Caking Cutting

Thomas Kesselheim   **Preview** Last Update: July 1, 2019

Today, we will consider a problem of fair division. Again our goal is to find a mechanism that works without money and makes all involved agents happy. Indeed, we experience such problems throughout our daily life. Suppose a group of employees have to assign their work shifts. Maybe for someone it is more important to not work at night, whereas somebody else would prefer to be off work during the World Cup final.

Our focus today will be on *cake cutting*. We are given a cake and have to divide it among the agents. It is different from the above example because the good can be divided arbitrarily. The cake is not the same everywhere and therefore the agents' preferences differ.

1 Model

The cake is modeled by the interval $[0, 1]$. We can cut it into pieces; each piece X is a union of subintervals of $[0, 1]$.

There are n agents N and each agent $i \in N$ has a valuation function V_i, which assigns to each piece X non-negative value $V_i(X)$. For simplicity, we assume that there is a valuation density function $v_i \colon [0, 1] \to \mathbb{R}_{\geq 0}$ such that the valuation V_i of X is determined as the integral of v_i over X

$$V_i(X) = \int_{x \in X} v_i(x)\mathrm{d}x \ .$$

So, in particular, if $X = [a, b]$, then $V_i(X) = \int_a^b v_i(x)\mathrm{d}x$.

If X is the union of disjoint intervals, then the valuation for X is the sum of the valuations for the intervals. That is, if $X = [a, b] \cup [c, d]$ for $b < c$, then

$$V_i(X) = \int_a^b v_i(x)\mathrm{d}x + \int_c^d v_i(x)\mathrm{d}x = V_i([a, b]) + V_i([c, d]) \ .$$

Note that it does not actually matter if we assign open or closed intervals because $\int_a^a v_i(x)\mathrm{d}x = 0$.

We assume that $V_i([0, 1]) = \int_0^1 v_i(x)\mathrm{d}x = 1$ for all i. That is, each agent i values the entire cake as 1 but the value will usually be distributed differently along the interval.

2 Fairness Properties

We will find an allocation $A = (A_1, \ldots, A_n)$, where the A_i are disjoint and their union is $[0, 1]$. Each A_i is a union of intervals.

What makes an allocation fair? There are three standard notions:

- *Proportionality:* For all $i \in N$, $V_i(A_i) \geq \frac{1}{n}$.

- *Envy-Freeness:* For all $i, j \in N$, $V_i(A_i) \geq V_i(A_j)$.

- *Equitability:* For all $i, j \in N$, $V_i(A_i) = V_i(A_j)$.

If all valuations are identical, these notions coincide. Proportionality requires that each agent values his own piece at least $\frac{1}{n}$. This, however, does not exclude that another agent gets a piece that he values even higher. This is explicitly forbidden in envy-freeness. In an envy-free allocation, no agent would like to get another agent's piece instead. Envy-freeness

implies proportionality because there always has to be one j for which $V_i(A_j) \geq \frac{1}{n}$. Otherwise $V_i([0,1]) < 1$.

Equitability is an even stronger requirement by asking that an agent would be equally happy with another agent's piece.

3 Two Agents: Cut and Choose

There is a world-famous cake cutting protocol for two agents: cut and choose. Agent 1 cuts the point t such that $\int_0^t v_1(x)\mathrm{d}x = \int_t^1 v_1(x)\mathrm{d}x = \frac{1}{2}$. Then agent 2 chooses between $[0,t)$ and $[t,1]$, depending on which value is higher for him.

This allocation is proportional: $V_i(A_i) \geq \frac{1}{2}$ for both i. For agent 1, this is clear because he cuts the cake exactly such that both pieces have value $\frac{1}{2}$. For agent 2, one of the two pieces has value at least $\frac{1}{2}$.

The allocation is also envy-free: $V_1(A_1) = V_1(A_2)$ by definition. Furthermore, $V_2(A_2) \geq V_2(A_1)$ because agent 2 chooses the preferred piece.

However, the allocation is not necessarily equitable. Suppose that $v_1(x) = 1$ for all x and $v_2(x) = 2$ for $x \leq \frac{1}{2}$ and 0 otherwise. In this case $t = \frac{1}{2}$ and $V_2(A_2) = 1$, $V_2(A_1) = 0$.

In this example, there is also an equitable allocation: Set $A_1 = [\frac{1}{4}, \frac{3}{4}]$, $A_2 = [0,1] \setminus A_1$. In this allocation, both agents value their piece exactly as $\frac{1}{2}$. In contrast, there is no *contiguous* equitable allocation. It is impossible to assign to each agent only a single interval such that both are equally valuable for both.

4 Proportionality for Any Number of Agents

There is a reasonably simple algorithm by Dubins-Spanier (1961) to determine a proportional allocation for any number of agents n. The algorithm runs for n iterations. In each of them, one agent is allocated and leaves. We determine cuts $t_1, \ldots, t_{n-1}$ such that in the first iteration, we allocate $[0, t_1)$, in the second $[t_1, t_2)$ and so on.

- Initialize $t_0 = 0$, $N_1 = N$

- For $k = 1$ to $n - 1$

  - For each agent i in N_k, let $t_{k,i}$ be the value such that $V_i(t_{k-1}, t_{k,i}) = \frac{1}{n}$.
  - Let i^* be the agent with the smallest (i.e. leftmost) $t_{k,i}$ and let $t_k = t_{k,i^*}$, $A_{i^*} = [t_{k-1}, t_k)$.
  - Set $N_{k+1} = N_k \setminus \{i^*\}$.

- Assign the remainder to the remaining agent in N_n.

Theorem 23.1. *The allocation computed by the algorithm fulfills proportionality.*

The idea is that in every iteration we only remove an interval that no agent values more than $\frac{1}{n}$. Therefore, at any point in time, there is enough cake left to make the respective cuts.

Proof. Note that there could, in principle, be two reasons why the allocation does not fulfill proportionality. On the one hand, it could be that one of the t_k is larger than 1, which would make the algorithm ill-defined. On the other hand, it could be that the remaining agent values the remainder less than $\frac{1}{n}$.

Both is ruled out by the following invariant: For any k and any $i \in N_k$, $V_i(t_{k-1}, 1) \geq \frac{n-k+1}{n}$. That is, at the beginning of every iteration, the agents value the remaining cake at least $\frac{n-k+1}{n}$.

We can show the invariant by induction on k. For $k = 1$, it is trivially true. So, let us now consider some $k + 1 > 1$. For every $i \in N_{k+1}$, we have $t_{k,i} \geq t_k$ because t_k is the smallest value that we saw in round k. Therefore, $V_i(t_{k-1}, t_k) \leq V_i(t_{k-1}, t_{k,i}) = \frac{1}{n}$. By this inequality and by induction hypothesis,

$$V_i(t_k, 1) = V_i(t_{k-1}, 1) - V_i(t_{k-1}, t_k) \geq \frac{n - k + 1}{n} - \frac{1}{n} = \frac{n - (k+1) + 1}{n} \ .$$

This completes the induction.   □

It is easy to see that for three or more agents this allocation is not always envy-free. It can easily happen that the agent who got the first piece envies one of the other agents because they get a piece that he values more than $\frac{1}{n}$.

5   Envy-Freeness for Three Agents

Finding an envy-free allocation is significantly more involved. But it is possible. In the following, we will consider the case of exactly three agents. There, it is sufficient to make five cuts, meaning that we get six intervals. The algorithm has an interesting history. It is named after John Selfridge and John Horton Conway, who both discovered it independently but unfortunately it was never published by either of them.

The algorithm is as follows:

- Agent 1 cuts the cake into three pieces X_1, X_2, X_3 such that $V_1(X_1) = V_1(X_2) = V_1(X_3)$.

- Rename the pieces such that $V_2(X_1) \geq V_2(X_2) \geq V_2(X_3)$.

- Agent 2 cuts off $X' \subseteq X_1$ such that $V_2(X_1 \setminus X') = V_2(X_2)$.

First assignment phase, assigning $[0, 1] \setminus X'$:

- Agent 3 chooses one of $X_1 \setminus X'$, X_2, and X_3.

- If agent 3 chose X_2 or X_3, agent 2 gets $X_1 \setminus X'$, otherwise he gets X_2.

- Agent 1 gets X_2 or X_3 depending on what is left.

Second assignment phase, assigning X': Let $T \in \{2, 3\}$ be the agent who got $X_1 \setminus X'$ in the first phase, $\bar{T} \in \{2, 3\}$ the other one.

- Agent $\bar{T}$ cuts X' into three pieces X_1', X_2', X_3' such that $V_{\bar{T}}(X_1') = V_{\bar{T}}(X_2') = V_{\bar{T}}(X_3')$.

- Agent T chooses one of X_1', X_2', X_3'.

- Agent 1 chooses among the remaining two.

- Agent $\bar{T}$ gets the remaining piece.

Theorem 23.2. *The allocation is envy-free.*

Proof. Without loss of generality, we can assume that in the second phase agent T chooses X_1', agent 1 chooses X_2' and X_3' is left over for agent $\bar{T}$.

We now have to argue that no agent envies the outcome of the other agent.

Consider agent T. He gets $(X_1 \setminus X') \cup X_1'$. Regardless of whether this is agent 2 or agent 3, we always have $V_T(X_1 \setminus X') \geq V_T(X_2)$ and $V_T(X_1 \setminus X') \geq V_T(X_3)$. Furthermore, $V_T(X_1') \geq V_T(X_2')$ and $V_T(X_1') \geq V_T(X_3')$ because he chooses first. So, no matter how the remaining pieces are allocated among the other agents, he never envies one of them.

First cuts by agent 1:
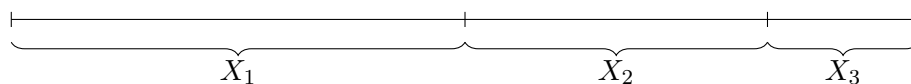
Agent 2 makes the larger two pieces the same size:
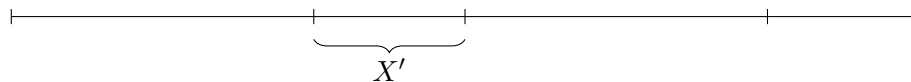
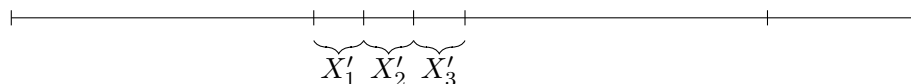Agent 2 or 3 cuts X' into three equal parts:

Figure 1: A potential outcome.

Now, consider agent $\bar{T}$. If this is agent 2, then he left a piece of the exact same value as $X_1 \setminus X'$ that he will select in the first phase and another one that might have a smaller value. Agent 3, by definition, chooses his most preferred piece in the first phase. Therefore, agent $\bar{T}$ does not consider any piece in the first assignment phase larger than his own. In the second assignment phase, he considers all pieces identically valuable because he is the one to cut them.

Finally, let us consider agent 1. He does not envy agent T because agent T only gets a subset of X_1, whereas agent 1 cut the original pieces so that $V_1(X_1) = V_1(X_2) = V_1(X_3)$. He does not envy agent $\bar{T}$ either. We have $V_1(X_2) = V_1(X_3)$ because he cut these pieces and $V_1(X_2') \geq V_2(X_3')$ because he chooses first. $\qquad\square$

6 Outlook

The topic of fair division is an active research area. For example, how can one find an envy-free allocation for any number of agents? Recently, there was a result that the number of steps is always bounded in terms of a function that only depends on n, regardless of the valuation functions. However, one is very far from "efficient" algorithms. This may not be surprising given how complicated things become already for $n = 3$.

There is also the question of incentive compatibility. As a matter of fact, none of the algorithms so far is robust against agents strategically misreporting their valuations.

Further Reading

- Chapter 13 by Ariel Procaccia in "Handbook of Computational Social Choice" edited by Brandt, Conitzer, Endriss, Lang, and Procaccia.