Harvard CS 1360, Spring 2025
Instructor: Ariel Procaccia
Scribe: Rhea Acharya

Fair Division 3: Indivisible Goods

—

Lecture 12

1 Introduction

An indivisible goods allocation problem involves:

- A set G of m indivisible goods.
- A set of players $N = \{1, \ldots, n\}$.
- Valuations V_i for each player, where valuations are additive unless stated otherwise.
- Valuations are additive if for all $S \subseteq G$ and $i \in N$, $V_i(S) = \sum_{g \in S} V_i(g)$
- An allocation is a partition of the goods, denoted $\mathbf{A} = (A_1, \ldots, A_n)$

The core challenge with indivisible goods is that envy-freeness and proportionality are generally infeasible. Unlike divisible settings (like cake cutting), we cannot simply divide each good to ensure everyone gets their proportional share.

Note for practical application: Allocation of indivisible goods appears in many real-world scenarios: dividing inheritance items among heirs, allocating tasks among team members, assigning dormitory rooms to students, and distributing computing resources in data centers. The fundamental challenge is that we often cannot achieve perfect fairness, so we need to explore approximations that maintain some fairness properties.

2 Maximin Share Guarantee

2.1 Definition

The Maximin Share (MMS) guarantee of player i is given by:

$$\max_{X_1, \ldots, X_n} \min_j V_i(X_j) \tag{1}$$

Intuitively, the MMS of player i represents the maximum value they could guarantee themselves if they were to partition goods into n bundles and then receive the worst bundle (i.e., "you cut, adversary chooses"). An MMS allocation ensures that $V_i(A_i) \geq$ MMS guarantee for all $i \in N$.

Intuition: Think of the Maximin Share guarantee as answering the question: "What's the best I could do if I had to divide the items into n piles, knowing I'll get the least valuable pile?" It captures a concept of fairness that's weaker than proportionality but can often be achieved with indivisible goods. The intuition comes from imagining you're dividing a set of items among people, including yourself, but you know you'll be the last to choose.

2.2 Existence Theorems

- For $n = 2$, an MMS allocation always exists.
- Theorem: For $n \geq 3$, there exist additive valuation functions that do not admit an MMS allocation.

Counterexample Construction

Let us examine a counterexample for $n = 3$ players showing that MMS allocations do not always exist. Before we even define any values, we will start with a weird combinatorial construction:

17	25	12	1
2	22	3	28
11	0	21	23

Now, we claim that there are exactly 3 ways of dividing these numbers into 3 subsets of 4 numbers each such that the number in each subset adds up to 55. The total sum of the whole grid is 165.

Specifically, we can divide it into the following 3 configurations:

17	25	12	1
2	22	3	28
11	0	21	23

17	25	12	1
2	22	3	28
11	0	21	23

17	25	12	1
2	22	3	28
11	0	21	23

One can verify that each subset of 4 numbers within the grid (which are colored a specific color) add up to 55. No other partitions of this particular grid exist that satisfy the equal groups adding up to an equal sum property.

With this in mind, we will present our counterexample. We have 12 goods with values determined by three components:

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times 10^6}_{\text{base valuation}} + \underbrace{\begin{bmatrix} 17 & 25 & 12 & 1 \\ 2 & 22 & 3 & 28 \\ 11 & 0 & 21 & 23 \end{bmatrix} \times 10^3}_{\text{second-order values}} + \underbrace{\text{player-specific perturbations}}_{\text{small adjustments}} \tag{2}$$

Each player's valuation function is:

$$V_i(g) = 10^6 \cdot (\text{base value}) + 10^3 \cdot (\text{second-order value}) + (\text{player-specific perturbation})$$

The player-specific perturbation matrices are:

3	-1	-1	-1
0	0	0	0
0	0	0	0

Table 1: Player One

3	-1	0	0
-1	0	0	0
-1	0	0	0

Table 2: Player Two

3	0	-1	0
0	0	-1	0
0	0	0	-1

Table 3: Player Three

To explain how these valuations work, let us consider the good that corresponds to the top right corner of the matrix:

A	B	C	D
E	F	G	H
I	J	K	L

Then, we can use the equation to determine each of the players will value the good as follows:

- Player 1: $1 \cdot 10^6 + 1 \cdot 10^3 - 1$
- Player 2: $1 \cdot 10^6 + 1 \cdot 10^3 + 0$
- Player 3: $1 \cdot 10^6 + 1 \cdot 10^3 + 0$

We see that first two parts of the calculation are the same; the players only differ in the final perturbation value. Since the magnitude of the first two components is scaled up so much, the valuations are broadly similar but just different at the end.

Note how this mirrors the construct of the three equal partitions from the initial fair division construction. Each of the partitions maps to one of these player's perturbation matrices, such that within that partition the perturbations values of the subsets all sum to 0.

We claim that the maximin share of each player is $4,055,000$. Each player can achieve this by dividing the goods according the partition shown in their perturbation matrix; for each bundle, the total value will be $4,000,000 + 55,000 + 0 = 4,055,000$, as the numbers in the second matrix add up to 55, and the numbers in the third matrix add up to 0.

But it is impossible to guarantee a value of $4,055,000$ to all players simultaneously. Indeed, since the magnitudes of the differences are so different between the components of the valuation calculation, we want each stage to be split equally, as we will not be able to make up for substantial differences in later stages. Thus, for each player, the base valuation of these four goods will sum to be $4 \cdot 10^6$. Since all of the cells are the same in this matrix, we can split it any way we want (into 3 groups of 4) and the valuation will be equal. Then, since the second-order values parallel the original matrix, these component's values will sum to $55 \cdot 10^3$. So we will have to split according to one of the special partitions, which we have colored earlier. Finally, the sum of all perturbation values for each player is zero, so in an ideal partition, the perturbation component would contribute 0 to each player's total.

But here's the problem. Let's say we pick the partition corresponding to player one's perturbation matrix. For the pink and the yellow subsets (corresponding to the second and third rows), both player 2 and 3 have a perturbation sum of -1 for those two rows. For the top row, they both have a sum of 2 for these rows. This means that even if we assign player 1 to rows 2 or 3 (since all the rows have an equal perturbation sum for them), then at least one of player 2 or player 3 will also have to have one of these "suboptimal" rows, resulting in a total valuation of $4,054,999$ for them, which is below the maximin share guarantee. A similar tension results if we pick the partition of player 2 or player 3, where we find that one of the other two players can't get their maximin share guarantee.

Note for in practice: The notion of Maximin guarantees only came about in the 2010's and it took about 4 years to get to this counterexample. In practice, maximin share guarantee is usually feasible, but it's not something that you can guarantee theoretically, as a hard fairness guarantee.

Later work has shown that we can always find allocations that provide each player at least $\frac{3}{4}$ of their MMS, and in many realistic settings, full MMS allocations do exist. This is a pattern we often see: theoretical worst-case results might be negative, but practical instances tend to behave much better.

3 Approximate Envy-Freeness

Since perfect envy-freeness can be impossible with indivisible items, we focus on approximate notions. We first assume general monotonic valuations, meaning for all $S \subseteq T \subseteq G$ and $i \in N$, $V_i(S) \leq V_i(T)$.

An allocation $(A_1, \ldots, A_n)$ is **Envy-Free up to one good (EF1)** if:

$$\forall i, j \in N, \exists g \in A_j \text{ such that } V_i(A_i) \geq V_i(A_j \setminus \{g\}) \tag{3}$$

Note: We can read this expression as saying that there is always a good, which we will call g, in the bundle of player j such that if we remove it from A_j, then i is no longer envious. In simple terms: "I might envy what you have, but if you remove your best item, I'd be satisfied with my allocation."

This is a natural relaxation of envy-freeness for indivisible goods - it captures the intuition that small differences shouldn't matter too much. In practice, it's often sufficient for maintaining social harmony, as people tend to be willing to overlook minor discrepancies.

Theorem: An EF1 allocation always exists and can be computed in polynomial time, even under very loose assumptions about the valuation functions.

To prove this theorem, we work with partial allocations and examine the structure of the resulting envy graphs.

3.1 Envy Graphs and EF1 Allocations

We have a partial allocation A of the goods, which is where only a subset of the goods have been allocated. From this allocation A, we can construct an envy graph, where we draw directed edges from (i, j) if i envies j. This graph shows which players are envious of which other players. Formally, we say that this envy graph can be denoted as (N, E), where N is the set of nodes that represent the players and E is the set of directed edges that indicate the presence of envy between two players.

Lemma: An EF1 partial allocation A can be transformed in polynomial time into an EF1 partial allocation B of the same goods with an acyclic envy graph.

Proof. The proof works by showing that we can eliminate cycles in the envy graph while maintaining the EF1 property.

If the envy graph has a cycle C, we shift allocations along C to obtain a new allocation A'. Specifically, if there is a cycle $(i_1, i_2, \ldots, i_k, i_1)$ where each player envies the next player in the cycle, we can reassign bundles so that player i_j gets the bundle previously held by player i_{j+1}, and player i_k gets the bundle previously held by player i_1.

This shifting process clearly maintains the EF1 property because we're just performing a permutation of the bundles among a subset of players, and the value players have for their own bundles can only increase.

The key insight is that the number of edges in the envy graph of A' decreases:

- Edges between $N \setminus C$ (players not in the cycle) remain the same. This is because players outside the cycle keep their bundles and valuations unchanged, so their envy relationships with each other don't change.
- Edges from $N \setminus C$ to C shift but their count stays the same. This is because outside players still envy the same bundles, but those bundles now belong to different players. The envy follows the bundles, so the total number of these edges stays constant.
- Edges from C to $N \setminus C$ can only decrease (because players in C now have bundles they valued more before). This is because cycle players get better bundles after the permutation, so they may envy fewer outside players. Their increased satisfaction means they won't develop new envy toward outside players
- Most importantly, edges inside the cycle C must decrease by at least one, because every player in the cycle receives a strictly more preferred bundle (as the cycle was structured based on cyclical envy, where each player envied the bundle of the next)

By iteratively removing cycles this way, we arrive at an acyclic envy graph while maintaining the EF1 property. □

Given the above lemma, we can now complete the proof of the main theorem:

Proof of Theorem. We use an incremental approach:

1. Start with an empty allocation where each player has received nothing. This trivially satisfies EF1 and has an acyclic envy graph.
2. In round 1, allocate good g_1 to an arbitrary player. The envy graph remains acyclic (possibly with some edges now), and the allocation is EF1.
3. Suppose goods $g_1, \ldots, g_{k-1}$ are allocated in an acyclic and EF1 allocation A.
4. To allocate good g_k, find a source i in the envy graph (a player that no one envies, which must exist because the graph is acyclic).
5. Give g_k to player i, resulting in allocation B.

6. For any player $j \neq i$, we have $V_j(B_j) = V_j(A_j) \geq V_j(A_i) = V_j(B_i \setminus \{g_k\})$, so the allocation remains EF1.

7. If new cycles appear in the envy graph, use the lemma to eliminate them.

This algorithm terminates after allocating all goods and yields an EF1 allocation. □

Algorithm insight: The beauty of this algorithm is that it always gives the next item to someone who isn't currently envied by anyone. This is a clever approach because it ensures we won't create too much new envy with each allocation step. The process of selecting a "source" player in the acyclic envy graph guarantees that we're maintaining balance in how desirable the bundles are.

The cycle elimination procedure is similar to techniques used in matching markets and trade networks, where cycles of desire (or envy) can be resolved by appropriate reassignments.

4 Round-Robin and Efficiency

4.1 Simple EF1 Mechanisms

When we return to the case of additive valuations, proving the existence of an EF1 allocation becomes much simpler. In fact, we can use a simple round-robin procedure:

- Order the players arbitrarily: $1, 2, \ldots, n$
- In rounds $r = 1, 2, \ldots$, player ($r \bmod n$) picks their favorite remaining good

This round-robin allocation always ensures an EF1 allocation. The intuition is that player i cannot envy player j by more than one good, since if j chose before i in one round, i will choose before j in the next round.

Within each phase, we will ensure EF1 allocation, and then since the valuations are additive, we can extend the claims across rounds.

Real-world application: The round-robin procedure is widely used in practice due to its simplicity and transparency. Examples include fantasy sports drafts, allocating dormitory rooms in colleges, and even children taking turns selecting toys. Its fairness guarantees (specifically EF1) provide a theoretical justification for why this procedure feels fair to participants.

However, round-robin does not guarantee Pareto efficiency, meaning there could be alternative allocations where some players are better off without making others worse off.

4.2 Efficiency and Fairness

An allocation A is **Pareto efficient** if there is no allocation A' such that $V_i(A'_i) \geq V_i(A_i)$ for all $i \in N$, and $V_j(A'_j) > V_j(A_j)$ for some $j \in N$.

In other words, we can't make anyone better off without making someone else worse off.

Between round-robin allocation and maximizing utilitarian social welfare (sum of all values), which is Pareto efficient?

- Round-robin: Not necessarily Pareto efficient
  - Players make myopic choices optimizing only their current pick
  - Prevents mutually beneficial trades between players
  - Example: For two players with values $(1, 1)$ and $(1, 0)$ for items a and b, round-robin might assign a to player 1 and b to player 2, but swapping benefits player 2 without harming player 1. (There are also examples where both players strictly benefit.)

- Max utilitarian social welfare: Always Pareto efficient but may be highly unfair
  - By contradiction: Any Pareto improvement would increase the sum of values
  - Since we maximize this sum, no Pareto improvement can exist
  - However, may assign all items to one player with marginally higher valuations

So, can we have the best of both worlds? Yes, using the Maximum Nash Welfare! Read on to learn more about this allocation rule.

4.3 Maximum Nash Welfare

The Nash welfare of an allocation A is the product of values:

$$\text{NW}(A) = \prod_{i \in N} V_i(A_i) \tag{4}$$

The maximum Nash welfare (MNW) solution chooses an allocation that maximizes the Nash welfare.

Theorem: Assuming additive valuations, the MNW solution is both EF1 and Pareto efficient.

Here we see we can have both fairness and efficiency simultaneously. The MNW solution provides a principled way to make fairness-efficiency tradeoffs by maximizing the geometric mean of players' values.

In the case of divisible goods, we can also see that maximizing the product coincides with other notions that are intuitively envy-free.

> **Intuition for Nash welfare:** Nash welfare is particularly sensitive to inequality - if any player gets very little value, the product becomes small. This makes it balance both efficiency (high total welfare) and fairness (relatively equal distribution).
>
> The logarithm of Nash welfare equals the sum of logarithms of individual utilities, which aligns with maximizing utilities on a logarithmic scale - a concept familiar in economics for modeling diminishing marginal utility.
>
> While computing the exact MNW solution is NP-hard, approximation algorithms exist and work reasonably well in practice, as shown by Caragiannis et al. (2016). Their experiments demonstrate that computing MNW is feasible for problems with up to 50 players in reasonable time.

4.4 Practical Implementation

Spliddit (spliddit.org) – which Prof. Procaccia created! – is a website that implements these algorithms to help people divide goods fairly. These platforms allow users to express their values for items and then compute provably fair allocations. The algorithms discussed here have direct real-world applications in inheritance division, resource allocation, and task assignment scenarios. You should use it if you ever have to divide up rooms & rent between roommates!

5 Open Problems

5.1 Envy-Free up to Any Good (EFX)

An allocation $A_1, \ldots, A_n$ is envy-free up to any good (EFX) if and only if:

$$\forall i, j \in N, \forall g \in A_j, V_i(A_i) \geq V_i(A_j \setminus \{g\}) \tag{5}$$

This is strictly stronger than EF1 and strictly weaker than perfect envy-freeness (EF). The key difference from EF1 is that EFX requires no envy after removing *any* good from the envied bundle, not just a carefully chosen one.

Research frontier: EFX is currently one of the most intriguing open problems in fair division. We know:

- An EFX allocation exists for two players with monotonic valuations (relatively easy proof)
- An EFX allocation exists for three players with additive valuations (proved in 2019-2020, with a complex argument, and proved more recently in a slightly simpler way)
- For $n \geq 4$ players with additive valuations, the existence of EFX allocations remains an open problem. Prof. Procaccia thinks this is the biggest open problem in fair division – a problem that is so easy state but so hard to solve.

The fact that we can't even prove or disprove the existence of EFX allocations for just 4 players makes this an exciting research frontier! Some various approximations and partial results have already been discovered towards a solution.