Algorithmic Game Theory - Fair Division **Winter 2022/23**

Lectures 22 and 23: The cake-cutting problem

Lecturers: Prof. Dr. Martin Hoefer, Dr. Giovanna Varricchio *Scribes: Dr. Giovanna Varricchio*

Summary: In the cake-cutting problem we are given a divisible resource, a.k.a. a *cake*, which has to be split among n agents. Each agent has some preference over different pieces and the goal is to split the cake among agents in a fair manner. In this lectures, we first introduce two criteria for establishing fairness: we define the notion of proportionality and of envy-freeness and discuss their existence and computation. We also have a look into further desirable properties other than fairness; namely, Pareto and Nash optimality. Finally, we give a short overview on equitability and its computation.

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).
- *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 Setting

We are given a divisible resource, a *cake* C, which is mathematically represented by the interval $[0, 1]$, and a set $\mathcal{N} = \{1, \ldots, n\}$ of n agents. We call a *piece of cake* any subset of C that it is an union of disjoint intervals in C; a piece of cake is said to be *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 $\mathcal{A} = (A_1, \ldots, A_n)$ *is a partition of C into pieces each of them assigned to an 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 said to be *simple* if each agent receives a connected piece of C, i.e. $\mathcal{A}$ is obtained by cutting C in $n - 1$ points. The goal is to compute a *fair* allocation of the cake to the agents. We will later define what fair means, before that, we need to introduce how agents evaluate any piece of the cake.

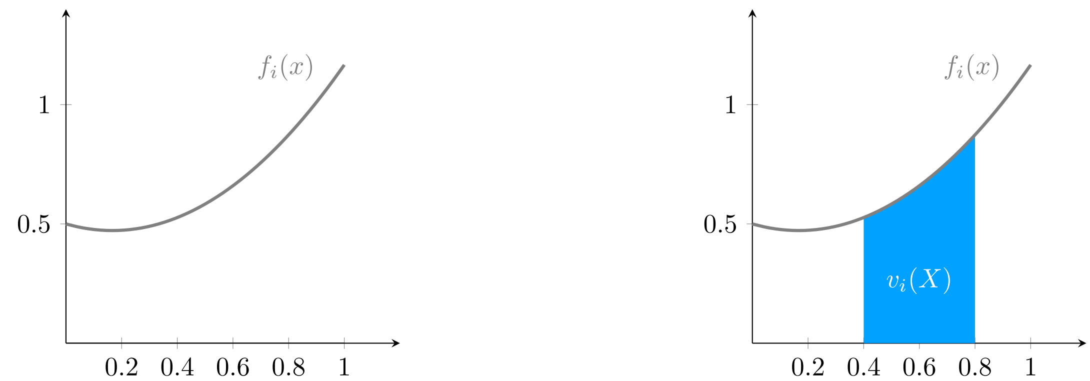
1.1 Agents' valuations

We ask each agent to be able to 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, an example of 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 $X = [\frac{2}{5}, \frac{4}{5}]$.*



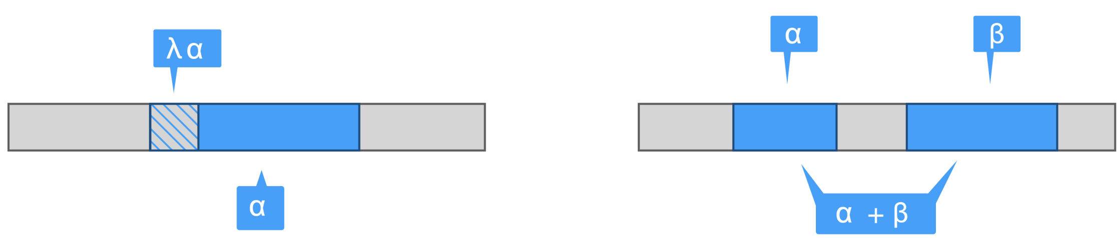
(a) Density function of agent i: $f_i(x) = x^2 - \frac{1}{3}x + \frac{1}{2}$.

(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. Under the aforementioned assumptions, for each $i \in \mathcal{N}$, v_i satisfies the following:

- *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: Valuations' properties.

REMARK! Whenever we say the value of a piece in a given allocation it is meant the value for the owner.

1.2 Query Complexity

We should consider the density functions of the agents as a possible input. This could imply infinitely many bits for representation. In this setting, it is more reasonable to have an oracle able to reply to some meaningful questions about agents' valuations.

Robertson-Webb model. We assume there exists an oracle able to reply to the following two queries:

- $Eval_i(x, y)$ returns $v_i([x, y])$;

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

We, therefore, evaluate the performances of an algorithm by its *query complexity*, that is, the number of queries required during its execution.

2 Fairness Criteria

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

2.1 Definitions

We distinguish between threshold-based and comparison-based criteria. A threshold-based criteria requires that each agent receives at least a prefixed 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.
The most natural threshold-value one can think of is the *proportional share* which is the value an agent has for 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, thus, each agent has a proportional share of $\frac{1}{n}$.

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

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

Concerning comparison-based criteria, the most reasonable one can think of is *envy-freeness* which requires that no agent envies no other agent.

Definition 3 (Envy-freeness)**.** *An allocation $\mathcal{A}$ is said to be* 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*.*

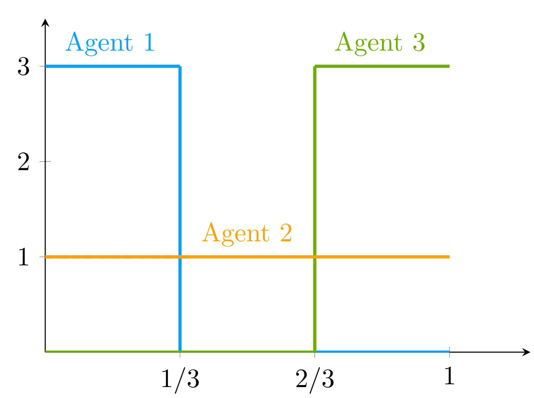


Figure 3: Example instance with three agents.

2.2 Implications between fairness notions

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

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

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

Summing up for all $i \in \mathcal{N}$ we get

$$n \cdot v_i(A_i) \geq \sum_{j \in \mathcal{N}} v_i(A_j) \overset{(1)}{=} v_i\left(\cup_{j \in \mathcal{N}} A_j\right) \overset{(2)}{=} v_i(C) \overset{(3)}{=} 1 \ ,$$

and therefore $v_i(A_i) \geq \frac{1}{n}$. Note that (1) holds because of the additive property, (2) because the cake is completely allocated while (3) because of normalization.
Applying the same argument to all the agents the thesis follows. □

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 $v_i(A_j) = v_i(C \setminus A_i) \leq \frac{1}{2} \leq v_i(A_i)$. □

3 Existence and Computation of fairness criteria

We now discuss the existence and computation of the aforementioned fairness criteria. Let us start by considering the simplest scenario where only two agents are involved. We will then discuss the more general setting where the number of agents is any $n \geq 2$.

3.1 Fairness for two agents – 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.
The protocol 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 $Eval_2(0, y)$.

3.2 Proportionality

We now turn our attention to the computation of a proportional allocation for n agents. Proportionality is the easiest criteria to achieve and several protocols have been defined.

3.2.1 Dubins-Spainer protocol – Achieving proportionality with $O(n^2)$ queries

In this section, we introduce the Dubins-Spainer protocol also known as the MovingKnife.
Here, we pretend to move a knife along our cake starting from position 0. As soon as we reach one point of the cake which values at least the proportional share of some agent i, we cut the cake in that point and assign the piece to i. Then, i is removed as well as the assigned piece of cake. We then start again (from the current position), we move the knife to the right, and find again the first position where a new agent gets her proportional share.
Formally, let x_k 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, that is, if $\exists i \in \mathcal{N}$ 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 continues on the remaining cake and the remaining agents.

- Finally, when $|\mathcal{N}|= 1$, assign $[y_\ell, 1]$ to the unique agents in $\mathcal{N}$ 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? Using, at round ℓ, $Cut_i(y_{\ell-1}, \frac{1}{n})$ for all $i \in \mathcal{N}$. Roughly speaking, we ask to all the agents when they will shout. By taking the minimum of all these positions we know who will be the shouter and where to cut.

Example 3. *Let us give 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 to every agent where to cut to obtain her proportional share, i.e. the protocol asks for $Cut_i(0, 1/n)$ for each $i \in \{1, 2, 3\}$, and selects the agent declaring the minimum value. Since, $Cut_i(0, 1/n) = 1/9$, $Cut_2(0, 1/n) = 1/3$, and $Cut_3(0, 1/n) = 7/9$, agent 1 is selected and receives $[0, 1/9]$. Now only agents 2 and 3 are involved in the process, and the knife is positioned in $x_k = 1/9$. Since $Cut_2(1/9, 1/n) = 4/9$ and $Cut_3(1/9, 1/n) = 7/9$, 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: by definition of the algorithm every agent receives a contiguous piece of cake.
Proportional: by definition of the algorithm every agent (but the last one) is receiving a piece will of value her proportional share. We need to show that the protocol never consumes the entire cake before every agent receives one piece and the remaining of the cake is enough to guarantee the proportional share to the remaining agents.
To this aim, 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 it means that 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 to the remaining agents where we should cut to give him her proportional share. Hence, we need $\sum_{i=1}^{n-1} n - i + 1 = \theta(n^2)$ queries.
Can we do better?

3.2.2 Even-Paz protocol – Achieving 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 $n = 1$ give $[x, y]$ to the agent and terminate;
- otherwise 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 $[x, y]$;
- Repeat on $[x, z^*]$ with the left $n/2$ agents, and on $[z^*, y]$ with the right $n/2$ agents.

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

Proof idea. Invariant on each recursion: 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 → $O(n \log n)$.

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!

3.3 Envy-freeness

We have seen that the CutAndChoose protocol provides an EF allocation for two agents. In around 1960, Selfridge and Conway (independently) constructed the same envy-free algorithm for the case of three agents.

3.3.1 Envy-freeness for three agents: The Selfridge-Conway protocol

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 denote the agent $i \in \{2, 3\}$ that received the trimmed piece by T, and the other agent by $\overline{T}$;
- 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.

Have an example on the instance depicted in Figure 3.

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

Proof. Let us denote by C_1 and C_2 cake 1 and 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 thesis follows.
Otherwise, let us consider the final allocation of C (that is after the division of C_1 and $C_2 \neq \emptyset$): The agent $\overline{T}$ who is splitting the cake is the one who will receive the remaining piece of C_2; anyway, she will be EF in the final allocation as the three pieces of C_2 are equally liked. The agent selecting first is also EF in the final allocation. It remains to show agent 1 is EF.
Agent 1 won't envy $\overline{T}$ since 1 selects before than $\overline{T}$ the piece of C_2. We need to show that agent 1 does not 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 $\{X_1 \setminus X'\} \cup C_2 = X_1$ and 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. □

3.3.2 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.

4 Efficiency

For the sake of 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*.*

The just provided 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.

4.1 Pareto optimality

The most prominent definition of efficiency is given by the one of *Pareto optimality.* Roughly speaking, an allocation is said to be 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 a per se interesting property, we will look for fair allocations which also satisfy such a property.

4.2 Nash social welfare – A way to achieve efficiency

A way to obtain Pareto optimal allocations is to use the optimization problem related to some welfare function. It turned out that maximizing the Nash social welfare function is an extremely fair objective and also guarantees 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.

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. *Any allocation $\mathcal{A}$ maximizing the NSW is PO.*

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

Notice that maximum Nash social welfare is scale-invariant, that is, if we divide the valuations of an agent by a positive number this does not change the set of Nash optimal allocations.
An interesting aspect of maximum Nash social welfare allocation is that it is a good (and fair) trade-off between maximum egalitarian and utilitarian social welfare allocation.

- 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 others.

5 Envy-free allocation existence

In this section, we discuss the existence of EF allocations.
The existence of EF allocations is known to be related to the strong Competitive Equilibrium from Equal Incomes (sCEEI) existence. In particular, an allocation is a sCEEI if and only if it is Nash optimal, and any sCEEI is EF and therefore it holds true also for maximum NSW allocations. This led to the following theorem:

Theorem 11. *Any maximum NSW allocation is also* EF.

In what follows, we show the theorem without using the connection with the competitive equilibrium. Let's first have an intuition on why the statement is true.

Example 6. *Consider an instance with two agents. Agent 1 has density function $U[0, 1/2]$ while agent 2 density function $U[0, 1]$. Given the allocation $\mathcal{A}$, where $A_1 = [0, 1/6]$ and $A_2 = [1/6, 1]$. Notice that $\mathcal{A}$ 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 of $\sqrt{\frac{1}{2} \cdot \frac{3}{4}} = \sqrt{\frac{3}{8}}$ and therefore the Nash welfare is increased.

Proof. Let $\mathcal{A}$ be a maximum NSW allocation. Let us assume by contradiction that the allocation is not EF. Therefore, 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 by moving Z from A_j to A_i the NSW improves.
To this aim, 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. Therefore, the following conditions hold:

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

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 sufficient to compare the product of the utilities of i and j in the two allocations.
Formally:

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

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

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

Therefore if $-\frac{1}{k} \cdot v_i(A_i) \cdot v_j(A_j) + \frac{1}{k}\left(1 - \frac{1}{k}\right) \cdot v_j(A_j) \cdot v_i(A_j) > 0$ we get our contradiction. Such an inequality holds iff $-v_i(A_i) + \left(1 - \frac{1}{k}\right) \cdot v_i(A_j) > 0$ if and only if $k > \frac{v_i(A_j)}{v_i(A_j) - v_i(A_i)}$.
To conclude, it is possible to find a piece of cake Z such that by moving Z from A_j to A_i the NSW improves. This is a contradiction to the optimality of $\mathcal{A}$, and hence the thesis follows. □

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

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

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

6 Another fairness notion – Equitability

To conclude our discussion on fairness, we finally introduce a different comparison-based notion that takes into account not only pairs of pieces but also the valuations of the owners.

Definition 6 (Equitability)**.** *An allocation $\mathcal{A}$ is said to be* equitable (EQ) *if for each $i, j \in \mathcal{N}$ it holds*

$$v_i(A_i) = v_j(A_j) \ .$$

Proposition 13. *Equitability is incomparable with both proportionality and envy-freeness.*

Proof. Consider an instance with two agents where agent 1 has a positive value only for the first half of the cake and agent 2 values only the second half. The allocation where $A_1 = [\frac{1}{2}, 1]$ and $A_2[0, \frac{1}{2}]$. Such an allocation is neither proportional nor envy-free.
It is also possible to provide an allocation that is envy-free, and therefore proportional, but not equitable. □

Theorem 14. *An equitable allocation always exists.*

We show the claim for only two agents.

Proof. We set $g_1(x) = v_1([0, x])$, which is an non-decreasing and contiguous function with $g_1(0) = 0$ and $g_1(1) = 1$, and $g_2(x) = v_2([x, 1])$, which is a non-increasing and contiguous function with $g_2(0) = 1$ and $g_2(1) = 0$. There exists x^* such that $g_1(x^*) = g_2(x^*)$. Therefore, the allocation $A_1 = [0, x^*]$ and $A_2 = [x^*, 1]$ is equitable. □

Problem: There is no way to implement an equitable protocol with a finite number of queries in the Robertson-Webb model!

For two players, to find x^* we can use a bisection algorithm, which possibly requires an infinite number of refinements steps to find the location of x^*.
However, if we perform a bounded amount of such refinements we can still obtain a good approximation to equability leading to the next theorem.
The bisection algorithm takes as input agents valuations and some value ε and proceeds as follows:

- Set $a \leftarrow Cut_1(0, \frac{1}{2})$, $b \leftarrow Cut_2(0, \frac{1}{2})$
- If $a = b$ output a and terminate
- If $b < a$ exchange agents identities, therefore, in what follows we always have $a > b$
- Initialize $j \leftarrow 2$, $c \leftarrow 0$
- While $(\frac{1}{2})^j \geq \varepsilon$ do
  - $x \leftarrow Cut_1(a, (\frac{1}{2})^j)$, $y \leftarrow Cut_2(c, (\frac{1}{2})^j)$
  - if $x = y$ return x and terminate
  - if $x < y$ then $a \leftarrow x$ and $b \leftarrow y$
  - if $x > y$ then $c \leftarrow y$
  - $j \leftarrow j + 1$
- output $(a + b)/2$

An allocation $\mathcal{A}$ is ε-EQ if $|v_i(A_i) - v_j(A_j)| \leq \varepsilon$ for each i, j.

Theorem 15 (Cechlárová and Pillárová, 2012)**.** *Using the bisection algorithm, for two agents, it is possible to find an ε-EQ allocation with $O(\log(1/\varepsilon))$ queries.*

The above theorem can be extended to any number of agents.

Theorem 16 (Cechlárová and Pillárová, 2012)**.** *It is possible to find an ε-EQ allocation with $O(n \cdot \log(1/\varepsilon))$ queries.*

References

[1] 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.

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

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

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

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

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

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

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