute a perfect matching, one can use the Hopcroft-Karp algorithm [33] which takes $O(E_X\sqrt{n})$ which in the worst case is $O(n^{2.5})$. This is the slowest step. Thus, each iteration takes $O(n^{2.5})$ time. The number of iterations is $O(n^2)$ as well, and thus Algorithm 8 will terminate in $O(n^{4.5})$. $\square$

Any doubly stochastic matrix of dimension n belongs to the convex hull of some permutation matrices in $\mathbb{R}^{n \times n}$. By virtue of Caratheodory's theorem, we know that any such matrix can be decomposed into a convex combination of at most $n^2 + 1$ permutation matrices. A more direct way to understand this is, since at every iteration of the algorithm, atleast one term becomes 0, the algorithm must terminate in atmost n^2 iterations. Infact, since the last iteration must make n terms simultaneously 0, it follows that $\ell \leq n^2 - n + 1$ iterations.

Algorithm 8: *B - vN Decomposition*

Input: A doubly stochastic $n \times n$ matrix X
Output: A set of tuples $\mathcal{A} = (A^k, p_k)_{k \in [\ell]}$ where for every k, A^k is an $n \times n$ permutation matrix and $0 \leq p_k \leq 1$

Initialize $i \leftarrow 0$ and $X' \leftarrow X$
while *X' has a non-zero entry* **do**
    $i \leftarrow i + 1$
    $G_{X'} \leftarrow$ construct the positivity graph of X'
    $M_i \leftarrow$ compute a perfect matching in $G_{X'}$; `// Compute the maximum matching in` $G_{X'}$ `using Hopcroft-Karp`
    $p_i \leftarrow$ smallest entry in the transversal corresponding to M_i
    $A_i \leftarrow n \times n$ permutation matrix consistent with the transversal ; `//` (j,k)`-th entry of` A_i `is 1 if` $(j,k) \in E(G_{X'})$`, and 0 otherwise`
    $X' \leftarrow X' - p_i \cdot A_i$
return $(A_1, \ldots, A_i)$ and $(p_1, \ldots, p_i)$

However, it should be noted that the B-vN decomposition is **not** a unique decomposition.

Theorem 2.3.6 (Caratheodory's Theorem)**.** *The convex hull of a set A (Conv(A))is the smallest convex set that contais A. If $x \in \mathrm{Conv}(A)$ for some $A \subset \mathbb{R}^d$, then x lies in some d-dimensional simplex with vertices $\in A$.*

This is a fundamental result whose proof we omit here. It follows that we can express x as a convex combination of at most $d+1$ points of A (the extremal points of the simplex). The Birkhoff–von Neumann Algorithm can also be extended to the case where the matrix is singly stochastic, meaning that either the row sums or the column sums (but not both) is equal to 1. Additionally, numerous extensions and variants of the algorithm have been widely studied in the literature [34, 35, 36, 37].

2.3.3 The PS-Lottery Algorithm

We shall now motivate the main result of Aziz, Freeman, Vaish and Shah [13] - the *Probabilistic Serial-Lottery* algorithm, which computes an allocation simultaneously ex-ante EF and ex-post EF1 for additive agents. The Simultaneous Eating mechanism and the Birkhoff-von Neumann Decomposition serve as key subroutines in the *PS-Lottery* algorithm introduced here.

Description of PS-Lottery: We first add some dummy goods to ensure that the total number of goods is a multiple of n. Let M' denote this new set of goods. We then run the Simutaneous Eating mechanism on M', which proceeds for c time units (where $cn = m$) producing a fractional allocation of size $n \times cn$. We expand this into a $cn \times cn$ square matrix, but taking c copies of every agent - where each copy represents the agent's consumption in each minute 1 through c. Observe that the each row in this square matrix sums to 1. We then use B-vN Decomposition to decompose it into a set of permutation matrices (in which, each agent copy receives exactly 1 good).

Finally, we remove the dummy goods and consolidate the allocations of each agent's copies into a single allocation for the original agent they represent. Taking a convex combination over these modified permutation matrices yields the desired randomized allocation.

Algorithm 9: *PS-Lottery*

Input: An additive goods instance $\langle N, M, \mathcal{V} \rangle$

Output: A *randomized* allocation $X = \sum_{k=1}^{\ell} p_k\, A^k$

$c \leftarrow \lceil m/n \rceil$
if n *divides* m **then**
  $D = \emptyset$
else
  $D = \{d_1, d_2, ..., d_{cn-m}\}$
$M' \leftarrow M \cup D$
for *for every* $d \in D$ *and* $g \in G$ **do**
  $g \succ_i d$ for all $i \in N$`// agent preferences over goods within` D `can be arbitrary`
$X' \leftarrow$ fractional allocation obtained by running Simultaneous Eating (SE) on M'
$N' \leftarrow \{i_1, i_2, ..., i_c : i \in N\}$
Construct a fractional allocation Y, where, for $t \in [c]$, the mass eaten by every agent $i \in N$ during the t^{th} time unit is allocated to i_t.
$(B^k, p_k)_{k \in [\ell]} \leftarrow$ B-vN Decomposition(Y)
Convert $Y = \sum_{k=1}^{\ell} p_k\, B^k$ back to $X = \sum_{k=1}^{\ell} p_k\, A^k$; `// remove the dummy goods and each agent` $i \in N$ `gets the bundles of all its copies`
return X and the decomposition $\sum_{k=1}^{\ell} p_k\, A^k$

Theorem 2.3.7. *The PS-Lottery algorithm produces a randomized allocation X that is ex-ante EF and ex-post EF1.*

Proof. The ex-ante guarantees simply follow from the fact that X is equivalent to the fractional allocation returned by SE in step 8 (one can simply assume the valuations of the dummy goods to be 0). To show ex-post EF1:
Fix any integral allocation A^k. Let us transform it back into B^k, as in Line 12. Let g_{i_t} be the good assigned to the agent i_t in B^k.

Lemma 2.3.2. *For all $t \in [c-1]$, it holds that $g_{i_t} \succ_i g_{j_{t+1}}$ for any $j \neq i$.*

Proof. Let's assume this is not true. So, for some $j \in N \setminus i$ and t, i prefers $g_{j_{t+1}}$ over g_{i_t}. However, see that i consumed a positive fraction of g_{i_t} in $[t-1, t]$ and j consumed a positive fraction of $g_{j_{t+1}}$ in $[t, t+1]$. Thus, $g_{j_{t+1}}$ was still available during $[t, t+1]$. This is a contradiction because i could have started eating g_{i_t} only after $g_{j_{t+1}}$ was completely exhausted. $\square$

By this lemma, we immediately obtain

$$\sum_{t\in[c-1]} v_i(g_{i_t}) \geq \sum_{t\in[c-1]} v_i(g_{j_{t+1}}) \quad \forall\, i,\, j \in N$$

Thus, A^k is EF1 (this is a very similar argument to show why the Round Robin algorithm is EF1). Hence, X is ex-post EF1. $\square$

The runnning time of the algorithm is dominated by the step where we invoke Birkhoff-von Neumann decomposition on a $cn \times cn$ matrix. Note that $cn \leq m+n$ and thus, takes $O((m+n)^{4.5})$. Other steps like computing X using SE and the transformations between X and Y take atmost $O(mn)$ time. Thus, the running time of PS-Lottery is $O((m+n)^{4.5})$.

Another notable remark is that the algorithm only requires ordinal agent preferences. Hence, X is also SD-EF allocation.

Proposition 2.3.2. *The randomized allocation X returned by PS-Lottery is SD-Envy Free.*

While the above algorithm and the discussions pertain to goods-only instances, the results and guarantees of PS-Lottery also extend when all the items are chores. The SE mechanism still produces a fractional allocation that is EF in the chores case, and the ex-post guarantee follows from a very similar proof style. Thus, Aziz *et al.* [13] positively resolve the question of ex-ante EF and ex-post EF1 in the additive setting. However, whether there always exists a randomized allocation that is ex-ante EF and ex-post (EF1+PO) is still an open question to date.

In contrast, if PO is replaced with fPO in the ex-post requirement, such an allocation is impossible to guarantee.

Theorem 2.3.8 (Impossibility)**.** *There exists an additive instance for which no randomized allocation is ex-ante EF and ex-post EF1 + fPO*

	g_1	g_2
a	1	2
b	1	3

Table 2.8: Counter example for ex-ante EF and ex-post EF1+fPO

Proof. There are only 2 EF1 allocations: one where g_1 goes to agent a, g_2 goes to agent b (and vice versa). However, the latter allocation (call it A) is not fPO. The fractional allocation where g_1 is completely given to a, and g_2 is equally split between both agents, will dominate A. Thus, any randomized allocation that is ex-post (fPO + EF1) has exactly one allocation in its support. Clearly, agent a envies agent b in this allocation $\implies$ not ex-ante EF. $\square$

The BoBW framework presents several promising directions for further exploration. For instance, one could investigate stronger ex-post fairness guarantees such as EFX. Additionally, there has been work in extending simultaneous fairness guarantees beyond the additive setting. Feldman *et al.* [20] showed an algorithm to compute randomized allocations for sub-additive agents that satisfies ex-ante $1/2$ EF, ex-post $1/2$ EFX and ex-post EF1.

The next chapter focuses on our efforts to strengthen ex-post EF1 to ex-post EFX while preserving ex-ante EF, albeit within a restricted subclass of additive valuations—lexicographic valuations.

Chapter 3

Results: BoBW Algorithms For Lexicographic Valuations

In the previous chapters, we saw that while EFX for additive valuations is still an open problem (for $n \geq 4$), not only does EFX always exists for lexicographic agents, but can be efficiently computed. Furthermore, when the items are all goods, the set of EFX + PO allocations can be efficiently characterized by Algorithm 6.

Building on the Best of Both Worlds framework we saw earlier, this chapter addresses the question: Given any lexicographic goods instance, can we efficiently compute a randomized allocation that is ex-ante EF and ex-post EFX? Moreover, can we achieve ex-post PO as well? This chapter comprises entirely of original work.

We begin by studying a natural approach to this problem and highlight its limitations. We then present our main result - the *Uniform Tailed Simultaneous Eating* algorithm, that computes a randomized allocation that is ex-ante $6/7$EF and ex-post (EFX+PO), and then formally prove it. Additionally, we discuss why the goods setting presents unexpected challenges, making the problem more intricate than it initially appears.

Extending this question to the case of chores, we present an algorithm that achieves ex-ante EF + ex-post EFX, but fails ex-post PO. We then detail the limitations of a very natural approach to strengthen the ex-post guarantee to EFX + PO. In the final section, we show that by relaxing ex-ante EF to ex-ante PROP, it is possible to obtain ex-ante fairness in conjunction with ex-post (EFX + PO) for both goods and chores settings.

3.1 Uniform Random Permutation

In this section and the next, we deal with the goods-only instances. Just like in the additive setting, one natural approach is to identify a set of EFX allocations and uniformly randomize over them. Algorithm 5 is a good candidate to start with. We uniformly pick a permutation of the agents and assign goods via one iteration of Round Robin. The remaining $m - n$ goods are given to the last agent of the picked permutation. (Refer to Algorithm 10)

However, we show that this approach works only for $n = 2$ and fails ex-ante EF for $n > 2$.

Algorithm 10: *Uniform Random Permutation*

Input: A lexicographic goods instance $\langle N, M, \mathcal{V} \rangle$

Output: A randomized allocation A

```
A ← (∅, ...., ∅)
𝓟 ← set of all permutations of agents in N = {1, 2, ..., n}
σ ← a permutation chosen uniformly at random from 𝓟
for each agent i in the order given by σ do
    g ← agent i's favourite available good
    A_i ← A_i ∪ g
The remaining m − n goods are allocated to the last agent in σ. Update A
  accordingly.
return A
```

Proposition 3.1.1. *For any lexicographic goods instance with $n = 2$ agents, Uniform Random Permutation returns a randomized allocation that is ex-ante EF + ex-post (EFX + PO).*

Proof. The proof of the ex-post guarantees follow from the proof of Algorithm 5. We now show ex-ante EF. There are only 2 orderings to choose from: each agent has 0.5 chance of receiving their top good in M and 0.5 chance of receiving the rest of the goods. Thus, from agent i's perspective, both agents receive an expected utility of $\dfrac{v_i(M)}{2}$ for both $i = 1, 2$. Hence, it is ex-ante EF. $\square$

We now show a very simple counterexample for why this does not extend to $n = 3$ agents.

Proposition 3.1.2. *For the given instance with $n = 3$ agents, Uniform Random Permutation fails to achieve ex-ante EF.*

Proof. This is the same counter example used to show why *Randomized Round Robin* fails ex-ante EF. It is easy to see why so, as when $m = n$, both algorithms return the same allocations.

	g_1	g_2	g_3
a	$1+\epsilon$	1	$1-4\epsilon$
b	$1-4\epsilon$	$1+\epsilon$	1
c	$1+\epsilon$	$1-4\epsilon$	1

Table 3.1: *Uniform Random Permutation* is not ex-ante EF

See that when $0.2 < \epsilon < 0.25$, the valuations are lexicographic. It is easy to verify that $\mathbb{E}[v_a(A_a)] = \dfrac{6-5\epsilon}{6}$ while $\mathbb{E}[v_a(A_b)] = \dfrac{6-4\epsilon}{6}$. Hence, agent a envies agent b $\square$

The intuition of why the algorithm doesn't work is again very similar to why *Randomized Round Robin* doesn't. For two agents, i and j, the probability of i going before or after j is the same, but the deficit accumulated by i in the latter case could still outweigh any advantage gained in the former. Furthermore, multiple agent orderings could lead to the same allocation that further amplifies the envy. However, in counterexample 3.1, agent a's envy towards b is minute ($\epsilon/6$). Infact, the allocation is ex-ante $^{19}/_{20}$ -EF.
This brings us to our first result:

Theorem 3.1.1. *Uniform Random Permutation is ex-ante $^1/_2$ -EF and ex-post EFX + PO*

Proof. Fix 2 agents i, j. Pick an arbitrary permutation $\boldsymbol{P}$ from the set of all permutations of agents $\boldsymbol{\mathcal{P}}$. WLOG say i goes before j in $\boldsymbol{P}$. Let k_1 and k_2 be the positions of agents i, j respectively, in $\boldsymbol{P}$. Define the *partner* of $\boldsymbol{P}$, denoted by $\boldsymbol{P}^*$, as the permutation obtained by swapping the positions of agents i and j while keeping all other agents in the same positions as in $\boldsymbol{P}$. Observe that the set $\boldsymbol{\mathcal{P}}$ can be partitioned into such pairs, each consisting of a permutation and its partner.

In $\boldsymbol{P}$, let agent i pick good g and agent j's bundle be Q_j. Likewise in $\boldsymbol{P}^*$, j picks good g' and i's bundle is Q_i. Note that if $k_2 = n$, then Q_i and Q_j could be a set of multiple goods, and not just a singleton.

Now, let us analyze i's envy towards j restricted to just the permutations $\boldsymbol{P}$ and $\boldsymbol{P}^*$.

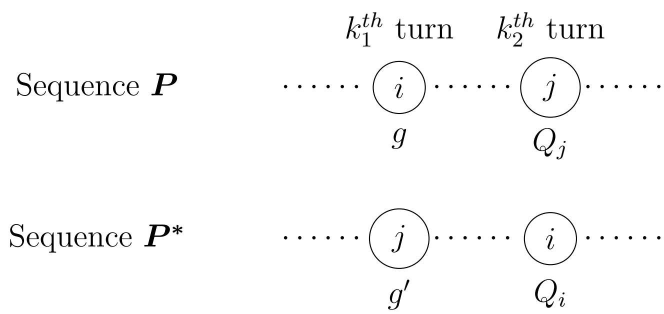

Figure 3.1: We consider permutations $\boldsymbol{P}$ and its partner $\boldsymbol{P}^*$ where the positions of agents i and j are swapped, while all other agents remain unchanged. Let k_1 and k_2 denote agents i and j's positions in $\boldsymbol{P}$ respectively. If $k_2 = n$, then Q_j and Q_i could represent a set of goods rather than a singleton.

Case 1: $g = g'$

Observe that all agents between i and j will pick the same items in both $\boldsymbol{P}^*$ and $\boldsymbol{P}$. Hence, the set of available goods is the same during agent i's turn in $\boldsymbol{P}^*$ and j's turn in $\boldsymbol{P}$. Therefore, $v_i(Q_i) \geq v_i(Q_j)$. Thus, agent i isn't envious of j.

Case 2: $g \neq g'$

Both sequences are drawn uniformly at random with some probability p. Let us define the expected utility of the agents (from agent i's perspective) restricted to $\boldsymbol{P}$ and $\boldsymbol{P}^*$ as follows:

$$\mathbb{E}^*[v_i(X_i)] = p \cdot [v_i(g) + v_i(Q_i)]$$

$$\mathbb{E}^*[v_i(X_j)] = p \cdot [v_i(g') + v_i(Q_j)]$$

In both $\boldsymbol{P}$ and $\boldsymbol{P^*}$, the agents before position k_1 are ordered identically. Therefore, in both sequences, the exact same goods are picked until position k_1. Hence, good g' is available during agent i's turn in $\boldsymbol{P}$, but since it picks g instead, $v_i(g) > v_i(g')$. Since the valuations are lexicographic, even if $|Q_j| \geq 2$, $v_i(g) > v_i(Q_j)$. If $g' \notin Q_j$, then $v_i(g) > v_i(Q_j \cup g') = v_i(Q_j) + v_i(g')$. It follows that:

$$\mathbb{E}^*[v_i(X_i)] \geq \mathbb{E}^*[v_i(X_j)]$$

Thus, given $g' \notin Q_j$, agent i is ex-ante EF towards j.

We now have to consider the case where $g' \in Q_j$. See that,

$$\mathbb{E}^*[v_i(X_j)] = p \cdot [2v_i(g') + v_i(Q_j \setminus g')]$$

Moreover, $g' \in Q_j$ means that all agents in between positions k_1 and k_2 prefer the good they picked in $\boldsymbol{P}$ over g'. Hence, in $\boldsymbol{P}^*$, these agents either pick the same good they picked in $\boldsymbol{P}$ or will pick good g. Hence, the set $Q_j \setminus g'$ is available during agent i's turn, implying that $v_i(Q_i) \geq v_i(Q_j)$.

$$\therefore \quad \frac{\mathbb{E}^*[v_i(X_j)]}{2} = p \cdot \left[v_i(g') + \frac{v_i(Q_j \setminus g')}{2} \right] < p \cdot [v_i(g) + v_i(Q_i)] = \mathbb{E}^*[v_i(X_i)]$$

For every such pair of permutations in $\boldsymbol{\mathcal{P}}$, agent i's expected utility for its own bundle is atleast half of i's expected utility for agent j's bundle. Thus, summing over all such permutation pairs, we have

$$2 \cdot \mathbb{E}[v_i(X_i)] \geq \mathbb{E}[v_i(X_j)].$$

Hence, agent i is ex-ante $\frac{1}{2}$ -EF towards agent j.

$\square$

Proof of ex-post EFX + PO. Let A be the allocation in some realization of the randomized algorithm. Pick any $i, j \in N$. Let us look at agent i's envy towards agent j under deterministic allocation A. If $|Aj| = 1$, then trivially i envy towards j is EFX. If $|A_j| > 1$, then it means that j was the last agent to pick according to the agent ordering. Since the valuations are lexicographic, no other agent envies agent j. Thus A is EFX.
It is trivial to see that A is sequencible. For lexicographic valuations, an allocation A is sequencible $\iff$ A is PO.
Thus, X is ex-post EFX+PO. $\square$

Remark 2. *If there exists a good h such that $g \succ_i h \succ_i g'$, then $v_i(g) > 2v_i(g')$ due to the lexicographic property. Thus, for every permutation pair $\boldsymbol{P}, \boldsymbol{P}^*$, if the correspondingly defined goods g and g' satisfy the above condition, then X is ex-ante* EF.

Theorem 3.1.2. *Uniform Random Permutation is not* ex-ante α -EF *for any* $\alpha > 0.75$

Proof. To show this, we construct an instance on which the algorithm returns an ex-ante $3/4 + \epsilon$ -EF allocation, where ϵ is sufficiently small

	g_1	g_2	g_3	g_4	g_5	g_6
1	$1 + 16\epsilon$	1	8ϵ	4ϵ	2ϵ	ϵ
2	1	$1 + 16\epsilon$	8ϵ	4ϵ	2ϵ	ϵ
3	1	ϵ	8ϵ	4ϵ	2ϵ	$1 + 16\epsilon$
4	1	ϵ	8ϵ	4ϵ	2ϵ	$1 + 16\epsilon$
5	1	ϵ	8ϵ	4ϵ	2ϵ	$1 + 16\epsilon$
6	1	ϵ	8ϵ	4ϵ	2ϵ	$1 + 16\epsilon$

Table 3.2:

Let us denote the randomized allocation as X. There are $6! = 720$ permutations and 168 distinct EFX+PO allocations. Let us look at agent 1's envy towards 2:

$$\frac{\mathbb{E}[v_1(X_1)]}{\mathbb{E}[v_1(X_2)]} = \frac{288v_1(g_1) + 144v_1(g_2) + 72v_1(g_3) + 96v_1(g_4) + 12v_1(g_5)}{576v_1(g_2) + 24v_1(g_3) + 48v_1(g_4) + 72v_1(g_5)}$$
$$= \frac{432 + 5805\epsilon}{576 + 528\epsilon}$$

This is a monotonic function with a minima of 0.75. Hence, for sufficiently small $\epsilon > 0$, X can not do better than ex-ante $3/4$ -EF. □

3.2 Uniform Tailed Simultaneous Eating

We now present our main result, the *Uniform Tailed Simultaneous Eating* algorithm. For any lexicographic goods instance, the allocation computed by our algorithm is ex-ante $6/7$-EF and ex-post (EFX + PO). We also identify the criteria for when the algorithm achieves exact ex-ante EF.

Description of the algorithm: The algorithm comprises of 2 Phases. In Phase 1, we run the SE mechanism for exactly 1 unit time as a subroutine in our algorithm. The fractional partial allocation obtained from this procedure is then decomposed into a convex combination of binary matrices (not permutation matrices!) using B-vN Decomposition. It is easy to see that in every binary matrix, exactly one entry per row is a one, while the rest are zeroes, thus corresponding to a matching. We sample one of these matchings as per the decomposition. Phase 2: We identify a set of agents who received a good that was not fully consumed in Phase 1. We term such a good as *fractionally consumed*. Observe that such agents would be unenvied in our matching. The *tail* (all the remaining unallocated goods) is then completely assigned to one of these agents chosen uniformly at random. (Refer Algorithm 11 for the pseudo-code)

Running time: The simultaneous eating protocol, though seemingly continuous, can be discretized by partitioning the interval $[0, 1]$ into breakpoints—specific time points where agents stop consuming a good and switch to another. The number of such breakpoints required is $O(f(n, m))$ for some polynomial f, ensuring a polynomial-time implementation

Algorithm 11: *Uniformly Tailed Simultaneous Eating*

Input: An instance $\langle N, M, \mathcal{V} \rangle$ with lexicographic valuations

Output: A randomized allocation Z

`/* Phase 1: Simultaneous Eating and Sampling */`

$X \leftarrow$ the fractional allocation obtained by running the Simultaneous Eating algorithm (*SE*) for 1 unit of time

$F \leftarrow \{g \in M : 0 < \sum_{i=1}^{n} X_{i,g} < 1\}$; `// Set of goods that are fractionally consumed but not completely exhausted after` *SE*

$U \leftarrow \{g \in M : \sum_{i=1}^{n} X_{i,g} = 0\}$; `// set of items untouched during` *SE*

$(p_k, A^k)_{k\in[q]} \leftarrow$ B-vN Decomposition (X)

$Z \leftarrow$ an integral allocation sampled from $(p_k, A^k)_{k\in[q]}$

`/* Phase 2: Allocate the tail */`

$R \leftarrow$ set of goods that are unallocated after Phase 1

$N_F \leftarrow \{i \in N : Z_i \cap F \neq \emptyset\}$; `// set of agents who receive some` $g \in F$ `in` Z_k

Pick an agent i uniformly at random from N_F

$Z_i \leftarrow Z_i \cup R$; `// assign` i `all remaining unallocated goods`

return Z

[31]. We then invoke B-vN Decomposition(X), that runs in $O(m^{4.5})$. Consequently, Phase 1 runs in polynomial time. Similarly, sampling a matching, computing N_F, and assigning all remaining goods to a single agent can each be performed efficiently, ensuring that Phase 2 also runs in polynomial time.

Theorem 3.2.1. *For every goods instance with lexicographic valuations, Uniformly Tailed Simultaneous Eating outputs a randomized allocation that is ex-ante $6/7$-EF and ex-post EFX+PO in polynomial time.*

Proof of ex-post EFX+PO. At the end of Phase 1 of *Uniformly Tailed SE*, each agent possesses exactly one item, making the partial allocation trivially EFX. In Phase 2, the remainder of the items go to some agent i who received a *fractional* good ($g \in F$) in Phase 1. Consequently, only an unenvied agent receives additional goods in Phase 2. Hence, the final allocation remains EFX, since the agents are lexicographic.

Firstly, observe that every binary matrix sampled is a matching i.e exactly 1 good is assigned to each agent. The goods that are completely consumed in Phase 1 are always allocated to some agent, while goods that are *fractionally* consumed are unallocated in some of the matchings that X is decomposed into. See that any integral allocation A in the support of X (obtained via SE for 1 time unit) is sequencible. An argument similar to the one we used to show SE is ex-post PO (when $m = n$) works here. Since all remaining goods are assigned to a single agent in Phase 2, the overall allocation remains sequencible. Since the instance is lexicographic, $\implies$ the final allocation is PO. $\square$

Proof of ex-ante $^6/_7$ *-EF.* For any good $g_h \in M$ and agent $h \in N$, let $x(h, g_h)$ denote the fraction of g_h consumed by agent h in Step 1. Let $X_h^f[t_1 : t_2]$ represent the total mass of goods consumed by agent h between time points $0 \leq t_1 \leq t_2 \leq 1$ during Step 1. Let X_h be the *r.v* that denotes the (singleton) bundle of agent h in the matching sampled in Step 5. Let Y_h denote the random bundle assigned to agent h in Phase 2. Thus, h's bundle in the final allocation is $Z_h = X_h \cup Y_h$. Note that, if $h \notin N_F$, then $Y_h = \emptyset$.

Fix agents i, j. During Step 1 of the algorithm, say agent i has eaten some fractions of goods $g_1, g_2, g_3 \cdots g_\ell$. The indices of these goods denote the order in which i ate from them. Similarly, let agent j eat from goods $g'_1, g'_2, g'_3, \cdots g'_u$ in Step 1. Out of these goods, only g_ℓ and g'_u could potentially belong to F.

Let us define $G' = F \cup U$. Note that G' is a superset of the 'tail' - the remaining unallocated goods after Phase 1. Let k be the number of agents in N_F, that is, the number of agents who received a fractional good in the matching sampled in Step 5. This is equivalent to saying that $n - k$ goods were completely consumed during Step 1 of the Algorithm.

With this notation in place, we now analyze agent i's envy towards agent j.

Case 1: $g'_u \notin F$

Running the Simultaneous Eating Algorithm for any amount of time yields a fractional allocation that is EF. Therefore, at the end of 1 unit time of SE, all agents are ex-ante EF. Note that $g'_u \notin F \implies j \notin N_F$, and therefore, agent j will never get the 'tail'. Thus, agent

i continues to remain ex-ante EF towards j regardless of whether agent j receives additional goods in Phase 2.

Case 2: $g'_u \in F$ *but* $g_\ell \notin F$

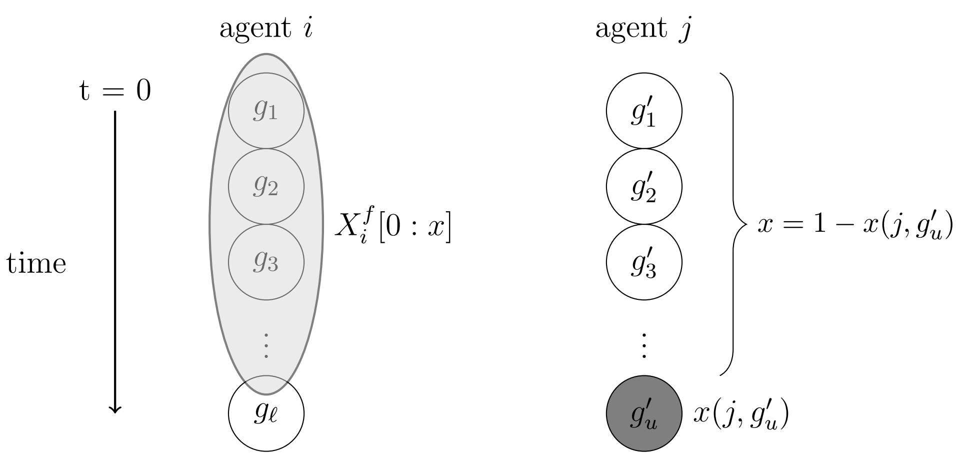

Figure 3.2: Case 2: The goods of agents i, j displayed in order of their consumption in Phase 1. All goods displayed in white are fully exhausted by the end of Phase 1, while the good g'_u $\in F$ (dark grey) is *fractionally* consumed. The quantity $x(j, g'_u)$ represents the fraction of g'_u consumed by j during Phase 1. The translucent ellipse $X_i^f[0:x]$ denotes the equivalent mass eaten by agent i till time $x = 1 - x(j, g'_u)$ during Phase 1.

In this case, $Y_i = \emptyset \implies Z_i = X_i$. Let us define $x = 1 - x(j, g'_u)$. Hence,

$$v_i(X_i^f[0:x]) \geq v_i(X_j^f[0:x]) \tag{3.1}$$

Observe that $X_j^f[x:1] = x(j, g'_u)$ and therefore,

$$v_i(X_i^f[x:1]) \geq x(j, g'_u) \cdot v_i(g'_u)$$

Agent i prefers any good that it ate during the time interval $[x:1]$, over any good in G'. Thus, by the lexicographic property:

$$v_i(X_i^f[x:1]) \geq x(j, g'_u) \cdot [v_i(g'_u) + v_i(G' \setminus g'_u)] \tag{3.2}$$

Summing the inequalities (3.1) and (3.2) gives:

$$\begin{aligned}\mathbb{E}[v_i(Z_i)] = v_i(X_i^f[0:x]) + v_i(X_i^f[x:1]) &\geq v_i(X_j^f[0:x]) + x(j, g_u')[v_i(g_u') + v_i(G' \setminus g_u')] \\ &\geq \mathbb{E}[v_i(X_j) + v_i(Y_j)] \\ &= \mathbb{E}[v_i(Z_j)]\end{aligned}$$

Hence, agent i is ex-ante EF towards agent j.

Case 3: $g_u' = g_\ell = g \in F$

In this case, both agents i and j's eat from the same good $g \in F$ during Step 1.

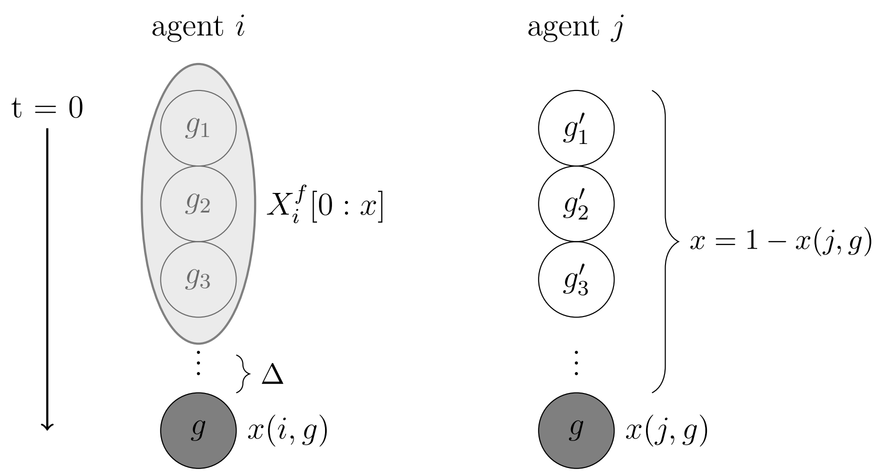

Figure 3.3: Case 3.a:
The goods of i, j displayed in order of their consumption in Phase 1. All goods displayed in white are fully exhausted by the end of Phase 1, while the good $g \in F$ (shaded in grey). The variable defined as $\Delta = x(j,g) - x(i,g)$. During the time interval $[x : x + \Delta]$, j is already eating g while i joins in on g at time $x + \Delta$.

We analyze the expectation:

Subcase 3(a): $x(i,g) < x(j,g)$

Define $x = 1 - x(j,g)$ and $\Delta = x(j,g) - x(i,g)$. In the matching sampled in Step 5, if agent i receives the partially eaten item g , it has a $1/k$ chance of receiving the tail. However, this does not necessarily mean that i will receive all of $G' \setminus g$, since fractional goods other than g may be allocated to other agents in the matching.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq v_i(X_j^f[0:x]) + x(j,g)v_i(g) + \frac{1}{k}x(j,g)v_i(G' \setminus g) \tag{3.3}$$

$$= v_i(X_j^f[0:x]) + (x(i,g) + \Delta)\, v_i(g) + \frac{1}{k}(x(i,g) + \Delta)\, v_i(G' \setminus g) \tag{3.4}$$

$$= v_i(X_j^f[0:x]) + \Delta \left[v_i(g) + \frac{1}{k} v_i(G' \setminus g) \right] + x(i,g) \left[(v_i(g) + \frac{1}{k} v_i(G' \setminus g) \right] \tag{3.5}$$

$$\leq v_i(X_i^f[0:x]) + v_i(X_i^f[x : x + \Delta]) + x(i,g)\, v_i(g) + \frac{x(i,g)}{k} v_i(G' \setminus g)) \tag{3.6}$$

$$= \mathbb{E}[v_i(X_i)] + \frac{x(i,g)}{k} v_i(G' \setminus g) \tag{3.7}$$

We bound this additional term containing $v_i(G' \setminus g)$:

$$\frac{1}{k} x(i,g)\, v_i(G' \setminus g) = \frac{1}{k} \frac{x(i,g)v_i(G' \setminus g)}{\cdot \mathbb{E}[v_i(X_i)]} \cdot \mathbb{E}[v_i(X_i)] \tag{3.8}$$

$$\leq \frac{\frac{1}{k} x(i,g)v_i(G' \setminus g)}{x(i,g)v_i(g) + [1 - x(i,g)]v_i(g_{\ell-1})} \cdot \mathbb{E}[v_i(X_i)] \tag{3.9}$$

$$\leq \frac{\frac{1}{k} x(i,g)v_i(G' \setminus g)}{x(i,g)v_i(G' \setminus g) + 2[1 - x(i,g)]v_i(G' \setminus g)} \cdot \mathbb{E}[v_i(X_i)] \tag{3.10}$$

$$= \frac{1}{k} \frac{x(i,g)}{2 - x(i,g)} \cdot \mathbb{E}[v_i(X_i)] \tag{3.11}$$

Since this function is monotone in $x(i,g)$ and $x(i,g) \leq 0.5$, and $k \geq 2$, we obtain:

$$\frac{x(i,g)}{k} v_i(G' \setminus g) \leq \frac{1}{6} \mathbb{E}[v_i(X_i)] \tag{3.12}$$

$$\therefore \quad \mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq \frac{7}{6} \mathbb{E}[v_i(X_i)] \leq \frac{7}{6} \mathbb{E}[v_i(X_i) + v_i(Y_i)].$$

Thus, i is $\frac{6}{7}$ ex-ante EF towards j.

Subcase 3(b): $x(i,g) \geq x(j,g)$

Define $x = 1 - x(i,g)$ and $\Delta = x(i,g) - x(j,g)$.

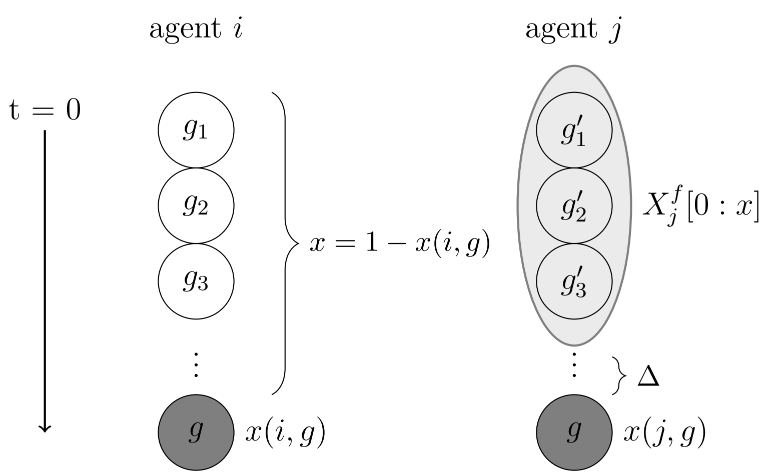

Figure 3.4: **Case 3(b)**: The goods of agents i, j displayed in order of their consumption in Step 1. Here, $\Delta = x(i,g) - x(j,g)$ and $x = 1 - x(i,g)$. During the time interval $[x : x+\Delta]$, agent i is already eating g while j joins in at time $x + \Delta$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq v_i(X_j^f[0:x]) + v_i(X_j^f[x:x+\Delta]) + x(j,g)\,v_i(g) + \frac{1}{k}x(j,g)\,v_i(G' \setminus g) \tag{3.13}$$

$$\leq \underbrace{v_i(X_i^f[0:x]) + \Delta \cdot v_i(g)} + x(j,g)\,v_i(g) + \frac{x(j,g)}{k}v_i(G' \setminus g) \tag{3.14}$$

$$\leq \underbrace{v_i(X_i^f[0:x]) + x(i,g)\,v_i(g)} + \frac{x(j,g)}{k}v_i(G' \setminus g) \tag{3.15}$$

$$= \mathbb{E}[v_i(X_i)] + \frac{x(j,g)}{k}v_i(G' \setminus g) \tag{3.16}$$

$$= \mathbb{E}[v_i(X_i)] + \frac{x(j,g)}{k}\frac{\mathbb{E}[v_i(X_i)]}{2 - x(i,g)} \qquad \text{(using eqn. (3.11))} \tag{3.17}$$

Now, we show that the additional term here is bounded as follows:

$$\frac{x(j,g)}{k} \cdot \frac{\mathbb{E}[v_i(X_i)]}{2 - x(i,g)} \leq \frac{x(j,g)}{k} \cdot \frac{\mathbb{E}[v_i(X_i)]}{1 + x(j,g)} \; where \; 0 < x(j,g) \leq 0.5 \tag{3.18}$$

Taking $k = 2$, this function is monotone in $x(j, g)$ and attains a maximum of 1/6 at $x(i, g) = x(j, g) = 0.5$

$$\therefore \quad \mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq \frac{7}{6}\mathbb{E}[v_i(X_i)] \leq \frac{7}{6}\mathbb{E}[v_i(X_i) + v_i(Y_i)]. \tag{3.19}$$

Thus, i is $\frac{6}{7}$ ex-ante EF towards j.

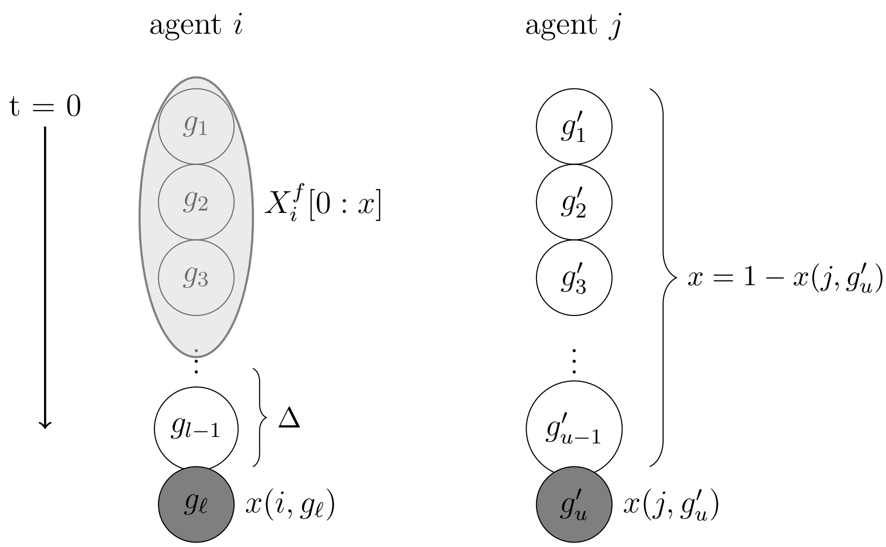

Figure 3.5: Case 4(a):
The goods of agents i, j displayed in order of their consumption in Step 1. The variable defined as $\Delta = x(j, g'_u) - x(i, g_\ell)$ and $x = 1 - x(j, g'_u)$. All goods displayed in white are fully exhausted at the end of Step 1, while the goods g'_u and g_ℓ are fractionally consumed. During the time interval $[x : x + \Delta]$, agent j is already eating g'_u while agent i starts eating g_ℓ only at time $x + \Delta$.

Case 4: $g'_u\,, g_\ell \in F\,, but\; g'_u \neq g_\ell$

Similar to our previous case, define $x(i, g_\ell)$ and $x(j, g'_u)$ denote the fraction of their final goods eaten by agents i and j, respectively. Let X_i denote the goods allocated to i after decomposing the fractional matrix X, and let Y_i denote the random bundle received by i in Phase 2. Observe that with probability $1 - x(i, g_\ell)$, $Y_i = \emptyset$.

Subcase 4(a): $x(i, g_\ell) < x(j, g'_u)$

Let $x = 1 - x(j, g'_u)$ and $\Delta = x(j, g'_u) - x(i, g_\ell)$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq v_i(X_j^f[0:x]) + x(j, g'_u)\left(v_i(g'_u) + \frac{1}{k}v_i(G' \setminus g'_u)\right) \tag{3.20}$$

Given j receives g'_u in Phase 1, it could receive g_ℓ in Phase 2, or g_ℓ could be allocated to some other agent in either of the Phases. We upper bound $\mathbb{E}[v_i(Y_j)]$ in the following way:

$$\begin{aligned}
\mathbb{E}[v_i(Y_j)] \leq\ & x(j, g'_u)\left(\frac{v_i(G' \setminus \{g'_u \cup g_\ell\})}{k}\right) \\
& + \frac{v_i(g_\ell)}{k}\Pr[\{g_\ell \text{ remains unallocated}\} \cap \{g'_u \text{ assigned to } j \text{ in Phase 1}\}] && (3.21) \\
\leq\ & x(j, g'_u)\left(\frac{v_i(G' \setminus \{g'_u \cup g_\ell\})}{k}\right) + \frac{v_i(g_\ell)}{k}\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\} && (3.22)
\end{aligned}$$

$$\therefore\ \mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq v_i(X_j^f[0:x]) + x(j, g'_u)\left(v_i(g'_u) + \frac{v_i(G' \setminus \{g'_u \cup g_\ell\})}{k}\right) \tag{3.23}$$

$$+ \frac{v_i(g_\ell)}{k}\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\} \tag{3.24}$$

$$\leq \underbrace{v_i(X_i^f[0:x])}_{\text{rewriting this}} + x(j, g'_u)\left(v_i(g'_u) + \frac{v_i(G' \setminus \{g'_u \cup g_\ell\})}{\underbrace{k}_{\text{ignoring } k}}\right) + \frac{v_i(g_\ell)}{k}\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\} \tag{3.25}$$

$$\leq [\mathbb{E}[v_i(X_i)] - x(i, g_\ell)v_i(g_\ell) - \Delta \cdot v_i(g_{\ell-1})] + x(j, g'_u)v_i(G' \setminus g_\ell) + \frac{v_i(g_\ell)}{k}\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\} \tag{3.26}$$

Due to lexicographic preferences of $i:\quad v_i(g_{\ell-1}) > v_i(g_\ell) + v_i(G' \setminus g_\ell)$

Thus, we replace $v_i(g_{\ell-1})$ with the smaller term to upper bound eqn. (3.26) as follows:

$$
\begin{aligned}
&\leq \mathbb{E}[v_i(X_i)] - \underbrace{x(i,g_\ell)\, v_i(g_\ell) - \Delta \cdot v_i(g_\ell)} - \underbrace{\Delta \cdot v_i(G' \setminus g_\ell) + x(j,g'_u)\, v_i(G' \setminus g_\ell)} \\
&+ \frac{v_i(g_\ell)}{k} \min\{x(j,g'_u),\, 1 - x(i,g_\ell)\} && (3.27) \\
&= \mathbb{E}[v_i(X_i)] - x(j,g'_u)\, v_i(g_\ell) + x(i,g_\ell)\, v_i(G' \setminus g_\ell) + \frac{v_i(g_\ell)}{k} \min\{x(j,g'_u),\, 1 - x(i,g_\ell)\} && (3.28) \\
&= \mathbb{E}[v_i(X_i)] - \underbrace{v_i(g_\ell)}_{\text{replace with } v_i(G' \setminus g_\ell)} \left(\underbrace{x(j,g'_u) - \frac{\min\{x(j,g'_u),\, 1 - x(i,g_\ell)\}}{k}}_{\geq 0} \right) + x(i,g_\ell)\, v_i(G' \setminus g_\ell) \\
& && (3.29) \\
&\leq \mathbb{E}[v_i(X_i)] + v_i(G' \setminus g_\ell) \left(x(i,g_\ell) - x(j,g'_u) + \frac{\min\{x(j,g'_u),\, 1 - x(i,g_\ell)\}}{k} \right) && (3.30) \\
&\leq \mathbb{E}[v_i(X_i)] + \frac{\mathbb{E}[v_i(X_i)]}{2 - x(i,g_\ell)} \left(x(i,g_\ell) - x(j,g'_u) + \frac{\min\{x(j,g'_u),\, 1 - x(i,g_\ell)\}}{k} \right) && \text{(using (3.11))} \\
& && (3.31) \\
&\leq \frac{7}{6} \cdot \mathbb{E}[v_i(X_i)] \qquad \text{(refer appendix 4)} && (3.32)
\end{aligned}
$$

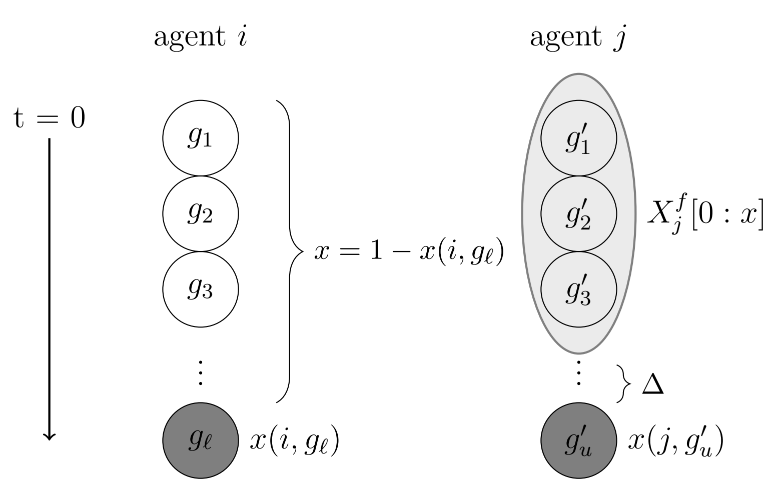

Figure 3.6: Case 4(b): The goods of agents i, j displayed in order of their consumption in Step 1. The variables $\Delta = x(i,g_\ell) - x(j,g'_u)$ and $x = 1 - x(i,g_\ell)$. During the time interval $[x : x + \Delta]$, i is already eating g_ℓ while j starts eating g'_u only at time $x + \Delta$.

Subcase 4(b): $x(i, g_\ell) \geq x(j, g'_u)$
Let $x = 1 - x(i, g_\ell)$ and $\Delta = x(i, g_\ell) - x(j, g'_u)$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] \leq v_i(X_j^f[0:x]) + v_i(X_j^f[x : x+\Delta])$$
$$+\ x(j, g'_u) \left[v_i(g'_u) + \frac{v_i(G' \setminus \{g'_u \cup g_\ell\})}{\underbrace{k}_{\text{ignore the k}}} \right] + v_i(g_\ell) \left(\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \right) \tag{3.33}$$

$$\leq \underbrace{v_i(X_i^f[0:x])}_{\text{write as a subtraction}} + \Delta \cdot v_i(g_\ell) + x(j, g'_u)\, v_i(G' \setminus g_\ell) + v_i(g_\ell) \left(\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \right) \tag{3.34}$$

$$\leq \mathbb{E}[v_i(X_i)] - x(i, g_\ell) v_i(g_\ell) + \Delta \cdot v_i(g_\ell) + x(j, g'_u)\, v_i(G' \setminus g_\ell) + v_i(g_\ell) \left(\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \right) \tag{3.35}$$

$$= \mathbb{E}[v_i(X_i)] - x(j, g'_u)\, v_i(g_\ell) + x(j, g'_u)\, v_i(G' \setminus g_)) + v_i(g_\ell) \left(\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \right) \tag{3.36}$$

$$= \mathbb{E}[v_i(X_i)] - \underbrace{v_i(g_\ell)}_{\text{replace with } v_i(G' \setminus g_\ell)} \left[\underbrace{x(j, g'_u) - \frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k}}_{\geq 0} \right] + v_i(G' \setminus g_\ell)\, x(j, g'_u) \tag{3.37}$$

$$\leq \mathbb{E}[v_i(X_i)] + v_i(G' \setminus g_\ell) \cdot \frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \tag{3.38}$$

$$\leq \mathbb{E}[v_i(X_i)] + \frac{\mathbb{E}[v_i(X_i)]}{2 - x(i, g_\ell)} \cdot \frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \qquad \text{(using (3.11))} \tag{3.39}$$

$$\leq \frac{7}{6} \mathbb{E}[v_i(X_i)] \qquad \text{(refer Appendix 4)} \tag{3.40}$$

Yet again, i is $\frac{6}{7}$ ex-ante EF towards j.

Hence, *Uniformly Tailed SE* is ex-ante $\frac{6}{7}$ EF and ex-post (EFX+PO). $\square$

Proposition 3.2.1. *If exactly $n-1$ goods are completely consumed during Step 1, i.e, $M \setminus \{U \cup F\} = n-1$), the randomized allocation is ex-ante EF and ex-post EFX+PO.*

Proof. The main difference from the general case (handled by the previous proof) is that, now, exactly $n-1$ goods are completely exhausted in Step 1 of the Algorithm. Thus, in the matching sampled in Step 5, we have $|N_F| = k = 1$. Let h be the singular agent in N_F, and say it is allocated the fractional good g_h in this matching. Agent h would simply receive the entirety of the set $G' \setminus g_h$, since all goods in $F \setminus g_h$ remain unallocated after Phase 1.

Let's proceed by case analysis, similar to the previous proof. The reader may refer to the corresponding diagrams and notations from the previous proof.

Case 1: Case 1: $g'_u \notin F$

Case 2: Case 2: $g'_u \in F$ *but* $g_\ell \notin F$

The arguments for Case 1 and Case 2 proceed identically to the previous proof.

Case 3: $g'_u = g_\ell = g \in F$

Subcase 3(a): $x(i,g) < x(j,g)$
Let us define $x = 1 - x(j,g)$ and $\Delta = x(j,g) - x(i,g)$.

$$\begin{aligned}
\mathbb{E}[v_i(X_j) + v_i(Y_j)] &= v_i(X_j^f[0:x]) + x(j,g)\,[v_i(g) + v_i(G' \setminus g)] \qquad (\because k=1) &(3.41)\\
&= v_i(X_j^f[0:x]) + (x(i,g) + \Delta)\,[v_i(g) + v_i(G' \setminus g)] &(3.42)\\
&= \underbrace{v_i(X_j^f[0:x]) + \Delta\,[v_i(g) + v_i(G' \setminus g)]} + x(i,g)\,[v_i(g) + v_i(G' \setminus g)] &(3.43)\\
&\leq v_i(X_i^f[0:x+\Delta]) + x(i,g)\,[v_i(g) + v_i(G' \setminus g)] \qquad \text{(by lexicographic property)} &(3.44)\\
&= \mathbb{E}[v_i(X_i) + v_i(Y_i)] &(3.45)
\end{aligned}$$

Subcase 3(b): $x(i,g) \geq x(j,g)$
Define $x = 1 - x(i,g)$ and $\Delta = x(i,g) - x(j,g)$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] = v_i(X_j^f[0:x]) + \underbrace{v_i(X_j^f[x:x+\Delta])}_{\leq \Delta \cdot v_i(g)} + x(j,g)\,[v_i(g) + v_i(G' \setminus g)] \tag{3.46}$$

$$\leq v_i(X_i^f[0:x]) + \underbrace{\Delta \cdot v_i(g) + x(j,g)[v_i(g)} + v_i(G' \setminus g)] \tag{3.47}$$

$$= v_i(X_i^f[0:x]) + x(i,g)\,v_i(g) + x(j,g)\,v_i(G' \setminus g) \tag{3.48}$$

$$\leq v_i(X_i^f[0:x]) + x(i,g)\,v_i(g) + x(i,g)\,v_i(G' \setminus g) \qquad [\because x(i,g) \geq x(j,g)] \tag{3.49}$$

$$= \mathbb{E}[v_i(X_i) + v_i(Y_i)] \tag{3.50}$$

Thus, agent i is ex-ante EF towards agent j.

Case 4: $g'_u\,, g_\ell \in F\,, but\, g'_u \neq g_\ell$

Subcase 4(a): $x(i,g_\ell) < x(j,g'_u)$
Let $x = 1 - x(j,g'_u)$ and $\Delta = x(j,g'_u) - x(i,g_\ell)$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] = v_i(X_j^f[0:x]) + x(j,g'_u)\,[v_i(g'_u) + v_i(G' \setminus g'_u)] \qquad (\because k = 1) \tag{3.51}$$

$$= v_i(X_j^f[0:x]) + x(j,g'_u)\,v_i(G') = v_i(X_j^f[0:x]) + (x(i,g_\ell) + \Delta)\,v_i(G') \tag{3.52}$$

$$= \underbrace{v_i(X_j^f[0:x]) + \Delta \cdot v_i(G')}_{\leq v_i(X_i^f[x:x+\Delta])} + x(i,g_\ell)\,v_i(G') \tag{3.53}$$

$$\leq v_i(X_i^f[0:x+\Delta]) + x(i,g_\ell)v_i(G') \qquad \text{(by lexicographic property)} \tag{3.54}$$

$$\leq v_i(X_i^f[0:x+\Delta]) + x(i,g_\ell)[v_i(g_\ell) + v_i(G' \setminus g_\ell)] \tag{3.55}$$

$$= \mathbb{E}[v_i(X_i) + v_i(Y_i)] \tag{3.56}$$

Subcase 4(b): $x(i,g_\ell) \geq x(j,g'_u)$

Let $x = 1 - x(i, g_\ell)$ and $\Delta = x(i, g_\ell) - x(j, g'_u)$.

$$\mathbb{E}[v_i(X_j) + v_i(Y_j)] = v_i(X_j^f[0:x]) + v_i(X_j^f[x:x+\Delta]) + x(j, g'_u)\,[v_i(g'_u) + v_i(G' \setminus g'_u)] \tag{3.57}$$

$$= v_i(X_j^f[0:x]) + \underbrace{v_i(X_j^f[x:x+\Delta])}_{\leq \Delta \cdot v_i(g_\ell)} + x(j, g'_u)\,[v_i(g_\ell) + v_i(G' \setminus g_\ell)] \tag{3.58}$$

$$\leq v_i(X_j^f[0:x]) + \underbrace{\Delta \cdot v_i(g_\ell) + x(j, g'_u)\,v_i(g_\ell)} + x(j, g'_u)\,v_i(G' \setminus g_\ell) \tag{3.59}$$

$$\leq v_i(X_i^f[0:x]) + x(i, g_\ell)\,v_i(g_\ell) + \underbrace{x(j, g'_u)}_{\leq x(i, g_\ell)}\,v_i(G' \setminus g_\ell) \tag{3.60}$$

$$\leq v_i(X_i^f[0:x]) + x(i, g_\ell)\,v_i(g_\ell) + x(i, g_\ell)\,v_i(G' \setminus g_\ell) \tag{3.61}$$

$$= \mathbb{E}[v_i(X_i) + v_i(Y_i)] \tag{3.62}$$

Thus, agent i is ex-ante EF towards agent j.

□

While $k = 2$ gives us a ratio of $6/7$, the above analysis shows that as k increases, this ratio becomes closer to 1. This comparison between the case where $k = 1$ and $k > 1$ also gives us an insight into why the *Uniform Tailed Simultaneous Eating* fails to give exact EF.

When multiple agents could receive fractional goods in Phase 2, they could receive different subsets of $F \implies$ the tails one could receive might be different for different samplings. For instance, when $k = 2$, suppose that in matching M, agents i and j have a chance at receiving the tail, while in matching L, j and k do. It could be that i prefers the tail the other 2 agents might receive in L than the tail it receives in M. Furthermore, if the probabilities of sampling M is lesser than L, this worsens the envy.

The most obvious way to circumvent this issue, would be to uniformly assign the tail among every agent who has consumed a fractional good apriori to the matching. Indeed, this does achieve ex-ante EF, but, assigning the tail apriori could assign the tail to an agent envied in the sampled matching, thus losing ex-post EFX. One possible way to get around this, is to somehow force agents that ate comparable fractions of goods in F to receive similarly valued tails. For example, one could try weighting the probabilities of assigning the tail (rather than uniformly choose an agent) depending on how much the agent has consumed a fractional good.

Experimental Verification using Linear Programming

As a basic sanity check, for very small structured instances, we have verified the existence of ex-ante EF and ex-post (EFX+PO) allocations using LP simulations. The cardinal utilities of individual items are taken as consecutive powers of some small integer $k \geq 2$.

Theorem 3.2.2. *Any lexicographic goods instance with $n \leq 4$ and $m = n + 1$, where the utilities of individual items are consecutive powers of $2 \leq k \leq 10$, $k \in \mathbb{N}$, admits a randomized allocation that is ex-ante* EF *and ex-post* EFX+PO*.*

For every instance, we compute all EFX+PO allocations using Algorithm 6. We define our variables to be probability weights assigned to each EFX+PO allocation. We impose constraints on these weights so that for every pair of agents i, j, the expected valuation of i's bundle, according to agent i, is atleast the expected valuation of j's bundle according i.

Code for the simulations are available at: https://github.com/SuryaPanchapakesan/Fair-Division

3.3 Ex-ante EF and Ex-Post EFX for Chores

Moving to the chores setting, we present an algorithm that achieves ex-ante EF + ex-post EFX. However, as we shall see later, this approach does not guarantee ex-post efficiency (ex-post PO or fPO).

Theorem 3.3.1. *For every lexicographic chores instance, Algorithm 12 returns an allocation A that is ex-ante EF and ex-post EFX.*

Proof. Firstly, note that the algorithm takes time $O(n^{4.5})$. Once again, the slowest step is the B-vN Decomposition of the $n - 1$ sized square matrix. Every realization of Algorithm 12 is an allocation where a single agent is given $m - n + 1$ tasks while the remainder have

Algorithm 12:

Input: A lexicographic chores instance $\langle N, M, \mathcal{V} \rangle$
Output: A randomized allocation A

$A \leftarrow (\emptyset, \emptyset, ..., \emptyset)$
$i \leftarrow$ choose an agent uniformly at random from N
$X_i = \{c_1, c_2..., c_{m-n+1}\} \leftarrow$ agent i's least disliked $m-n+1$ chores in M
$A_i \leftarrow A_i \cup X_i$
$U \leftarrow$ an $n-1$ dimension square matrix with every entry as $1/n$
Arbitrarily label agents in $N \setminus i$ with rows of U and chores in $M \setminus X_i$ to the columns of U
$(Y^k, p_k)_{k \in [n-1]} \leftarrow$ B-vN Decomposition(U)
$Y \leftarrow$ a permutation matrix sampled from $(Y^k, p^k)_{k \in [n-1]}$
for *each agent* $j \in N \setminus i$ *and chore* $c \in M \setminus X_i$ **do**
  if $Y_{j,c} = 1$ **then**
    $A_j \leftarrow A_j + c$
return A

exactly 1 chore each. On decomposing the uniform fractional matrix U, we get $n-1$ distinct matchings, each sampled with a probability $^1/_{n-1}$. To show ex-post EFX: every realization is exactly the same as Algorithm 7. Hence, the proof follows.
To show ex-ante EF: Pick any agents $i, j \in N$. Let's look at i's envy towards j.

	i's bundle	**j's bundle**
i is picked	X_i	$g \in M \setminus X_i$
j is picked	$h \in M \setminus X_j$	X_j
neither are picked	some task c_i	some task c_j

Let $M \setminus X_i = \{g_1, g_2, ...g_{n-1}\}$ and $M \setminus X_j = \{h_1, h_2, ...h_{n-1}\}$. See that when neither i nor j are picked, both agents will get the same expected bundle. It suffices to show that, according to i, the expected valuation of the bundle of i is more than expected valuation of the bundle of j with respect just the firs two cases.

$$\mathbb{E}^*[v_i(A_i)] = \frac{1}{n}v_i(X_i) + \frac{1}{n}\frac{\sum_{i=1}^{n-1} v_i(h_i)}{n-1}$$
$$E^*[v_i(A_j)] = \frac{1}{n}v_i(X_j) + \frac{1}{n}\frac{\sum_{i=1}^{n-1} v_i(g_i)}{n-1}$$

It suffices to show: $\frac{1}{n}v_i(X_i) - \frac{1}{n}v_i(X_j) \geq \frac{1}{n}\frac{\sum_{i=1}^{n-1} v_i(g_i)}{n-1} - \frac{1}{n}\frac{\sum_{i=1}^{n-1} v_i(h_i)}{n-1}$

X_i is the top $m-n+1$ least hated chores of i and therefore, weakly prefers X_i to any set of $m-n+1$ chores. Also, the set $G = \{g_i\}_{i\in[n-1]}$ are the most hated chores of i, and thus weakly prefers any set of $n-1$ chores to G. It follows that the LHS is non-negative and RHS is non-positive. Hence, i is ex-ante EF towards j. □

Proposition 3.3.1. *Algorithm 12 fails ex-post PO.*

Proof. After the chosen agent has been assigned its least hated chores, any of the remaining chores could go to any agent with equal probability irrespective of the agents' preferences. Hence, the support could have allocations that are not sequencible.

Consider an example with 3 agents such that

$$v_1 : a \succ b \succ c$$
$$v_2 : a \succ b \succ c$$
$$v_3 : a \succ c \succ b$$

When v_1 is chosen, chore a is given. Then the uniform matrix is decomposed as:

$$U = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} = 0.5\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 0.5\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

In the second matching, both agents do not get their least hated chore, and thus is not sequencible.

□

One natural approach to tweak this algorithm to achieve ex-post PO, would be to replace the uniform distribution of $n-1$ chores among $n-1$ agents, to some other strategy that respects the agent preferences, such as the SE mechanism. It is easy to see that the allocation returned is ex-post EFX+PO. However, it loses ex-ante EF. Infact, we cannot guarantee any approximate ex-ante EF as well. See that for chores, α-EF is defined for $\alpha \geq 1$ (The other agent's disutility is scaled up before comparing the bundles).

Proposition 3.3.2. *Modifying Algorithm 12 by replacing the Uniform Matrix with the fractional allocation returned by* SE *gives a randomized allocation that is* ex-post EFX + PO*, but fails* ex-ante EF*. Moreover, given any $\alpha \geq 1$, there exists an instance on which the randomized allocation cannot do better than* ex-ante α-EF.

Proof. Consider an instance with 3 agents and preference profiles:

$$
\begin{aligned}
i &: \; a_1 \succ a_2 \succ b_1 \succ c \succ b_2 \\
j &: \; a_1 \succ a_2 \succ c \succ b_1 \succ b_2 \\
k &: \; b_1 \succ b_2 \succ c \succ a_1 \succ a_2
\end{aligned}
$$

	a_1	a_2	c	b_1	b_2
i	$-\frac{\epsilon}{4}$	$-\frac{\epsilon}{2}$	-2ϵ	$-\epsilon$	$-K$

Table 3.3: Here, $K \geq 1$ is sufficiently large and $0 < \epsilon << 1$

Let us look at i's envy towards j.
When i is chosen, it receives $\{a_1, a_2, b_1\}$ and j eats the whole of c. When j is chosen, it receives $\{a_1, a_2, c\}$ while i eats $b_1/2$ and $b_2/2$. When k is chosen, i and j both eat $a_1/2$ and $a_2/2$.

Thus,

$$\frac{\mathbb{E}[v_i(A_i)]}{\mathbb{E}[v_i(A_j)]} = \frac{v_i(a_1+a_2+b_1) + \dfrac{v_i(b_1+b_2)}{2} + \dfrac{v_i(a_1+a_2)}{2}}{v_i(a_1+a_2+c) + v_i(c) + \dfrac{v_i(a_1+a_2)}{2}}$$

$$= \frac{\dfrac{3\epsilon}{8} + \dfrac{3\epsilon}{4} + \dfrac{3\epsilon}{2} + \dfrac{K}{2}}{\dfrac{3\epsilon}{8} + \dfrac{3\epsilon}{4} + 4\epsilon}$$

$$\leq \frac{4\epsilon + K/2}{4\epsilon}$$

Taking K to be very large $\implies$ the ratio is unbounded.

□

3.4 Ex-ante PROP and Ex-post EFX+PO

In this final section, we show that on relaxing our ex-ante fairness guarantees, one can compute an allocation that is ex-ante PROP along with ex-post (EFX+PO) for both goods and chores.

Similar to *Randomized Round Robin* and *Uniform Random Permutation*, we choose a set of EFX+PO allocations and randomize over them. For the goods scenario, we choose an ordering to instantiate Algorithm 5, but rather than randomizing over all permutations, we restrict the pick to just the *cyclic permutations* of the agents. For the chores setting, we instantiate Algorithm 7 with a permutation chosen the same way.

Algorithm 13:

Input: A lexicographic goods-only or chores-only instance $\langle N, M, \mathcal{V}\rangle$

Output: A randomized allocation A

$A \leftarrow (\emptyset, \emptyset, ..., \emptyset)$
$\mathcal{P}_{cycle} \leftarrow$ set of **cyclic** permutation of agents in N
$\sigma \leftarrow$ a cyclic permutation chosen uniformly at random from $\mathcal{P}_{cycle}$
Relabel the agents from $1, 2, ..., n$ based on their positions in σ
if *M is a set of chores* **then**
  $C \leftarrow$ agent n's least hated $m - n + 1$ chores in M
  $A_n \leftarrow A_n \cup C$
  for *i from $n-1$ to 1* **do**
    $c_i \leftarrow$ agent i's least hated *available* chore
    $A_i \leftarrow A_i + c_i$
if *M is a set of goods* **then**
  for *i from 1 to n* **do**
    $g_i \leftarrow$ agent i's top *available* good
    $A_i \leftarrow A_i + g_i$
  $R \leftarrow$ unallocated goods in M
  $A_n \leftarrow A_n \cup R$
return A

Theorem 3.4.1. *For any lexicographic goods or chores instance, Algorithm 13 returns an allocation A that is* ex-ante PROP *and* ex-post (EFX+PO)

Proof. We show for the goods setting first. Since every realization is an instantiation of Algorithm 5, it immediately follows that A is ex-post EFX+PO. To show ex-ante PROP: Given an agent i, it occupies any position of σ with $1/n$ probability. Let g_k denote its k^{th} favourite good. When i is the k^{th} agent, i receives a good that it weakly prefers to g_k. Thus, $\mathbb{E}[v_i(A_i)] \geq \dfrac{1}{n} \sum_{k=1}^{m} v_i(g_k) \geq \dfrac{v_i(M)}{n}$ □

The same arguments give the required result in the chores case as well. Moreover, it is easy to see that the above result is strengenthed to ex-ante EF and ex-post (EFX+PO) when the input is an *ordered* instance.

Chapter 4

Conclusion, Open Problems and Future Directions

The fair division of indivisible items has been an exciting problem of interest, and has garnered significant attention especially in the past 10 years. In this thesis, we took a look at some of the fundamental algorithmic techniques such as the envy cycle procedure and the SE mechanism among others, and studied their guarantees and limitations. We studied the setting where agents have lexicographic preferences - one of the few domains where EFX allocations can be guaranteed. We then revisited key ideas from Aziz *et al.* [13], which established the first BoBW results for additive valuations. In Chapter 3, we presented our original attempts to strengthen these guarantees to ex-post EFX (and PO) in the lexicographic setting. Our main algorithm *Uniform Tailed Simultaneous Eating* computes an ex-ante $^6/_7$ -EF and ex-post EFX+PO for lexicographic goods.

We believe that the approximation ratio of 6/7 can be significantly improved with a refined approach. As discussed in Section 3.2, a key challenge is ensuring that agents consuming comparable portions of the same fractionally assigned goods receive tails that are as similar as possible. Rather than assigning the tail in one go, one could also allow a subset of unenvied agents to consume the remaining goods after Phase 1, such as allocating goods via two rounds of SE.

One promising avenue for improvement is replacing the B-vN Decomposition step with a dependent rounding technique, such as the approach by Gandhi *et al.* [38]. Here, the probabilities of agents receiving additional goods are negatively correlated, which is not the case when sampled via B-vN decomposition. We believe that adopting this strategy may help push beyond the $^6/_7$ barrier.

Recall that the *PS-Lottery* Algorithm [13] computes an allocation that is ex-ante SD-EF and ex-post EF1. In a similar vein, we define a restricted notion of SD-EF, termed *Lexicographic SD-envyfreeness* (Lex-SD-EF), which, unlike SD-EF requires the SD-EF property to hold for not all additive instances consistent with the rankings of the individual items, but only for those additive instances that are consistent with the lexicographic ordinal preferences over the entire set of bundles. While it is known that ex-ante SD-EF and ex-post EFX need not always exist, it remains to be seen if ex-ante Lex-SD-EF can always be attained alongside ex-post EFX.

Open Problems

We now present some closely related open problems that stem from the central questions studied in this thesis.

1. Deterministic allocations:
   (a) Given a lexicographic goods instance, all EFX + PO allocations can be characterized by the family of algorithms in Theorem 2.2.3. It is still not known if such a characterization is possible for the chores setting.
2. Randomized allocations:
   (a) Given a lexicographic goods instance, can we efficiently compute a randomized allocation that is (exact) ex-ante EF and ex-post EFX? Additionally, can we achieve ex-post PO as well?
   (b) Does every lexicographic instance admit a randomized allocation that is ex-ante Lex-SD-EF and ex-post EFX?
   (c) Does every additive instance admit a randomized allocation that is ex-ante EF and ex-post (EF1 + PO)?

Bibliography

[1] D. Foley, "Resource allocation and the public sector," Yale Economic Essays, vol. 7, no. 1, pp. 45–98, 1967.

[2] H. R. Varian, "Equity, envy, and efficiency," Journal of Economic Theory, vol. 9, no. 1, pp. 63 – 91, 1974.

[3] Haris Aziz, Simon Mackenzie, A discrete and bounded envy-free cake cutting protocol for any number of agents, in: Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS), 2016, pp. 416–427.

[4] Noga Alon, Splitting necklaces, Advances in Mathematics, Volume 63, Issue 3, 1987, Pages 247-253, ISSN 0001-8708,

[5] E. Budish, "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes," Journal of Political Economy, vol. 119, no. 6, pp. 1061 – 1103, 2011.

[6] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. 2016. The Unreasonable Fairness of Maximum Nash Welfare. In Proceedings of the 2016 ACM Conference on Economics and Computation (EC '16). Association for Computing Machinery, New York, NY, USA, 305–322.

[7] Benjamin Plaut and Tim Roughgarden. 2020. Almost Envy-Freeness with General Valuations. SIAM J. Discret. Math. 34, 2 (January 2020), 1039–1068.

[8] Lipton, Richard J., Evangelos Markakis, Elchanan Mossel, and Amin Saberi. "On approximately fair allocations of indivisible goods." In Proceedings of the 5th ACM Conference on Electronic Commerce, pp. 125-131. 2004.

[9] Hosseini, Hadi, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2021. "Fair and Efficient Allocations under Lexicographic Preferences". Proceedings of the AAAI Conference on Artificial Intelligence 35 (6):5472-80.

[10] Xiaolin Bu, Jiaxin Song, and Ziqi Yu. 2023. EFX Allocations Exist for Binary Valuations. In Frontiers of Algorithmics: 17th International Joint Conference, IJTCS-FAW

2023 Macau, China, August 14–18, 2023 Proceedings. Springer-Verlag, Berlin, Heidelberg, 252–262.

[11] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2023. Fairly Dividing Mixtures of Goods and Chores under Lexicographic Preferences. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 152–160.

[12] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2019. Equitable allocations of indivisible goods. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19). AAAI Press, 280–286.

[13] Haris Aziz, Rupert Freeman, Nisarg Shah, and Rohit Vaish. 2024. Best of Both Worlds: Ex Ante and Ex Post Fairness in Resource Allocation. Oper. Res. 72, 4 (July-August 2024), 1674–1688.

[14] Bogomolnaia A, Moulin H (2001) A new solution to the random assignment problem. J. Econom. Theory 100:295–328

[15] Birkhoff G (1946) Three observations on linear algebra. Universidad Nacional Tucum´an Rev. A 5:147–151

[16] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. 2024. EFX Exists for Three Agents. J. ACM 71, 1, Article 4 (February 2024), 27 pages.

[17] Vishwa Prakash H.V., Pratik Ghosal, Prajakta Nimbhorkar, Nithin Varma. 2024. EFX Exists for Three Types of Agents

[18] Steinhaus, "The problem of fair division," Econometrica, vol. 16, no. 1, pp. 33–111, 1948.

[19] Benny Lehmann, Daniel Lehmann, and Noam Nisan. 2001. Combinatorial auctions with decreasing marginal utilities. In Proceedings of the 3rd ACM conference on Electronic Commerce (EC '01). Association for Computing Machinery, New York, NY, USA, 18–28.

[20] Michal Feldman, Simon Mauras, Vishnu V. Narayan, and Tomasz Ponitka. 2024. Breaking the Envy Cycle: Best-of-Both-Worlds Guarantees for Subadditive Valuations. In Proceedings of the 25th ACM Conference on Economics and Computation (EC '24). Association for Computing Machinery, New York, NY, USA, 1236–1266.

[21] Jugal Garg and Eklavya Sharma. Best-of-both-worlds fairness of the envy-cycle-elimination algo- rithm. arXiv:2410.08986, 2024.

[22] Jon Kleinberg and Eva Tardos. 2005. Algorithm Design. Addison-Wesley Longman Publishing Co., Inc., USA.

[24] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. In Proceedings of the 24th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), pages 1:1–1:23, 2021.

[25] Christodoulou, G., Fiat, A., Koutsoupias, E., Sgouritsa, A.: Fair allocation in graphs. In: Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, pp. 473–488. Association for Computing Machinery, New York (2023)

[26] Misra, N., Sethia, A. (2025). Envy-Free and Efficient Allocations for Graphical Valuations. In: Freeman, R., Mattei, N. (eds) Algorithmic Decision Theory. ADT 2024. Lecture Notes in Computer Science(), vol 15248. Springer, Cham.

[27] Procaccia, Ariel (2009). "Thou Shalt Covet Thy Neighbor's Cake". IJCAI'09 Proceedings of the 21st International Joint Conference on Artificial Intelligence: 239–244

[28] H. Aziz. 2019. A Probabilistic Approach to Voting, Allocation, Matching, and Coalition Formation. In The Future of Economic Design, J.-F. Laslier, H. Moulin, R. Sanver, and W. S. Zwicker (Eds.). Springer-Verlag.

[29] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2022. On Best-of-Both-Worlds Fair-Share Allocations. In Web and Internet Economics: 18th International Conference, WINE 2022, Troy, NY, USA, December 12–15, 2022, Proceedings. Springer-Verlag, Berlin, Heidelberg, 237–255.

[30] Haris Aziz, Aditya Ganguly, and Evi Micha. 2023. Best of Both Worlds Fairness under Entitlements. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23).

[31] F. Kojima. Random Assignment of Multiple Indivisible Objects. Mathematical Social Sciences, 57(1):134– 142, 2009.

[32] J. v. Neumann. A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem. In W. Kuhn and A. W. Tucker, editors, Contributions to the Theory of Games, volume 2, pages 5–12. Princeton University Press, 1953.

[33] Hopcroft, J. E., Karp, R. M., 1973. An n5/2 algorithm for maximum match- ings in bipartite graphs. SIAM J. Comput. 2 (4), 225–231.

[34] Vazirani, Vijay V. (2020-10-14). "An Extension of the Birkhoff-von Neumann Theorem to Non-Bipartite Graphs"

[35] Valls, Victor; Iosifidis, Georgios; Tassiulas, Leandros (Dec 2021). "Birkhoff's Decomposition Revisited: Sparse Scheduling for High-Speed Circuit Switches"

[36] Budish, Eric; Che, Yeon-Koo; Kojima, Fuhito; Milgrom, Paul (2013-04-01). "Designing Random Allocation Mechanisms: Theory and Applications". American Economic Review. 103 (2): 585–623

[37] Johnson, Diane M.; Dulmage, A. L.; Mendelsohn, N. S. (1960-09-01). "On an Algorithm of G. Birkhoff Concerning Doubly Stochastic Matrices"

[38] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. 2006. Dependent rounding and its applications to approximation algorithms. J. ACM 53, 3 (May 2006), 324–360. https://doi.org/10.1145/1147954.1147956

[39] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, Xiaowei Wu, Fair division of indivisible goods: Recent progress and open questions, Artificial Intelligence, Volume 322, 2023, 103965, ISSN 0004-3702,

[40] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In Proceedings of the 2018 ACM Conference on Economics and Computation (EC '18). Association for Computing Machinery, New York, NY, USA, 557–574.

[41] Georgios Amanatidis, Evangelos Markakis, Apostolos Ntokos, Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination, Theoretical Computer Science, Volume 841, 2020, Pages 94-109, ISSN 0304-3975,

[42] Ryoga Mahara, Existence of EFX for two additive valuations, Discrete Applied Mathematics, Volume 340, 2023, Pages 115-122, ISSN 0166-218X,

[43] Hadi Hosseini, Aghaheybat Mammadov, and Tomasz Was. 2023. Fairly allocating goods and (terrible) chores. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI '23). Article 305, 2738–2746.

Appendix

We now revisit some parts of the proof of *Uniform Tailed Simultaneous Eating* that were omitted.

Theorem 4.0.1. ***Subcase 4.a*** *(eqn. (3.31) in the proof):*

$$\mathbb{E}[v_i(X_i)] + \frac{\mathbb{E}[v_i(X_i)]}{2 - x(i, g_\ell)}\left(x(i, g_\ell) - x(j, g'_u) + \frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k}\right) \leq \frac{7}{6}\mathbb{E}[v_i(X_i)]$$

Proof. As $k \geq 2$, it suffices to show $\dfrac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{2}\right)}{2 - x(i, g_\ell)} \leq 1/6$

- **Case A:** $x(j, g'_u) \leq 1 - x(i, g_\ell)$

$$\frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{2}\right)}{2 - x(i, g_\ell)} \tag{4.1}$$

$$= \frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{x(j, g'_u)}{2}\right)}{2 - x(i, g_\ell)} \tag{4.2}$$

$$= \frac{x(i, g_\ell) - \dfrac{x(j, g'_u)}{2}}{2 - x(i, g_\ell)} \tag{4.3}$$

$$\leq \frac{x(i, g_\ell) - \dfrac{x(i, g_\ell)}{2}}{2 - x(i, g_\ell)} \qquad [\because x(i, g_\ell) \leq x(j, g'_u)] \tag{4.4}$$

$$= \frac{x(i, g_\ell)}{4 - 2x(i, g_\ell)} \tag{4.5}$$

Note that $x(i, g_\ell) \leq 1 - x(j, g'_u) \quad \& \quad x(i, g_\ell) \leq x(j, g'_u) \implies 0 < x(i, g_\ell) \leq 0.5$

Thus, the expression in eqn. (4.5) attains a maxima of $\frac{1}{6}$ at $x(i, g_\ell) = 0.5$

- **Case B:** $x(j, g'_u) \geq 1 - x(i, g_\ell)$

  Observe that in this case, we have $x(j, g'_u) \geq 0.5$ and $x(j, g'_u) + x(i, g_\ell) = c$ for some $1 \leq c \leq 2$. To analyze the function, we examine its behavior along the line $x(j, g'_u) + x(i, g_\ell) = c$ for some fixed c.

$$\frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{2}\right)}{2 - x(i, g_\ell)} \tag{4.6}$$

$$= \frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{1 - x(i, g_\ell)}{2}\right)}{2 - x(i, g_\ell)} \tag{4.7}$$

$$= \frac{\left(x(i, g_\ell) - c + x(i, g_\ell) + \dfrac{1 - x(i, g_\ell)}{2}\right)}{2 - x(i, g_\ell)} \tag{4.8}$$

$$= \frac{1 - 2c + 3x(i, g_\ell)}{4 - 2x(i, g_\ell)} \tag{4.9}$$

$$= \frac{7 - 2c}{4 - 2x(i, g_\ell)} - \frac{3}{2} \tag{4.10}$$

Since this function is increasing in $x(i, g_\ell)$, and we are constrained by the line $x(i, g_\ell) + x(j, g'_u) = c$, the function is maximized when $x(i, g_\ell) = x(j, g'_u)$ given that we are in the regime where $x(i, g_\ell) \leq x(j, g'_u)$)

$$\therefore \frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{1 - x(i, g_\ell)}{2}\right)}{2 - x(i, g_\ell)} = \frac{1 - x(i, g_\ell)}{4 - 2x(i, g_\ell)} = \frac{1}{2} - \frac{1}{4 - 2x(i, g_\ell)} \tag{4.11}$$

This grows as $x(i, g_\ell)$ decreases, and since $x(i, g_\ell) = x(j, g'_u) \geq 0.5$, the function is maximized to $\frac{1}{6}$ at $x(i, g_\ell) = x(j, g'_u) = 0.5$

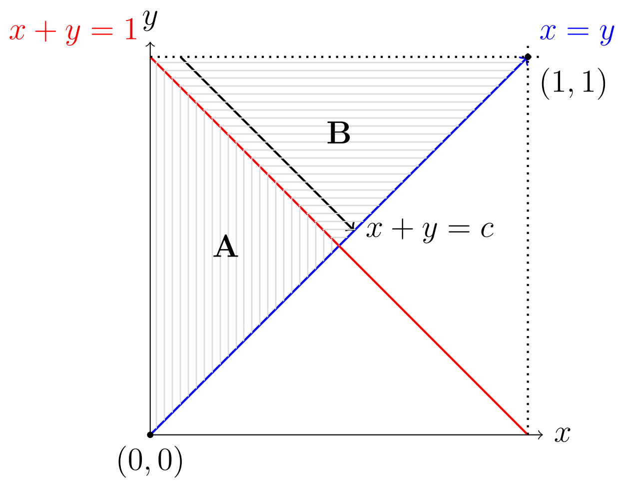

Figure 4.1: The variables are defined as $y = x(j, g'_u)$ and $x = x(i, g_\ell)$
The region labeled **A** corresponds to **Case A**, and similarly, **B** represents **Case B**.

Thus, $$\frac{\left(x(i, g_\ell) - x(j, g'_u) + \dfrac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k}\right)}{2 - x(i, g_\ell)} \leq \frac{1}{6} \qquad (\because k \geq 2)$$ □

.

Theorem 4.0.2. *Subcase 4.(a) (eqn. (3.39) in the proof):*

$$\mathbb{E}[v_i(X_i)] + \frac{\mathbb{E}[v_i(X_i)]}{2 - x(i, g_\ell)} \cdot \frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{k} \leq \frac{7}{6}\mathbb{E}[v_i(X_i)]$$

Proof. Note that we are in the regime where $x(i, g_\ell) \geq x(j, g'_u)$. It suffices to show that:

$$\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{2(2 - x(i, g_\ell))} \leq \frac{\min\{x(i, g_\ell),\, 1 - x(i, g_\ell)\}}{2(2 - x(i, g_\ell))} \leq \frac{1}{6} \qquad (\because k \geq 2)$$

- **Case A:** $x(i, g_\ell) \leq 0.5$

$$\therefore \frac{\min\{x(i, g_\ell),\, 1 - x(i, g_\ell)\}}{2(2 - x(i, g_\ell))} = \frac{x(i, g_\ell)}{4 - 2x(i, g_\ell)} \tag{4.12}$$

This is increasing in $x(i, g_\ell)$ and is maximized to $\frac{1}{6}$ at $x(i, g_\ell) = 0.5$

- **Case B:** $x(i, g_\ell) \geq 0.5$

$$\frac{\min\{x(i, g_\ell),\, 1 - x(i, g_\ell)\}}{2(2 - x(i, g_\ell))} \tag{4.13}$$

$$= \frac{1 - x(i, g_\ell)}{4 - 2x(i, g_\ell)} \tag{4.14}$$

$$= \frac{y}{2(y+1)} \qquad \text{(where } y = 1 - x(i, g_\ell) \leq 0.5\text{)} \tag{4.15}$$

This is increasing in y and is maximized to $\frac{1}{6}$ at $y = 0.5$

Thus, $\frac{\min\{x(j, g'_u),\, 1 - x(i, g_\ell)\}}{2(2 - x(i, g_\ell))} \leq \frac{1}{6}$ and the rest of the proof follows. $\square$