Martin Hoefer, Giovanna Varricchio

Fair Division

Summary: We first discuss the cake-cutting problem, in which we are given a divisible resource, a.k.a. a *cake*, which has to be split among n agents. Each agent has a preference over different pieces, and the goal is to split the cake among the agents in a *fair* manner.
We first concentrate on two criteria for fairness: Proportionality and envy-freeness. We discuss their existence and computational complexity.
We then look into further desirable properties other than fairness; namely, Pareto and Nash optimality.
Subsequently, we also study fair division with indivisible goods. Here multiple items have to be assigned, but each item can be assigned to at most one agent. We consider relaxations of classical fairness notions and explore their computation. We also consider efficiency in conjunction with fairness.

Resources:

- *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A.D. Procaccia, 2016, Cambridge University Press (Chapther 13, Cake Cutting Algorithms).
- *Fair Division of Indivisible Goods: A Survey* G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. A. Voudouris `https://arxiv.org/pdf/2202.07551.pdf`
- *Trends in computational social choice.* Chapter 12 – Approximation Algorithms and Hardness Results for Fair Division with Indivisible Goods.
- *Tutorial on Recent Advances in Fair Resource Allocation*, Rupert Freeman and Nisarg Shah `https://www.cs.toronto.edu/~nisarg/papers/Fair-Division-Tutorial.pdf`
- Further readings in the references.

1 Cake-Cutting

1.1 Setting

We are given a divisible resource, a *cake* C. It is abstractly represented by the interval $[0, 1]$. Moreover, there is a set $\mathcal{N} = \{1, \dots, n\}$ of n agents. A *piece of cake* is any union of disjoint intervals in C. A piece of cake is called *connected* if it is an interval $[x, y] \subseteq C$. The goal is to partition the cake into n pieces and assign each of them to a distinct agent, which means that the cake has to be fully allocated.

Definition 1 (Allocation). *An* allocation $\mathcal{A} = (A_1, \dots, A_n)$ *is a partition of C into pieces, each of which is assigned to a unique agent. For each $i \in \mathcal{N}$, we denote by A_i the piece received by agent i in the allocation $\mathcal{A}$. An allocation must be complete, that is,* $\cup_{i\in\mathcal{N}} A_i = C$.

An allocation $\mathcal{A}$ is called *simple* if each agent receives a connected piece of C, i.e. $\mathcal{A}$ is obtained by cutting C at $n-1$ points. The goal is to compute a *fair* allocation of the cake to the agents. We will later define what fair means. First, we need to introduce how agents evaluate a piece of cake.

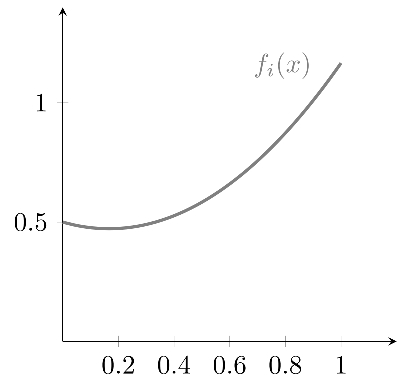
1.1.1 Agent Valuations

We assume each agent can evaluate any subset of the cake. For this reason, each agent $i \in \mathcal{N}$ is endowed of an integrable density function $f_i : C \to \mathbb{R}_{\geq 0}$. Therefore, given any agent $i \in \mathcal{N}$ and any piece of cake $X \subseteq C$, the value of agent i for X is given by

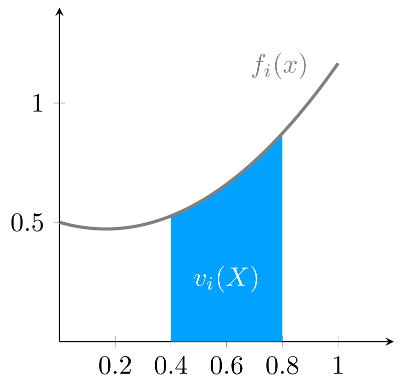
$$v_i(X) = \int_{x \in X} f_i(x) dx \,.$$

We can assume, without loss of generality, that $\int_{x\in[0,1]} f_i(x)dx = 1$, for each agent $i \in \mathcal{N}$.

Example 1. *In Figure 1 we see an example of a density function of an agent i: $f_i(x) = x^2 - \frac{1}{3}x + \frac{1}{2}$ and of the value of the piece $X = [\frac{2}{5}, \frac{4}{5}]$.*



(a) Density function of agent i: $f_i(x) = x^2 - \frac{1}{3}x + \frac{1}{2}$.

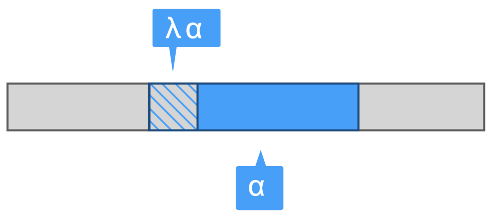


(b) Value of piece $X = [\frac{2}{5}, \frac{4}{5}]$.

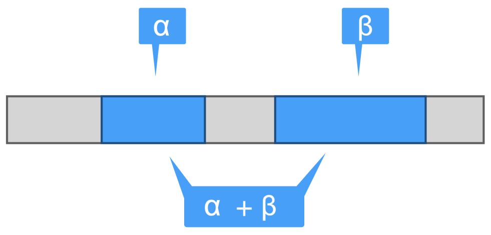
Figure 1: Example of a density function f_i and valuation of a piece X.

Properties. Throughout our treatment we assume that the every valuation function v_i, for each $i \in \mathcal{N}$, satisfies the following properties:

- *normalized*, i.e. $v_i(C) = 1$;
- *divisible*, i.e. $\forall \lambda \in [0, 1]$ and $I = [x, y] \subseteq C$ there exists $z \in I$ such that $v_i([x, z]) = \lambda \cdot v_i([x, y])$;
- *additive*, i.e. for any pair of disjoints intervals $I, I' \subset C$, $v_i(I \cup I') = v_i(I) + v_i(I')$;
- *non-negative*, i.e. for each $I \subseteq C$, $v_i(I) \geq 0$.



(a) Divisible.



(b) Additive.

Figure 2: Properties of valuations.

Whenever we refer to the value of a piece in a given allocation, we mean the *value for the owner*!

1.1.2 Query Complexity

Suppose the density functions of the agents are given as input. They could require infinitely many bits for representation. In contrast, we study a more reasonable way to access the valuations – by *oracle access.* We assume to have a so-called oracle for each agent i that can answer some meaningful questions about v_i.

Robertson-Webb model. For each agent i, we assume there exists an oracle that can answer the following two queries:

- $Eval_i(x, y)$ returns $v_i([x, y])$;
- $Cut_i(x, \alpha)$ returns y such that $v_i([x, y]) = \alpha$.

As a consequence, we evaluate the performance of an algorithm by its *query complexity*, that is, the number of queries required during its execution.

1.2 Fairness Criteria

In this section, we introduce some fundamental fairness criteria and discuss their relations.

1.2.1 Definitions

We distinguish between threshold-based and comparison-based criteria. A threshold-based criterion requires that each agent receives at least some pre-defined value for her piece of cake; a comparison-based criteria determines the satisfaction of an agent by comparing the piece she receives with the pieces received by the others.
A very natural threshold value is the proportional share: The value of the entire cake divided by the number of agents. Formally, the *proportional share of agent i* is given by $\mathsf{PS}_i = \frac{v_i(C)}{n}$. We assumed valuations to be normalized, which implies that each agent has a proportional share of $\frac{1}{n}$.

Definition 2 (Proportionality)**.** *An allocation $\mathcal{A}$ is called* proportional (PROP) *if each agent receives at least her proportional share, that is, $\forall i \in \mathcal{N}$ it holds*

$$v_i(A_i) \geq \frac{1}{n} \ .$$

Towards comparison-based criteria, a very intuitive one is *envy-freeness.* It requires that no agent envies any other agent.

Definition 3 (Envy-freeness)**.** *An allocation $\mathcal{A}$ is called* envy-free (EF) *if for each $i, j \in \mathcal{N}$ it holds*

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

Example 2. *Let us consider a cake-cutting instance with three agents having valuations as depicted in Figure 3. The allocation $A_1 = [0, 1/6]$, $A_2 = [1/6, 5/6]$, and $A_3 = [5/6, 1]$ is* PROP *and* EF.

1.2.2 Implications

Proposition 1. *Any* EF *allocation is always* PROP.

Proof. Let us show the statement for an arbitrary agent $i \in \mathcal{N}$. Since $\mathcal{A}$ is EF, for each $j \in \mathcal{N}$ it holds

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

Summing up over all $j \in \mathcal{N}$ we get

$$n \cdot v_i(A_i) \geq \sum_{j \in \mathcal{N}} v_i(A_j) \overset{(1)}{=} v_i(A_1 \cup \ldots \cup A_n) \overset{(2)}{=} v_i(C) \overset{(3)}{=} 1, \quad \text{and, thus,} \quad v_i(A_i) \geq \frac{1}{n} \ .$$

Note that (1) holds because of the additivity of valuations and the fact that the pieces in A are non-overlapping, (2) because the cake is completely allocated, and (3) because of normalization. □

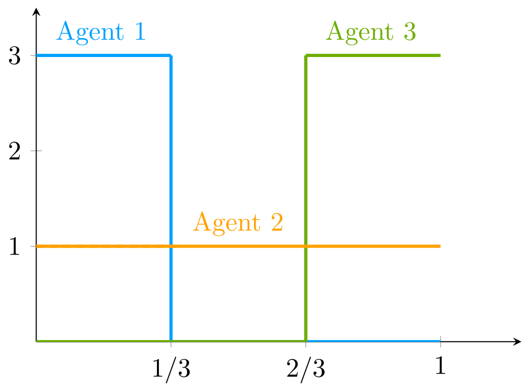


Figure 3: Example instance with three agents.

Proposition 2. *For $n = 2$, any* PROP *allocation is always* EF*.*

Proof. Roughly speaking, if an agent receives a piece she values more than half, from her perspective, the other agent is receiving less than half.
Formally, given $i \in \{1, 2\}$, by proportionality $v_i(A_i) \geq \frac{1}{2}$ and therefore $v_i(A_j) = v_i(C \setminus A_i) \leq \frac{1}{2} \leq v_i(A_i)$. □

1.3 Existence and Computation

Let us discuss the existence and computation of PROP and EF allocations. We start by considering the simplest scenario with two agents.

1.3.1 Two Agents: The Cut-and-Choose Protocol

For two agents, where PROP ⇔ EF, there exists a simple but nonetheless interesting protocol for achieving both PROP and EF, the so-called CutAndChoose. It works as follows:

Cut: Agent 1 cuts the cake into two pieces of value $\frac{1}{2}$ for her.
Choose: Agent 2 selects the piece she prefers the most and agent 1 receives the other.

Theorem 3. *The* CutAndChoose *protocol outputs a proportional (and hence envy-free) allocation.*

Proof. Agent 1 receives a piece of value $\frac{1}{2}$ for him. Agent 2 select the most preferred piece whose has necessarily value $\geq \frac{1}{2}$. □

And the query complexity? Only two queries! We only ask $y \leftarrow Cut_1(0, \frac{1}{2})$ and $Eval_2(0, y)$.

1.3.2 The Dubins-Spainer Protocol – Proportionality with $O(n^2)$ Queries

We now turn our attention to the computation of a proportional allocation for n agents. Proportionality is the easiest criterion to achieve, and several protocols have been proposed. We introduce the Dubins-Spainer protocol also known as the MovingKnife.
Intuitively, starting from the left-most position 0, we pretend to move a knife along our cake to the right. Consider the first a point at which the unassigned piece to the left has value of the proportional share to some remaining agent i. At that point, we cut the cake and assign the piece to i. Then, i is disregarded as well as the assigned piece of cake. We continue from the current position, move the knife to the right, and find again the first position where the next agent gets her proportional share.
Formally, let $x_k \in [0, 1]$ be the position of the knife. The MovingKnife protocol proceeds as follows:

- During the ℓ-th iteration of the algorithm, the knife is positioned at $x_k = y_{\ell-1}$, where $y_0 = 0$. Then, x_k is slowly (and contiguously) moved to the right.
- The agents are allowed to shout as soon as the piece $[y_{\ell-1}, x_k]$ reaches their proportional share. Hence, whenever there is an agent i such that $v_i([y_{\ell-1}, x_k]) \geq \frac{1}{n}$ agent i shouts; if there exists more than one such agent we break ties arbitrarily.
- As soon as one agent shouts, that agent receives the piece of cake $[y_{\ell-1}, x_k]$ and leaves the protocol, i.e. $\mathcal{N} \leftarrow \mathcal{N} \setminus \{i_\ell\}$. Set $y_\ell = x_k$. The process repeats on the remaining cake with the remaining agents. Note, however, that each agent continues to strive for a proportional share of $1/n$, where n remains the original number of agents!
- Finally, when $|\mathcal{N}|= 1$, assign $[y_\ell, 1]$ to the unique remaining agent and terminate.

We described the MovingKnife protocol as a contiguous process over the cake. How to implement such a process in the Robertson-Webb model? At round ℓ, we use $Cut_i(y_{\ell-1}, \frac{1}{n})$ for all $i \in \mathcal{N}$. Roughly speaking, we ask all the agents at which point they would shout. By taking the minimum of all these positions, we know the first shouter and the place to cut.

Example 3. *We consider an example run of* MovingKnife *protocol on the instance depicted in Figure 3. At the beginning, no agent owns any piece of cake. The protocol starts from the position $x_k = 0$ and asks every agent where to cut to obtain her proportional share, i.e. $Cut_i(0, 1/3)$ for each $i \in \{1, 2, 3\}$. Since $Cut_i(0, 1/3) = 1/9$, $Cut_2(0, 1/3) = 1/3$, and $Cut_3(0, 1/3) = 7/9$, agent 1 decleares the smallest value, gets selected, and receives the piece $[0, 1/9]$. Agents 2 and 3 remain in the process, and the knife is positioned in $x_k = 1/9$. Since $Cut_2(1/9, 1/3) = 4/9$ and $Cut_3(1/9, 1/3) = 7/9$, agent 2 receives $[1/9, 4/9]$ and agent 3 receives $[4/9, 1]$.*

Theorem 4. *The* MovingKnife *protocol outputs a simple and proportional allocation.*

Proof. Simple: The algorithm assigns every agent a contiguous piece of cake.
Proportional: The algorithm assigns every agent (but the last one) a piece with value of the proportional share. We need to show that (1) it never consumes the entire cake before every agent receives one piece, and (2) the remaining of the cake is always enough to guarantee the proportional share to the remaining agents. We prove the following claim: At the beginning of any round $\ell \in [n-1]$, let $C' \subseteq C$ be the remaining of the cake; if agent i has not received a piece of cake yet, then $v_i(C') \geq 1/n$.
Let $C_h = [y_{h-1}, y_h]$ be the piece of cake assigned at the h-th round. Since i has not received the cake, no such piece has value more than $1/n$. Therefore, $v_i(C') = v_i(C) - \sum_{h<\ell} v_i(C_h) \geq 1 - (\ell - 1)/n \geq 1/n$.
Since the above argument applies also to the agent receiving the last piece, proportionality holds also for her. □

And query complexity? At each step, we ask the remaining agents where we should cut for her proportional share. Hence, we need $\sum_{i=1}^{n-1} n - i + 1 = \Theta(n^2)$ queries.
Can we do better?

1.3.3 The Even-Paz Protocol – Proportionality with $O(n \log n)$ Queries

We present a recursive protocol. For simplicity, we assume $n = 2^k$ for some integer k.

Input: An interval $[x, y]$ and n agents
- If $n = 1$, then give $[x, y]$ to the agent and terminate;
- else each agent i computes z_i such that $v_i([x, z_i]) = \frac{1}{2} v_i([x, y])$.
- Select z^* the $n/2$-th value z_i from the left in $[x, y]$;
- Recruse on $[x, z^*]$ with the left $n/2$ agents, and on $[z^*, y]$ with the right $n/2$ agents.

Theorem 5. *The Even-Paz protocol returns a proportional allocation.*

Proof idea. Invariant for each recursive call: There is enough cake for the involved players to get at least their proportional share. By induction, at the last step, only one agent is considered and therefore she receives her proportional share. □

Query complexity: The protocol runs in $O(\log n)$ rounds; in every round each agent replies to exactly one query $\rightarrow O(n \log n)$.

Theorem 6 (Edmonds and Pruhs, 2006)**.** *Any proportional protocol needs* $\Omega(n \log n)$ *queries in the Robertson-Webb model.*

Therefore the Even-Paz protocol is asymptotically optimal!

1.3.4 Envy-Freeness for Three Agents: The Selfridge-Conway Protocol

We have seen that the CutAndChoose protocol provides an EF allocation for two agents. Around 1960, Selfridge and Conway (independently) constructed the same algorithm for three agents.

Initialization:

- Agent 1 divides the cake into three equally-valued pieces $X_1, X_2, X_3 : v_1(X_1) = v_1(X_2) = v_1(X_3) = 1/3$.
- Agent 2 trims the most valuable piece according to v_2 to create a tie for the most valuable. For example, we assume w.l.o.g. that X_1 is the most valuable piece for agent 2.
  - If $v_2(X_1) > v_2(X_2) \geq v_2(X_3)$, agent 2 removes $X' \subseteq X_1$ such that $v_2(X_1 \setminus X') = v_2(X_2)$. We call the three pieces – one of which is trimmed – *cake 1* ($\{X_1 \setminus X'\} \cup X_2 \cup X_3$), and we call the trimmings (X') *cake 2*.
  - If $v_2(X_1) = v_2(X_2)$, *cake 2* is empty.

Division of cake 1:

- Agent 3 chooses one of the three pieces of cake 1;
- If agent 3 chose the trimmed piece ($X_1 \setminus X'$), agent 2 chooses between the two other pieces of cake 1. Otherwise, agent 2 receives the trimmed piece. We call refer to the agent $i \in \{2, 3\}$ that received the trimmed piece by agent T. The other agent is agent $\overline{T}$;
- Agent 1 receives the remaining piece of cake 1.

Division of cake 2:

- Agent $\overline{T}$ divides cake 2 into three equally-valued pieces;
- Agents $T, 1, \overline{T}$ select a piece of cake 2 each, in that order.

For an example, apply the procedure on the instance depicted in Figure 3.

Theorem 7. *The Selfridge-Conway protocol outputs an* EF *allocation for three agents.*

Proof. Let us denote cakes 1 and 2 by C_1 and C_2, respectively.
Observation: The division of C_1 is EF. Indeed, agent 1 always receives a piece of value 1/3, and no other piece has a higher value; agent 2 always receives one of the most two preferred (and equally liked) pieces; agent 3 selects the most preferred piece. Therefore, if $C_2 = \emptyset$, the theorem follows.
Otherwise, let us consider the final allocation of C (that is after the division of C_1 and $C_2 \neq \emptyset$): Agent $\overline{T}$, who is splitting the cake, is the one who receives the remaining piece of C_2; anyway, she will be EF in the

final allocation, because she likes the three pieces of C_2 equally. The agent T selecting first piece is also EF in the final allocation. It remains to show agent 1 is EF.
Agent 1 does not envy $\overline{T}$, since 1 selects the piece of C_2 before $\overline{T}$. But does agent 1 envy agent T in the final division of the whole cake C?
On the one hand, T is the agent who received $X_1 \setminus X'$, and $\{X_1 \setminus X'\} \cup C_2 = X_1$. Therefore, no matter which piece of C_2 agent T receives, agent 1 will value the final piece of T at most 1/3 (because of the initialization of the algorithm X_1 has a value of 1/3 for agent 1). On the other hand, during the allocation of C_1, agent 1 received a piece of value 1/3 and by adding another piece from C_2 cannot decrease the value attained by 1, showing that 1 does not envy T. □

1.3.5 Envy-Freeness for any Number of Agents

What about the existence of EF allocations for general n? We will see in the next section that they always exist. What about computation?

Theorem 8 (Aziz and Mackenzie, 2016)**.** *There exists a finite protocol for computing an* EF *allocation with a query complexity of* $O\left(n^{n^{n^{n^{n^{n}}}}}\right)$.

Theorem 9 (Procaccia 2009)**.** *Any protocol for finding an envy-free allocation requires $\Omega(n^2)$ queries in the Robertson-Webb model.*

There still is a large gap between these lower and upper bounds.

1.4 Efficiency

To obtain fairness, we might produce extremely inefficient partitions of a cake, as the following example shows.

Example 4. *Assume there are two agents $\mathcal{N} = \{1, 2\}$, $f_1 = U[0, 1/2]$ and $f_2 = U[1/2, 1]$, where $U[x, y]$ is the uniform distribution over $[x, y]$. Consider the partition where agent 1 receives $[0, 1/4] \cup [3/4, 1]$ and agent 2 receives $[1/4, 3/4]$. Such an allocation is* EF *and therefore* PROP*; notice that both agents receive their proportional share. However, there is a much better allocation which is still* EF *but both agents get a utility of 1; namely, it is sufficient to cut the cake at $x = 1/2$ and assign the left side o agent 1, and the right side to agent 2.*

This example might look artificial; it is not hard to verify that all the protocols we provided so far are inefficient in the sense we are going to define in the next subsection.

1.4.1 Pareto Optimality

The most prominent definition of efficiency is *Pareto optimality.* Roughly speaking, an allocation is called Pareto optimal (or Pareto efficient) if there exists no other allocation where each agent is not decreasing and at least one agent is strictly increasing her utility. Formally:

Definition 4 (Pareto optimal allocation)**.** *Given a pair of allocations $\mathcal{A}, \mathcal{B}$ of the cake C, $\mathcal{B}$* Pareto dominates *$\mathcal{A}$, if for each $i \in \mathcal{N}$, $v_i(B_i) \geq v_i(A_i)$ and at least one of the inequalities is strict.*
An allocation $\mathcal{A}$ of C is Pareto optimal *(PO) if there is no allocation $\mathcal{B}$ that Pareto dominates $\mathcal{A}$.*

Notice that a Pareto optimal allocation always exists. Pareto optimality is not per se an interesting property, we will strive for fair allocations which also satisfy Pareto optimality.

1.4.2 Nash Social Welfare – Fair and Efficient

A way to obtain Pareto optimal allocations is to resort to an optimization problem related to a particular welfare function. It turns out that by maximizing the Nash social welfare function, we obtain an extremely fair allocation and also guarantee Pareto optimality.

Definition 5. *Given an allocation $\mathcal{A}$ of a cake C, the* Nash social welfare *(NSW) of $\mathcal{A}$ is defined as follows:*

$$NSW(\mathcal{A}) = \left(\prod_{i \in \mathcal{N}} v_i(A_i) \right)^{\frac{1}{n}} .$$

Clearly, an allocation maximizing the Nash social welfare always exists. We call such an allocation *NSW-optimal.*

Example 5. *Consider the instance depicted in Figure 3 and the allocation $A_1 = [0, 1/6]$, $A_2 = [1/6, 5/6]$, and $A_3 = [5/6, 1]$ has a Nash welfare of $\left(\frac{1}{2} \cdot \frac{4}{6} \cdot \frac{1}{2}\right)^{\frac{1}{3}} = \left(\frac{1}{6}\right)^{\frac{1}{3}}$. This allocation is* EF *but is not Nash optimal. Consider the allocation where the first third of the cake is assigned to agent 1, the second third to agent 2, and the remaining piece to agent 3. Such an allocation has a Nash welfare of $\left(1 \cdot \frac{1}{3} \cdot 1\right)^{\frac{1}{3}} = \left(\frac{1}{3}\right)^{\frac{1}{3}}$*

Proposition 10. *Every NSW-optimal $\mathcal{A}$ is PO.*

Proof. By contradiction, if an allocation $\mathcal{B}$ Pareto dominates $\mathcal{A}$ then $\mathcal{B}$ has strictly better NSW. $\square$

Notice that maximum Nash social welfare is scale-invariant: if we multiply the valuation v_i of any agent i by any positive number $\alpha_i > 0$, then this does not change the optimal allocations w.r.t. NSW.
An interesting aspect of the NSW is that it is a good (and fair) trade-off between egalitarian and utilitarian social welfare objectives.

- Utilitarian social welfare: Sum of agents' utilities, i.e. $USW(\mathcal{A}) = \sum_i v_i(A_i)$. The USW only focuses on overall happiness without caring about each individual.
- Egalitarian social welfare: Minimum of agents' utilities, i.e. $ESW(\mathcal{A}) = \min_i v_i(A_i)$. The ESW cares about a specific individual without caring about the overall happiness.

1.5 Existence of Envy-Free Allocations

In this section, we discuss the existence of EF allocations. The main result is the following:

Theorem 11. *Every NSW-optimal $\mathcal{A}$ is* EF.

Let's first build an intuition for why this is true.

Example 6. *Consider an instance with two agents. Agent 1 has density function $U[0, 1/2]$ while agent 2 density function $U[0, 1]$. Suppose we have the allocation $\mathcal{A}$ with $A_1 = [0, 1/6]$ and $A_2 = [1/6, 1]$. Now $\mathcal{A}$ is not* EF *and has a Nash welfare of $\sqrt{\frac{1}{3} \cdot \frac{5}{6}} = \sqrt{\frac{5}{18}}$.*
To prove our theorem, we show that whenever an allocation is not EF *it is possible to move a piece of cake from the bundle of the envied to the bundle of the envious agent while increasing the Nash welfare.*
In the specific example, 1 envies 2 and we can move $[1/6, 1/4]$ from A_2 to A_1. After this move, we obtain a new allocation $\mathcal{A}'$, and the new Nash welfare is $\sqrt{\frac{1}{2} \cdot \frac{3}{4}} = \sqrt{\frac{3}{8}}$. Thus, we increased the Nash welfare.

Proof. Let $\mathcal{A}$ be a NSW-optimal allocation. It is easy to see that in such an optimal allocation we have $NSW(\mathcal{A}) > 0$ (since this is already true, e.g., for every allocation that is PROP). Thus, $v_i(A_i) > 0$ for all agents $i \in \mathcal{N}$.

Let us assume by contradiction that $\mathcal{A}$ is not EF. Then there exist $i, j \in \mathcal{N}$ such that $v_i(A_j) > v_i(A_i)$. To reach a contradiction, we show there exists a piece of cake $Z \subset A_j$ such that we can strictly improve the NSW by moving Z from A_j to A_i.
Let j split A_j into k equally liked pieces for her, for some $k \in \mathbb{N}$, i.e. each of them has value $\frac{1}{k} \cdot v_j(A_j)$. Let i choose the most preferable piece, let's call this piece Z. Then the following conditions hold:

$$v_j(Z) = \frac{1}{k} \cdot v_j(A_j) \qquad \text{and} \qquad v_i(Z) \geq \frac{1}{k} \cdot v_i(A_j).$$

Let us call $\mathcal{A}'$ the allocation obtained by moving Z from A_j to A_i.
Notice that only i and j have a different utility in $\mathcal{A}$ and $\mathcal{A}'$, respectively. Therefore, to understand which allocation is better it suffices to compare the product of the utilities of i and j in the two allocations. Formally:

$$\frac{\mathrm{NSW}(\mathcal{A}')}{\mathrm{NSW}(\mathcal{A})} > 1 \quad \Leftrightarrow \quad \left(\frac{\prod_{k\in\mathcal{N}} v_k(A'_k)}{\prod_{k\in\mathcal{N}} v_k(A_k)}\right)^{\frac{1}{n}} > 1 \quad \Leftrightarrow \quad \left(\frac{v_i(A'_i)\cdot v_j(A'_j)}{v_i(A_i)\cdot v_j(A_j)}\right)^{\frac{1}{n}} > 1 \quad \Leftrightarrow \quad \frac{v_i(A'_i)\cdot v_j(A'_j)}{v_i(A_i)\cdot v_j(A_j)} > 1$$

If $v_i(A'_i) \cdot v_j(A'_j) > v_i(A_i) \cdot v_j(A_j)$, we reach the desired contradiction.
We have

$$\begin{aligned} v_i(A'_i) \cdot v_j(A'_j) &= (v_i(A_i) + v_i(Z)) \cdot (v_j(A_j) - v_j(Z)) \\ &\geq \left(v_i(A_i) + \frac{1}{k} \cdot v_i(A_j)\right)\left(1 - \frac{1}{k}\right) \cdot v_j(A_j) \\ &= v_i(A_i) \cdot v_j(A_j) - \frac{1}{k} \cdot v_i(A_i) \cdot v_j(A_j) + \frac{1}{k}\left(1 - \frac{1}{k}\right) \cdot v_j(A_j) \cdot v_i(A_j)\,. \end{aligned}$$

Therefore, we reach the desired contradiction if

$$\begin{aligned} -\frac{1}{k} \cdot v_i(A_i) \cdot v_j(A_j) + \frac{1}{k}\left(1 - \frac{1}{k}\right) \cdot v_j(A_j) \cdot v_i(A_j) &> 0 \\ \Longleftrightarrow \qquad -v_i(A_i) + \left(1 - \frac{1}{k}\right) \cdot v_i(A_j) &> 0 \qquad \qquad (\text{since } v_j(A_j) > 0) \\ \Longleftrightarrow \qquad \frac{v_i(A_j)}{v_i(A_j) - v_i(A_i)} &< k. \end{aligned}$$

To conclude, if i envies j, then by choosing k large enough, it is possible to find a piece of cake Z such that by moving Z from A_j to A_i, the NSW strictly improves. This is a contradiction to the optimality of $\mathcal{A}$. The theorem follows. $\square$

Since any NSW-optimal allocation is PO we also have the following:

Proposition 12. *There always exists an allocation that is simultaneously* EF *and PO.*

What about computation? It is not necessarily easy and depends on the type of valuations we are considering.

2 Indivisble Goods

2.1 Setting

We are given a set of m indivisible resources, a.k.a. *items* or *goods*, $\mathcal{G} = \{g_1, \dots, g_m\}$, and a set $\mathcal{N} = \{1, \dots, n\}$ of n agents.

Definition 6 (Allocation)**.** *An* allocation $\mathcal{A}$ *is a partition of* $\mathcal{G}$ *into disjoint sets, each of them assigned to at most one agent. For each* $i \in \mathcal{N}$*, we denote by* $A_i \subseteq \mathcal{G}$ *the* bundle *(that is, the set of items) received by agent* i *in the allocation* $\mathcal{A}$*.*

Usually, the allocation is also required to be complete, that is, $\bigcup_i A_i = \mathcal{G}$.

Agents' valuations. Agents have preferences over possible bundles they might receive. Preferences are usually quantifiable and are expressed by means of valuations functions.

Definition 7 (Valuations)**.** *The* valuation function *of agent i is a mapping $v_i : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$.*

We will mostly focus on valuation functions that are additive. The results we provide hold only for additive valuations unless specified otherwise.

Definition 8 (Additive Valuations)**.** *A valuation function $v : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$ is called* additive *if for each $X \subseteq \mathcal{G}$, $v(X) = \sum_{g \in X} v(\{g\})$.*

For our convenience, in what follows we will write $v(g)$ instead of $v(\{g\})$.

Example 7. *Consider the example of an instance with additive valuations depicted in Table 1. We have three agents and five items. The depicted allocation gives a value of* 17 *to agent 1, 12 to agent 2, and 3 to agent 3.*

	g_1	g_2	g_3	g_4	g_5
Agent 1	$\textcircled{15}$	3	$\textcircled{2}$	2	6
Agent 2	7	5	5	$\textcircled{5}$	$\textcircled{7}$
Agent 3	20	$\textcircled{3}$	3	3	3

Table 1: An example of additive valuations. Line i corresponds to agent i. Circles correspond to the allocated items in a possible (complete) allocation.

2.2 Fairness Criteria

In this section, we reintroduce some fairness criteria from cake cutting. We will see that the solutions we defined are no longer guaranteed to exist. Therefore, we introduce some relaxations to circumvent this problem.

2.2.1 Definitions

The *proportional share* of agent i is given by $\mathsf{PS}_i = \frac{v_i(\mathcal{G})}{n}$.

Definition 9 (Proportionality)**.** *An allocation $\mathcal{A}$ is called* proportional (PROP) *if each agent receives at least her proportional share, that is, $\forall i \in \mathcal{N}$ it holds*

$$v_i(A_i) \geq \mathsf{PS}_i \ .$$

Also redefining envy-freeness is straightforward.

Definition 10 (Envy-freeness)**.** *An allocation $\mathcal{A}$ is called* envy-free (EF) *if for each $i, j \in \mathcal{N}$ it holds*

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

Observe that for indivisible items $\mathsf{EF} \implies \mathsf{PROP}$, and there might exist allocations which are PROP but not EF.
Unfortunately, EF and PROP allocations may not exist.

Example 8. *Let us consider two agents and one valuable (positively valued by both agents) item. No matter who receives the item the resulting allocation is neither* EF *nor* PROP.

Such an impossibility holds even when $m > n$, see our example in Table 1. Indeed, this instance does not admit any envy-free or proportional allocation. Consider agent 3. She must get at least $\{g_1\}$ or $\{g_2, g_3, g_4, g_5\}$. The latter case, leaves one item to be allocated to either agent 1 or 2 which cannot lead to a proportional allocation. On the other hand, if agent 3 gets g_1, agent 1 must receive at least three of the remaining four goods and 2 must get at least two, which is not possible.

Due to this impossibility result, relaxed versions of EF and PROP have been introduced and studied.

Definition 11 (Proportionality up to one Good)**.** *An allocation $\mathcal{A}$ is called* proportional up to one good (PROP1) *if, for each $i \in \mathcal{N}$, either $A_i = \mathcal{G}$ or there exists $g \in \mathcal{G} \setminus A_i$*

$$v_i(A_i \cup \{g\}) \geq \mathsf{PS}_i \ .$$

Notice that $v_i(A_i \cup g) = v_i(A_i) + v_i(g)$ when valuations are additive.

Definition 12 (Envy-Freeness up to one Good)**.** *An allocation $\mathcal{A}$ is called* envy-free up to one good (EF1) *if, for each $i, j \in \mathcal{N}$, either $A_j = \emptyset$ or there exists $g \in A_j$ such that*

$$v_i(A_i) \geq v_i(A_j \setminus \{g\}) \ .$$

Example 9. *The allocation depicted in Table 1 is an* EF1 *allocation.*

Remark Clearly, EF $\implies$ EF1 and PROP $\implies$ PROP1.

Proposition 13. *Any* EF1 *allocation is also* PROP1 *(in instances with additive valuations).*

Proof. Let us show the statement for agent i. Since i is EF1, for each $j \in \mathcal{N}$, with $A_j \neq \emptyset$, there exists $g_j \in A_j$ such that

$$v_i(A_i) \geq v_i(A_j \setminus \{g_j\}) \ .$$

Summing up for all $i \in \mathcal{N}$, and by additivity, we get

$$n \cdot v_i(A_i) \geq v_i(A_i) + \sum_{j \neq i} (v_i(A_j) - v_i(g_j)) = v_i(\mathcal{G}) - \sum_{j \neq i} v_i(g_j) \ ,$$

and therefore $v_i(A_i) \geq \mathsf{PS}_i - \frac{\sum_{j \neq i} v_i(g_j)}{n}$. If every $A_j = \emptyset$, then $A_i = \mathcal{G}$ making the allocation proportional for i. Otherwise, by selecting $g^* = \arg\max_{g \in \mathcal{G} \setminus A_i} v_i(g)$, we have $v_i(A_i) \geq \mathsf{PS}_i - \frac{(n-1)v_i(g^*)}{n} \geq \mathsf{PS}_i - v_i(g^*)$, and hence agent i is PROP1.
Applying the same argument to all the agents the statement follows. □

The statement does not necessarily hold beyond additive valuations (e.g., it might fail to hold for general monotone valuations defined below). Also, we can observe that the converse statement is false, even for additive valuations.

Proposition 14. *There exist allocations (for instances with additive valuations) that are* PROP1 *but not* EF1*.*

Proof. Exercise. □

2.3 Existence and Computation of EF1 and PROP1 Allocations

Let us discuss the computation of EF1 allocations for additive (where they are also PROP1) and monotone valuation functions.

2.3.1 Round-Robin Procedure

We show that, thanks to a Round-Robin procedure, EF1 allocations always exist for additive valuations and can be computed in polynomial time.
Let us first consider general sequential algorithms. Roughly speaking, in a sequential allocation of items, we create a vector (sequence) $s = (s_1, \dots, s_m)$ where the component s_h corresponds to the agent who will select her most preferred item at the h-th round of the procedure. The vector s is also known as *picking sequence.*

Sequential allocation algorithm. A sequential allocation algorithm takes as input a picking sequence s, the goods $\mathcal{G}$, the agents $\mathcal{N}$, and their valuations. The algorithm proceeds as follows:

```
• $\mathcal{A} \leftarrow (\emptyset, \dots, \emptyset)$
• For $h = 1, \dots, m$
    – $i \leftarrow s_h$
    – $g^* \leftarrow \arg\max_{g \in \mathcal{G}} v_i(g)$
    – $A_i \leftarrow A_i \cup \{g^*\}$, $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g^*\}$
```

The *Round-Robin procedure* is the sequential allocation algorithm executed with a picking sequence of length m in the form $s = (1, \dots, n, 1, \dots, n, 1, \dots)$, for a fixed ordering $1, \dots, n$ of the agents.

Example 10. *The allocation in Tab. 1 is the result of Round-Robin for the ordering* $1, 2, 3$.

Theorem 15. *The Round-Robin procedure outputs an* EF1 *allocation when agents' valuations are additive.*

Proof. Let us split the algorithm into rounds: We call Round k the k-th occurrence of $1, \dots, n$ in the picking sequence $s = (1, \dots, n, 1, \dots, n, 1, \dots)$. Therefore, in a round k, the agents receive a k-th item, if possible. Notice that in the last round it is possible that not all the agents have the opportunity to select an item.
We start by noticing that the first agent in the ordering (which is 1) is EF since the item she gets in a Round k is at least as good as the item selected by any other agent in the same round.
Let us consider agent i and remove the first $i-1$ agents in the sequence s (let's call this new sequence $s(i)$) and remove the items these agents selected in the first round. By running Round-Robin with $s(i)$ on the refined set of goods i is the first agent in the sequence. Notice that $s(i)$ is a Round-Robin sequence for the ordering $i, i+1, \dots n, 1, \dots i-1$, and hence we get an EF allocation for i. By reassigning the items we removed to their owner we get an EF1 allocation. Moreover, this allocation coincides with the outcome of the original Round-Robin with the sequence s, and therefore the statement follows. □

2.3.2 Envy-Graph and Envy-Cycle Elimination

Is it possible to achieve EF1 (but not necessarily PROP1) for more general valuation functions?

Definition 13 (Monotone Valuations)**.** *A valuation function $v : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$ is called* monotone *if for each $Y \subseteq X \subseteq \mathcal{G}$, $v(Y) \leq v(X)$.*

For our purposes we need to introduce the following instrument:

Definition 14 (Envy-Graph)**.** *Given a (partial) allocation $\mathcal{A}$, the* envy graph *for the allocation $\mathcal{A}$ is defined as follows:*

- *each agent i is represented by a node, for simplicity we call the node i;*
- *there exists a directed edge (i, j) if and only if $v_i(A_j) > v_i(A_i)$.*

Note that the directed edge represents the envy of i towards j. Hence, if i is a source in the envy-graph, she is not envied by any other agent.
The envy-graph is an extremely useful tool to reduce envy in an allocation. In fact, if there exists a cycle we can reduce the envy by trading bundles along the cycle. Formally, let us assume that an allocation $\mathcal{A}$ induces a cycle $C = i_1, i_2, \dots, i_k, i_1$ involving k (w.l.o.g.) distinct agents. Trading the bundles along the cycle means we create a new allocation $\mathcal{A}'$ where $A'_i = A_{i+1}$ for each $i = 1, \dots k-1$ and $A'_k = A_1$ while all the other bundles remains the same. By trading bundles along a cycle we reduce the number of edges in the envy-graph without creating new ones. More importantly, trading along an envy-cycle preserves the $\mathsf{EF1}$ property, as formalized in the following lemma:

Lemma 16. *Given an* $\mathsf{EF1}$ *allocation* $\mathcal{A}$*, if the envy-graph has a cycle* C*, then the allocation* $\mathcal{A}'$ *obtained from* $\mathcal{A}$ *by trading the cycle* C *is also* $\mathsf{EF1}$*.*

Proof. From the perspective of agents who are not in the cycle, the allocation is not changing significantly (we are only changing the owners of the bundles).
For the agents in the cycle, the value of their bundle increases. It is higher than the value attributed to the bundle they previously had. Therefore, in the new allocation, any agent i in the cycle does not envy her previous bundle while the previous edges (i, j) can only disappear. □

The envy-cycle elimination protocol. The envy-cycle elimination starts from an empty allocation (which is clearly $\mathsf{EF1}$). At each round, one available item g is allocated to some agent i who is not envied by any other (which is a source node in the envy-graph). This maintains the $\mathsf{EF1}$ property since i was note envied before inserting g. At the end of the round, to guarantee the existence of source nodes, if a cycle appears in the envy-graph, bundles along the cycle are traded. Therefore, at the end of each round there is no cycle in the envy-graph. Since $\mathsf{EF1}$ is preserved (by Lemma 16) at each step of the procedure, the final allocation is $\mathsf{EF1}$.
Formally, envy-cycle elimination can be summarized as follows:

- $\mathcal{A} \leftarrow (\emptyset, \dots, \emptyset)$
- Consider goods sequentially from g_1 up to g_m
- For $h = 1, \dots, m$
  - $i \leftarrow$ a sink node in the envy-graph
  - $A_i \leftarrow A_i \cup \{g_h\}$, $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g_h\}$
  - update the envy-graph
  - While there exists cycle C in the envy-graph: Trade along cycle C

Theorem 17 (Lipton et al. 2004)**.** *If agents' valuations are monotone, then envy-cycle elimination outputs an* $\mathsf{EF1}$ *allocation.*

Proof. The proof proceeds by induction as explained above. At each step the $\mathsf{EF1}$ property is satisfied. □

What is the complexity of envy-cycle elimination? This procedure guarantees the existence of an $\mathsf{EF1}$ allocation for monotone valuations; however, the time complexity of the procedure is not clear. This is closely related to the representation and the knowledge we have of the valuation functions.
At every iteration of the FOR-loop, after adding in the next item, the envy graph has to be updated, and therefore all agents' valuation functions have to be evaluated on the new bundle. The evaluation cannot be considered an atomic operation unless we have an oracle. Let us denote by T^* the time complexity for determining the value of any bundle by any agent. Then the runtime caused by the parts of the FOR loop not including the WHILE loop is $O(mnT^*)$.

To bound the overall number of cyclic trades within the WHILE loop, note that by performing a cycle, at least two edges disappear in the envy-graph. How many edges are introduced during the whole execution? In every iteration of the FOR loop at most $n-1$ new edges are introduced. Hence overall at most $O(mn)$ edges are introduced, and thus the number of cycles performed is also at most $O(mn)$. Performing a cycle can be done in $O(n^2)$, e.g., by using adjacency matrices for graph representation. The complexity of envy-cycle elimination is thus $O(mnT^* + n^3m)$, and this can be considered polynomial-time, since T^* is an intrinsic value depending on the given valuations.

2.4 EF1 and Efficiency

We now turn our attention again to additive valuations. Since EF1 allocations do always exist, we may try to ask for further properties for such a solution. A compelling notion is the one of efficiency, usually defined as Pareto optimality. Roughly speaking, we do not want to create waste while achieving EF1.
We have seen that maximizing welfare functions such as the utilitarian or the Nash social welfare leads to Pareto optimal allocations in cake cutting. It remains true also for indivisible goods (it is sufficient to apply the very same arguments). Interestingly, it turns out that any allocation maximizing Nash social welfare is also particularly fair.

Remark. There are scenarios in which the maximum Nash social welfare is 0. Consider for example the case of two agents and an item. In this case, we at first maximize the number of agents having positive value for their bundle, then maximize the Nash welfare among these agents. We call such allocations *Nash optimal.*

Theorem 18. *Let $\mathcal{A}$ be a Nash optimal allocation, then $\mathcal{A}$ is also* EF1.

Proof. Let be $\mathcal{A}$ a Nash optimal allocation.
Let us assume $\text{NSW}(\mathcal{A}) \neq 0$, which means, every agent has a positive value for her bundle, and therefore no bundle is empty. It is possible to show that the statement holds true even if $\text{NSW}(\mathcal{A}) = 0$ with a careful adaptation of the proof.
We want to show that for every $i, j \in \mathcal{N}$, there exists $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j) - v_i(g)$.
If $v_i(A_j) = 0$ the claim trivially follows since i does not envy j.
Otherwise, since $\mathcal{A}$ is Nash optimal, by moving any item $g \in A_j$ to A_i the Nash social welfare cannot improve. Hence,

$$\begin{aligned} v_i(A_i) \cdot v_j(A_j) \geq (v_i(A_i) + v_i(g)) \cdot (v_j(A_j) - v_j(g)) \qquad & \Leftrightarrow \\ v_j(g) \cdot (v_i(A_i) + v_i(g)) \geq v_i(g) \cdot v_j(A_j) \qquad & \end{aligned}$$

showing that, for each $g \in A_j$

$$v_i(A_i) + v_i(g) \geq \frac{v_i(g)}{v_j(g)} \cdot v_j(A_j) \ . \tag{1}$$

Let us select $g^* = \arg\min_{g \in A_j, v_i(g) > 0} \frac{v_j(g)}{v_i(g)}$. Notice that g^* is well defined; in fact, there must exist at least one positively valued good in j's bundle according to i's valuations because $v_i(A_j) > 0$. By definition of g^*, it holds

$$\frac{v_j(g^*)}{v_i(g^*)} \leq \frac{\sum_{g \in A_j} v_j(g)}{\sum_{g \in A_j} v_i(g)} \leq \frac{v_j(A_j)}{v_i(A_j)}$$

and hence, by inverting terms, $\frac{v_i(A_j)}{v_j(A_j)} \leq \frac{v_i(g^*)}{v_j(g^*)}$.
This inequality together with (1) shows that $g^* \in A_j$ is such that $v_i(A_i) \geq v_i(A_j) - v_i(g^*)$, concluding the proof. □

In conclusion, for additive valuations there exists an allocation that is simultaneously EF1 and Pareto optimal.

Proposition 19. *Under additive valuations, an allocation that is simultaneously* EF1 *and PO always exists.*

This is only an existential result, computing a maximum NSW allocation is in general hard, even for two agents with identical valuations.
So far we discussed the existence of EF1 and hence PROP1 allocations. Are there any other meaningful relaxations for EF and PROP?

2.5 Beyond EF1 Allocations – Envy-Freeness up to any Good

EF1 represents in first approximation a good relaxation of the EF fairness concept. While we know that EF $\Longrightarrow$ EF1, we see an example in which an EF1 allocation might be quite unfair.

Example 11. *Consider a fair division instance with two agents, three goods, and additive valuations depicted in Table 2.*
Let us consider the allocation $A_1 = \{g_1, g_3\}$ and $A_2 = \{g_2\}$. This allocation is EF1 *as agent 2 is allowed to remove g_1 to eliminate the envy. However, according to agent 2, $v_2(A_1) = 13$ and $v_2(A_2) = 3$ which is a quite large gap between the two bundles.*

	g_1	g_2	g_3
Agent 1	10	2	1
Agent 2	10	3	2

Table 2: Valuations in Example 11.

Roughly speaking, in the definition of EF1, allowing to remove *some* good could be too "generous", what about the removal of *any*?

Definition 15 (Envy-freeness up to Any Good)**.** *An allocation $\mathcal{A}$ is called* envy-free up to any good (EFX) *if, for each $i, j \in \mathcal{N}$, either $A_j = \emptyset$ or for every $g \in A_j$ such that $v_i(g) > 0$ it holds*

$$v_i(A_i) \geq v_i(A_j \setminus \{g\}) \ .$$

Example 12. *Consider the allocation in Table 1. The allocation is* EF1 *but not* EFX.
Consider for the same instance the allocation $A_1 = \{g_4, g_5\}$, $A_2 = \{g_2, g_3\}$, $A_3 = \{g_1\}$. This is an EFX *allocation.*

The following implications easily follow.

Proposition 20. EF $\Longrightarrow$ EFX $\Longrightarrow$ EF1

It is also easy to see that backward directions do not hold.

2.5.1 On EFX Existence

Unfortunately, the existence of EFX-allocations is unknown, even for additive valuations! It is only known to be guaranteed for special cases like two or three agents or identical valuations.

Two agents. We show the existence of EFX allocations for two agents.

Theorem 21. EFX *allocations always exists for $n = 2$ and can be efficiently computed.*

Proof. We show the existence by providing an algorithm that is a discrete version of the CutAndChoose protocol.

- Agent 1 computes a partition (A_1, A_2) such that $v_1(A_1) \geq v_1(A_2)$ and $v_1(A_1 \setminus \{g\}) \leq v_1(A_2)$ for each $g \in A_1$ such that $v_1(g) > 0$;
- Agent 2 selects the most favorite bundle for her;
- Agent 1 gets the remaining bundle.

The resulting allocation is EFX. Indeed, agent 2 does not envy agent 1. Agent 1, no matter which bundle she receives, is EFX by the conditions on the two bundles. We only need to clarify how to compute such a partition.
To compute the partition (A_1, A_2) we proceed as follows[1]:

- Sort goods in a non-increasing order of values according to 1, that is, $v_1(g_1) \geq v_1(g_2) \geq \ldots \geq v_1(g_m)$
- $(A_1, A_2) \leftarrow (\emptyset, \emptyset)$
- allocate items in the ordering $g_1 \ldots g_m$ to the bundle A_i, $i = 1, 2$, of minimum value for agent 1.

Assume without loss of generality that $v_1(A_1) \geq v_1(A_2)$, otherwise we change names to the bundles. We have that $v_1(A_1 \setminus \{g\}) \leq v_1(A_2)$ for each $g \in A_1$, such that $v_1(g) > 0$, must hold. Notice that this means that for 1, receiving A_2 would make her EFX. Hence, it is sufficient to show that $v_1(A_1 \setminus \{g^*\}) \leq v_1(A_1)$ for $g^* \in A_1$ such that $v_1(g^*) > 0$ and g^* is the least valued good in A_1. This holds true since before inserting g^* in A_1 the value of A_1 was smaller or equal to the value of A_2, according to 1. Furthermore, g^* is the smallest positively valued good in A_1, therefore the statement follows. □

Identical additive valuations. Now we assume that the agents have identical additive valuations, that is, $v_i = v$ for each $i \in \mathcal{N}$ where v is additive.
We start by showing that EFX allocations always exist in this setting.

Theorem 22. *If agents have identical valuations, every Nash optimal allocation is* EFX.

Proof. Assume $\mathcal{A}$ is Nash optimal. We denote by v the valuation function of the agents. Therefore, moving any positively valued g from any A_j to any A_i cannot strictly improve the Nash social welfare. Formally,

$$\begin{aligned} v(A_i) \cdot v(A_j) \geq (v(A_i) + v(g)) \cdot (v(A_j) - v(g)) \qquad & \Leftrightarrow \\ v(g) \cdot (v(A_i) + v(g)) \geq v(g) \cdot v_j(A_j) \,. & \end{aligned}$$

Since $v(g) > 0$, we have for each i, j and $g \in A_j$, $v(A_i) \geq v_j(A_j) - v(g)$ and the statement follows.
Notice we assumed that the NSW is not 0. If so, it means that at least one agent has no item in her bundle, and hence there are not enough items for the agents. Recall that in this case, we assume that we first maximize the number of agents with a positive value for their bundle and then the Nash welfare among those agents. Such an allocation assigns each agent at most one item, and hence it is EFX. □

Unfortunately, it is hard to compute such an allocation, even for $n = 2$.

Theorem 23. *It is NP-hard to compute a Nash optimal allocation, even with identical valuations and $n = 2$.*

Proof. Reduction from PARTITION.
Reduction:
We create a far division instance with two agents and identical valuations v. We set (with an abuse of notation) $\mathcal{G} = X$ and $v(x) = x$.
It is sufficient to notice that, for identical valuation, the more balanced the values of the two bundles the higher the Nash welfare. Therefore, denoted by $z = \sum_{i=1}^{t} x_i$, a partition exists if and only if the maximum Nash welfare is of $z/2$. □

Despite this negative result, it is still possible to compute an EFX allocation with good welfare guarantees. We next consider a greedy algorithm to compute an EFX allocation.

[1]Notice we will use the same idea in the next paragraph for identical valuations.

Partition

Input: A set $X = \{x_1, \dots, x_t\}$ of positive values
Problem: Does there exist a partition of X, i.e. $(S, X \setminus S)$, s.t. $\sum_{x \in S} x = \sum_{x \in X \setminus S} x$?

Greedy algorithm for identical valuations:

- $\mathcal{A} \leftarrow (\emptyset, \dots, \emptyset)$
- sort items $g_1, \dots, g_m$ in a non-increasing order, that is, $v(g_1) \geq v(g_2) \geq \dots \geq v(g_m)$
- For $h = 1, \dots, m$
  - $i \leftarrow \arg\min_{i \in \mathcal{N}} v(A_i)$ (break ties arbitrarily)
  - $A_i \leftarrow A_i \cup \{g_h\}$

Theorem 24 (Barman et al. 2018)**.** *The greedy algorithm for identical valuations computes an* EFX *allocation which is an* 1.061 *approximation in terms of Nash social welfare.*

Proof. We only prove EFX. In particular, at every iteration EFX is satisfied.
At the beginning, the allocation is empty, and therefore, EFX.
Consider an arbitrary iteration where good g is allocated to, say, agent i. Before allocating g no one envies i since she has the minimum valued bundle and the allocation is EFX. For every $j, k \neq i$, after allocating g, the allocations remains EFX if i is not considered. Let us now consider i and observe that no j EFX-envies i since g is the minimum item in the new bundle (the algo introduces items in a non-increasing order) and before introducing g so one was envying i. On the other hand, also i does not EFX-envy any j, since her bundle has increased. □

References

[1] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):1–28, 2017.

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

[3] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 7–13, 2018.

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

[5] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[6] Katarína Cechlárová and Eva Pillárová. A near equitable 2-person cake cutting algorithm. *Optimization*, 61(11):1321–1330, 2012.

[7] Katarína Cechlárová and Eva Pillárová. On the computability of equitable divisions. *Discrete Optimization*, 9(4):249–257, 2012.

[8] Lester E Dubins and Edwin H Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1P1):1–17, 1961.

[9] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. In *SODA*, volume 6, pages 271–278, 2006.

[10] Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984.

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

[12] Ariel D Procaccia. Thou shalt covet thy neighbor's cake. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[13] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692, 2014.

[14] Erel Segal-Halevi and Balázs R Sziklai. Monotonicity and competitive equilibrium in cake-cutting. *Economic Theory*, 68(2):363–401, 2019.