Harvard CS 1360, Spring 2026
Instructor: Ariel Procaccia
Scribe: Rhea Acharya, Jakob de Raaij

Indivisible Goods

—

Lecture 12

In our lecture on cake cutting, we assumed that we can cut the cake into as many pieces as we wish, that we then allocate to the players. However, what if the pieces of cake are already pre-cut and cannot be further divided? A more realistic example is an estate being divided among a group of heirs. Each item in the estate can only be given to one person, and this should be done in a *fair* way. Other real-world scenarios include allocating tasks among team members (a.k.a. indivisible *chores*) and assigning dormitory rooms to students (a.k.a. *matching*).

1 Introduction

Definition 1 (Fair Division of Indivisible Goods)**.** An instance of an *indivisible goods allocation problem* consists of

- a set of *players* $N = \{1, \ldots, n\}$,
- a set G of m *indivisible goods*,
- valuations V_i for each player, where $V_i(S)$ is the valuation of player i for any *bundle* of goods $S \subseteq G$.

We say that the valuations are *additive* if for all $S \subseteq G$ and $i \in N$, it holds that $V_i(S) = \sum_{g \in S} V_i(g)$. In other words, this means that each player has some value for each item, and their value for a bundle of items is the sum over their values for the items.[1]

An allocation is a partition of the goods into n sets, denoted $\boldsymbol{A} = (A_1, \ldots, A_n)$, where A_i is the bundle of goods received by player i.

Unlike divisible settings (like cake cutting), envy-freeness and proportionality may be infeasible when allocating indivisible goods. In particular, consider the case of two players, and one good desired by both players. No matter which player we allocate it to, the other player will be envious and receive disproportionally little utility (namely, 0).

Since the goods are indivisible, we cannot simply divide them to ensure everyone gets a fair share: We often cannot achieve perfect fairness due to the discreteness of the problem. Thus, we need to explore weaker fairness guarantees.

2 Maximin Share Guarantee

Definition 2 (Maximin Share)**.** The *Maximin Share (MMS) guarantee of player i* is the maximum valuation they can guarantee to get if they get to split the goods into n bundles $X_1, ..., X_n$ and then receive their least favorite bundle,

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

A *Maximin Share (MMS) allocation A* is an allocation in which each player receives at least their MMS guarantee,

$$V_i(A_i) \geq \max_{X_1,\ldots,X_n} \min_j V_i(X_j).$$

Intuitively, the MMS of player i represents the maximum value they could guarantee themselves if they were the cutter in a "I cut, you (all) choose" algorithm, as we saw in the cake cutting lecture.

For $n = 2$ and additive valuations, an MMS allocation always exists. To see why, imagine running the "I cut, you choose" algorithm with player 1 cutting. Player 1 divides the goods into two bundles X_1 and X_2 and receives one of the

[1] Additive valuations don't account for substitutes and complements. For example, if the goods to be divided are (identical) apples, my increase in value from getting the first apple might be a lot larger than my increase in value from getting a fifth apple (because I cannot eat five apples). Similarly, my value for either single sock of a pair of socks may be 0 while my value for both of them is non-zero.

two bundles, so at least their MMS guarantee. Player 2 gets to choose their favorite out of those two bundles, so gets at least 1/2 of their valuation for the set of all goods. However, the MMS guarantee of any player is at most 1/2 of their valuation for the set of all goods. Thus, the allocation from the "I cut, you choose" algorithm is a MMS allocation.

Unfortunately, this does not hold for any larger n.

Theorem 1. *For $n \geq 3$, there exist additive valuation functions that do not admit an MMS allocation.*

Proof for $n = 3$. Before we even define any values, we start with a weird combinatorial construction. Assume there are 12 goods with values as shown below

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

We claim (and it is not hard to computationally verify) that there are exactly 3 ways of dividing these numbers into 3 subsets of 4 numbers, such that the numbers in each subset add up to 55. Specifically, the 3 possible divisions are

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

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

where the player-specific perturbation matrices are

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

Player 1

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

Player 2

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

Player 3

Thus, each player's valuation for a good g is

$$V_i(g) = 10^6 + 10^3 \cdot (\text{value in table for } g) + (\text{player-}i\text{-specific perturbation for } g).$$

The player-specific perturbation is much smaller than the other parts of the valuation, so the players have almost identical valuations.

For example, consider the good g that corresponds to the top left corner of the matrix. The valuations are

$$\begin{aligned} V_1(g) &= 10^6 + 10^3 \cdot 17 + (-1) = 1,016,999, \\ V_2(g) &= 10^6 + 10^3 \cdot 17 + 0 = 1,017,000, \\ V_3(g) &= 10^6 + 10^3 \cdot 17 + 0 = 1,017,000. \end{aligned}$$

Note that the player-specific perturbations mirror the construction of the three equal partitions in the combinatorial construction. Each player perturbation matrix corresponds to one of these partitions, with the top left element in the perturbation matrix being 3 and the other three elements in this set of the partition being -1.

The maximin share of each player is $4,055,000$. Each player can achieve this by dividing the goods according to the partition shown in their perturbation matrix. The total value of each bundle is

$$4 \cdot 10^6 + 55 \cdot 10^3 + 0 = 4,055,000,$$

as the numbers in the second-order matrix add up to 55 and the perturbations within each set of this partition add up to 0.

However, it is impossible to guarantee a value of $4,055,000$ to all players simultaneously. Since the magnitudes of the components of the valuation calculation vary by so much, we need the utility from each component to be split equally, as we will not be able to make up for substantial differences in later stages. In particular, each player needs to receive four goods since any player with less goods will have less than $4 \cdot 10^6$ utility. Then, for the second-order matrix, we will have to split the items into three bundles according to one of the three possible partitions of the combinatorially constructed matrix, since otherwise one player would have less than $4 \cdot 10^6 + 55 \cdot 10^3$. Finally, the sum of the perturbation values for the items in the bundle of each player need to be at least zero, so that the player gets a utility of at least $4 \cdot 10^6 + 55 \cdot 10^3$.

However, this is not possible. Let's say we pick the partition by rows, corresponding to player one's perturbation matrix. Player 1 will have valuation $4,055,000$ for all three of those bundles. However, for the yellow and blue subsets (corresponding to the second and third rows), both player 2 and 3 have a perturbation sum of -1. Thus, at least one of player 2 or player 3 will also receive a bundle corresponding to one of these "suboptimal" rows, resulting in a total valuation of $4,054,999$, below the maximin share guarantee. Analogously, we can check that if we pick one of the other two partitions of the combinatorially constructed matrix, there always exists one player who gets utility at most $4,054,999$. Thus, in any allocation, some player cannot get their maximin share. □

The maximin share guarantee only was defined in the 2010s; it took about 4 years until this counterexample to its feasibility was found. In practice, the maximin share guarantee is usually feasible. Later work has shown that we can always find allocations that provide each player at least $3/4$ of their MMS. This is a frequent pattern: the theoretical worst-case results might be much more negative than typical instances behave in practice.

3 Approximate Envy-Freeness for General Monotonic Valuations

We saw that two demanding notions of fairness, envy-freeness and the maximin share guarantee, may not be possible in practice. We'll now show that if we slightly relax envy-freeness, we get a fairness guarantee that is always satisfiable.

Definition 3 (Envy-Free Up To One Good)**.** An allocation $\boldsymbol{A} = (A_1, \ldots, A_n)$ is *envy-free up to one good (EF1)* if for all players $i, j \in N$, there exists a good $g \in A_j$ such that

$$V_i(A_i) \geq V_i(A_j \setminus \{g\});$$

that is, we can remove a single good from player j's bundle so that player i no longer envies player j.

In simple terms: "I might envy what you have, but if you remove your best item, I'd be satisfied with my allocation." In practice, it's often sufficient for maintaining social harmony, if people are willing to overlook small differences. This seems reasonable, especially if the number of items is large.

It turns out that an allocation satisfying envy-freeness up to one good always exists, even for a more generalized class of valuations than additive valuations with positive utilities:

Definition 4 (General Monotonic Valuations)**.** The valuations are *general monotonic* if for every player $i \in N$ and any bundles $S, T \subseteq G$ with $S \subseteq T$, it holds that $V_i(S) \leq V_i(T)$.

Theorem 2. *Given general monotonic valuations, an EF1 allocation always exists and can be computed in polynomial time.*

To prove this theorem, we make two more definitions: First, in a *partial allocation* $\boldsymbol{A} = (A_1, ..., A_n)$ of the goods, only a subset of the goods have been allocated, so $A_1 \cup ... \cup A_n \subseteq G$ but not necessarily $A_1 \cup ... \cup A_n = G$. Second, given a (potentially partial) allocation A, we can construct an *envy graph*: Each player is a vertex, and we draw a directed edge from i to j if and only if i envies j. This graph shows which players are envious of which other players.

Lemma 1. *A partial allocation A satisfying EF1 can be transformed in polynomial time into a partial allocation B of the same goods, still satisfying EF1, 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 2. 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. □

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 EF1 and Efficiency for Additive Valuations

If we restrict our attention to additive valuations, proving the existence of an EF1 allocation becomes much simpler. It turns out that we can use a round-robin procedure:

Algorithm 1 Round-Robin Mechanism for EF1 with Additive Valuations

1: Order the players arbitrarily.
2: In round r, player (r mod n) picks their favorite good that hasn't been picked so far.

This round-robin allocation always ensures an EF1 allocation. Consider any player i and some player $j < i$ that got their first pick before i. We know that when player i made their first pick, player j's second pick was still available, so player i prefers their first pick over player j's second pick. Similarly, player i prefers their kth pick over player j's $(k+1)$th pick. Thus, if we remove the item player j chose on their first pick from their bundle, we know that player i will prefer their bundle over the remaining bundle of player j — they are envy-free with regard to j up to one good. By a similar logic, we can see that player i prefers their bundle over the entire bundle of any player $j > i$ — they are envy free with regard to player j. Thus, every player i is envy-free up to one good.

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, or even children selecting toys. Its theoretical fairness guarantee, EF1, offers an explanation for why this procedure feels fair to participants in practice.

However, it turns out that the allocation from a round-robin procedure can fail another desirable property:

Definition 5 (Pareto Efficiency)**.** An allocation $\boldsymbol{A}$ is *Pareto efficient* if there is no allocation $\boldsymbol{A'}$ 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, there does not exist another allocation that makes some player strictly better off and no player worse off.

In particular, consider a setting with two goods g and h and two players 1 and 2. Player 1 is indifferent between the two goods, assigning a valuation of 10 to both. Player 2 strongly prefers the first good, having a valuation of 100 for g and 0 for h. Since player 1 gets to pick first in the round-robin procedure, they may break their tie arbitrarily and pick item g, leaving player 2 with item h. Now, if those two players swapped items, player 2 would be strictly better off, while player 1 would be no worse off — the round-robin allocation is not Pareto efficient!

A different allocation mechanism that always guarantees Pareto efficiency is to maximize the utilitarian social welfare, i.e., the sum of the valuations, and pick an allocation from $\arg\max_{\boldsymbol{A}} \sum_{i \in N} V_i(A_i)$. However, this may lead to very unfair allocations; in particular, it may assign all items to the same player with marginally higher valuations.

So, can we have the best of both worlds? This answer is yes, using the Nash Welfare.

Definition 6. The *Nash welfare* of an allocation $\boldsymbol{A}$ is the product of the players' valuations for their bundles,

$$\mathrm{NW}(\boldsymbol{A}) = \prod_{i \in N} V_i(A_i).$$

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

Theorem 3. *Given positive*[2]*, additive valuations, the maximum Nash welfare solution is both envy-free up to one item and Pareto efficient.*

We can have both fairness and efficiency simultaneously! The MNW solution provides a principled way to arrive at a tradeoff between fairness and efficiency by maximizing the product (or geometric mean) of the players' values.

In particular, the Nash welfare is very sensitive to players getting very little utility; if any player gets 0 utility, the entire product is 0. If we take the logarithm of Nash welfare (note that optimizing a function or its logarithm is equivalent), it equals the sum of logarithms of individual utilities. Thus, it is implicitly maximizing the utilitarian welfare assuming that voters have diminishing marginal utility, i.e., receive less utility from an increase to their valuation the larger it already is.

Computing the exact MNW solution is computationally hard — in particular infeasible in polynomial time unless **P** = **NP**, we can calculate it reasonably well in practice using integer linear programming with commercial solvers.

[2] Whether envy-freeness up to one item and Pareto efficiency are always simultaneously satisfiable for *negative*, additive valuations is one of the main open problems in the field of fair division.

The experiments by Caragiannis et al. (2016) demonstrate that computing MNW is feasible on the scale of seconds for problems with up to 50 players.

On Spliddit — a website helping with every-day fair division problems created by Prof. Procaccia — this algorithm is implemented. Users can enter their additive valuations for the goods to be allocated and then compute provably fair allocations. On Spliddit, the algorithms we discussed in class have been used by tens of thousands of users to divide inheritances, decide rent splits, allocate resources, and assign tasks.

5 Envy-Free up to Any Good

So far, we have considered envy-freeness up to one good that we can freely pick. It states that no player i should envy another player j if the *most* valuable good from player j's bundle (in the eyes of player i) is removed. It is a natural strengthening to ask if this still holds if the *least* valuable good from player j's bundle is removed.

Definition 7 (Envy-Free up to Any Good (EFX))**.** An allocation $\boldsymbol{A} = (A_1, \ldots, A_n)$ is *envy-free up to any good (EFX)* if for any two players $i, j \in N$, for any good $g \in A_j$, it holds that

$$V_i(A_i) \geq V_i(A_j \setminus \{g\});$$

that is, if we remove *any* single good from player j's bundle, player i no longer envies them.

This is strictly stronger than EF1 and strictly weaker than perfect envy-freeness (EF). It is an open problem whether an EFX allocation always exists. We know that

- an EFX allocation always exists for two players with general monotonic valuations (relatively easy proof),
- an EFX allocation exists for three players with additive valuations (proved initially 2019-2020, with a very complex argument; reproved more recently in a slightly simpler way),

Already 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 to state but so hard to solve. Working towards a solution, various approximations and partial results have already been discovered.