Cake Cutting Algorithms

Fair Allocation of Divisible Goods

Joseph Chuang-Chieh Lin

Department of Computer Science & Engineering,
National Taiwan Ocean University

Fall 2026

You may find that cake cutting is not just a piece of cake.

DOI:10.1145/2483852.2483870

How to fairly allocate divisible resources, and why computer scientists should take notice.

BY ARIEL D. PROCACCIA

Cake Cutting: Not Just Child's Play

ADDRESSING SOME OF the great challenges of the 21st century involves a thorough understanding of fairness considerations. Fairly dividing limited natural resources (such as fossil fuels, clean water, and the environment's capacity to absorb greenhouse gases) is of utmost importance; for example, fairness plays a key role in the evolution of the United Nations Framework Convention on Climate Change, a global treaty for combating climate change, and its implementation; indeed, its language demands fairness. It seems intuitive that countries with developing economies would bear less of the burden, and developed countries, which share historical responsibility for pollution, bear more. However, only a concerted effort involving developing countries would yield a meaningful result. Moreover, some countries (such as small islands) are more susceptible to the adverse effects of climate change and call for global solidarity. With so many competing needs and demands, what would be a fair solution to dividing up the costs of pollution?

Similar (though simpler) questions about fairness have tantalized thinkers for millennia and can be traced back to the Bible and to ancient Greece. Indeed, the Bible famously recounts the tale of King Solomon's role in a maternity dispute. In response to two women both claiming to be the true mother of the same baby, Solomon suggested his guards cut the baby in two and give each woman half. When one of the women protested, begging Solomon to give the baby to the other woman, he declared she must be the true mother (1 Kings 3:16–27).

The 20th century experienced a shift toward mathematically rigorous approaches to fairness. In mathematics, the formal notion of "envy-freeness," with participants preferring to keep their own allocation to swapping with other participants, appeared in *Puzzle-Math*, a 1958 book of mathematical puzzles by Gamow and Stern.[23] In the economics literature, Foley[21] initiated the formal study of envy-freeness. The notion of proportionality, with each of n participants receiving at least $1/n$ of each participant's own value for getting everything, was considered by Steinhaus[39] as early as the 1940s.

In economics today, fair division is considered a significant subfield of microeconomic theory. Unlike Solomon, the literature usually distinguishes between allocation of divisible goods such as land (see Figure 1), time, and memory on a computer, and indivisible goods (such as a house or the computer itself, or a baby). Each variation has attracted ample attention, as in the book by Moulin.[31] However, com-

» key insights

- A cake-cutting algorithm divides a heterogeneous divisible good in a way that achieves formal fairness guarantees.
- Designing cake-cutting algorithms that are computationally efficient and immune to manipulation is a challenge for computer scientists.
- Insight from the study of cake cutting can be applied to the allocation of computational resources.

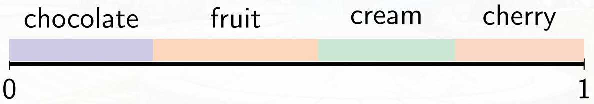
ILLUSTRATION BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

Outline

Outline

Why Cake Cutting?

- The "cake" is a heterogeneous divisible resource: land, time, bandwidth, advertising slots, or computational resources.
- Agents may value different parts of the cake differently.
- **Goal:** design protocols that give formal fairness guarantees using limited information.



Computational Social Choice Viewpoint

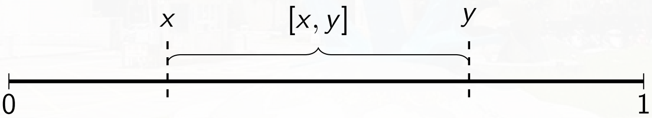
Two concerns interact

1. **Normative:** Which fairness properties should the allocation satisfy?
2. **Algorithmic:** How many operations or queries are needed to find such an allocation?

- Cake cutting is a clean testbed for continuous-input fair division.
- It also exposes the tension between fairness and social welfare.

Basic Setting

- Agents: $N = \{1, 2, \ldots, n\}$.
- Cake: the interval $[0, 1]$.
- Agent i has a valuation function V_i over intervals.
- We write $V_i(x, y)$ for $V_i([x, y])$.



$V_i(x, y)$ is agent i's value for this interval.

Assumptions on Valuations

For every agent i, the valuation V_i satisfies:

- **Normalization:** $V_i(0,1) = 1$.
- **Divisibility:** for any $[x, y]$ and $0 \leq \lambda \leq 1$, there exists $z \in [x, y]$ with

$$V_i(x, z) = \lambda V_i(x, y).$$

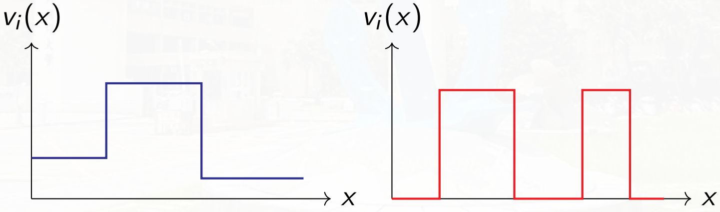
- **Nonnegativity:** $V_i(I) \geq 0$ for every interval I.
- **Additivity:** if I and I' are disjoint, then

$$V_i(I \cup I') = V_i(I) + V_i(I').$$

Value Density Functions

Often V_i is induced by a density v_i:

$$V_i(X) = \int_{x \in X} v_i(x)\,dx, \qquad \int_0^1 v_i(x)\,dx = 1.$$



piecewise constant

piecewise uniform

Pieces and Allocations

Definition

A **piece** *of cake is a finite union of disjoint intervals.*

Definition

An **allocation** *is* $A = (A_1, A_2, \ldots, A_n)$, *where* A_i *is agent* i*'s piece, the pieces are disjoint, and their union is the whole cake.*

Agent 2's piece is non-contiguous here.

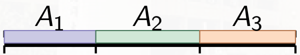
Contiguous vs. Non-Contiguous Allocations

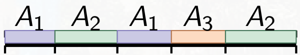
Contiguous

Each A_i is a single interval.

Non-contiguous

A piece may be a union of intervals.





Contiguity is natural in some applications, but it often makes algorithmic problems much harder.

Fairness Properties

Proportionality

Every agent receives at least its fair share:

$$V_i(A_i) \geq \frac{1}{n} \quad \forall i \in N.$$

Fairness Properties

Proportionality

Every agent receives at least its fair share:

$$V_i(A_i) \geq \frac{1}{n} \quad \forall i \in N.$$

Envy-freeness

No agent prefers another agent's piece:

$$V_i(A_i) \geq V_i(A_j) \quad \forall i, j \in N.$$

Fairness Properties

Proportionality

Every agent receives at least its fair share:

$$V_i(A_i) \geq \frac{1}{n} \quad \forall i \in N.$$

Envy-freeness

No agent prefers another agent's piece:

$$V_i(A_i) \geq V_i(A_j) \quad \forall i, j \in N.$$

Equitability

All agents assign the same value to their own pieces:

$$V_i(A_i) = V_j(A_j) \quad \forall i, j \in N.$$

Envy-Freeness Implies Proportionality

Claim 1

Every envy-free allocation is proportional.

Proof idea.

Fix agent i. Since the pieces partition the cake,

$$\sum_{j \in N} V_i(A_j) = V_i([0,1]) = 1.$$

Hence there exists some j such that $V_i(A_j) \geq 1/n$. If the allocation is envy-free,

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

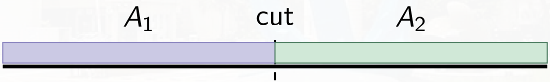
□

For **Two** Agents: Proportionality = Envy-Freeness

For $n = 2$, if $V_i(A_i) \geq 1/2$, then by additivity

$$V_i(A_{3-i}) = 1 - V_i(A_i) \leq 1/2 \leq V_i(A_i).$$

So agent i does not envy the other agent.



If each side is worth at least $1/2$ to its owner, no envy is possible.

For **Three or More** Agents: The Notions Split

- Envy-freeness is stronger than proportionality.
- A proportional allocation can still generate envy.

Example pattern

For $n = 3$, agent i may have

$$V_i(A_i) = \frac{1}{3}, \qquad V_i(A_j) = \frac{1}{2}$$

for another agent j. Proportionality holds for i, but envy-freeness fails.

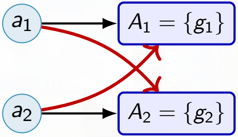
$$\boxed{V_i(A_i) = 1/3} \xrightarrow{\text{envy}} \boxed{V_i(A_j) = 1/2}$$

Note: Equitability Does Not Imply EF or PROP

Example

	g_1	g_2
a_1	0.4	0.6
a_2	0.6	0.4

Allocate $A_1 = \{g_1\}, \quad A_2 = \{g_2\}$.



- Equitable: since $v_1(A_1) = 0.4 = v_2(A_2)$.
- Not envy-free:

$$v_1(A_2) = 0.6 > 0.4 = v_1(A_1), \quad v_2(A_1) = 0.6 > 0.4 = v_2(A_2).$$

- Not proportional:

$$v_i(A_i) = 0.4 < \frac{v_i(M)}{2} = 0.5 \quad \text{for } i = 1, 2.$$

Outline

A Nonconstructive Benchmark

Theorem (Alon, 1987)

If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

A Nonconstructive Benchmark

Theorem (Alon, 1987)

If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

- A very strong existence benchmark: every agent values every piece equally

A Nonconstructive Benchmark

Theorem (Alon, 1987)

If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

- A very strong existence benchmark: every agent values every piece equally (equitability &

A Nonconstructive Benchmark

Theorem (Alon, 1987)

If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

- A very strong existence benchmark: every agent values every piece equally (equitability & envyfreeness

A Nonconstructive Benchmark

Theorem (Alon, 1987)

If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

- A very strong existence benchmark: every agent values every piece equally (equitability & envyfreeness & proportionality).
- We focus on constructive algorithms: how can we actually find fair allocations?

A Nonconstructive Benchmark

Theorem (Alon, 1987)

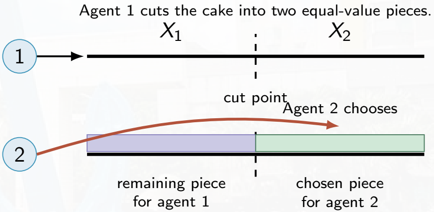
If $V_1, V_2, \ldots, V_n$ are induced by continuous density functions, the cake can be cut in $n^2 - n$ places and partitioned into n pieces $A_1, A_2, \ldots, A_n$ such that

$$V_i(A_j) = \frac{1}{n} \qquad \forall i, j \in N.$$

- A very strong existence benchmark: every agent values every piece equally (equitability & envyfreeness & proportionality).
- We focus on constructive algorithms: how can we actually find fair allocations?
    - Achievable but it requires $n^2 - n$ continuous operations [Brams et al. 2006].

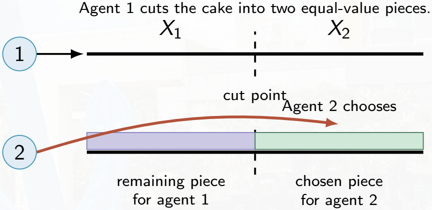
Cut and Choose: $n = 2$

1. Agent 1 cuts the cake into two pieces it values equally.
2. Agent 2 chooses the preferred piece.
3. Agent 1 gets the remaining piece.



Cut and Choose: $n = 2$

1. Agent 1 cuts the cake into two pieces it values equally.
2. Agent 2 chooses the preferred piece.
3. Agent 1 gets the remaining piece.

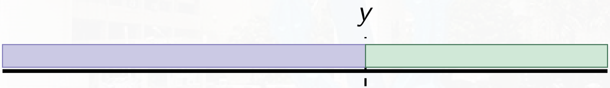


Proportional, and for two agents therefore envy-free.

Why Cut and Choose Works

- Agent 1 can guarantee value 1/2 by making an equal cut according to V_1.
- Agent 2 can guarantee value at least 1/2 by choosing its preferred side.

$V_1(0, y) = 1/2$ $\qquad$ $V_1(y, 1) = 1/2$

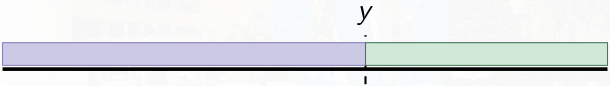


The cut need not be geometrically in the middle.

Why Cut and Choose Works

- Agent 1 can guarantee value 1/2 by making an equal cut according to V_1.
- Agent 2 can guarantee value at least 1/2 by choosing its preferred side.

$$V_1(0, y) = 1/2 \qquad\qquad V_1(y, 1) = 1/2$$



The cut need not be geometrically in the middle.

(one continuous operation is required)

Strategic Robustness of Cut and Choose

Dominant-style intuition

An agent can secure its fair share by following the protocol, regardless of the other agent's behavior.

- Cutter: if the two pieces are equal in its valuation, either remaining piece is worth $1/2$.
- Chooser: choosing its preferred piece gives value at least $1/2$.

This gives a maximin-style guarantee, but it is not a full strategic analysis of the induced game.

Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

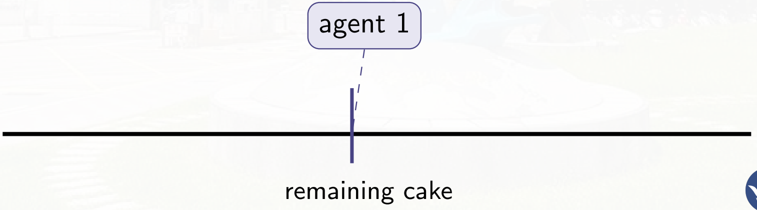
The leftmost mark determines the prefix to be allocated.

remaining cake

Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

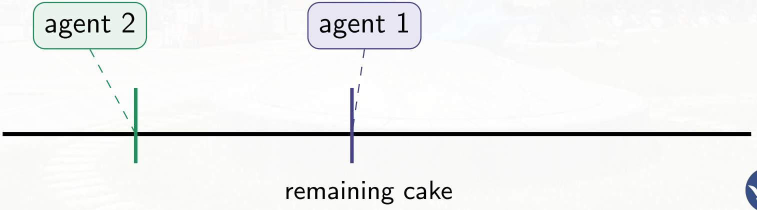
The leftmost mark determines the prefix to be allocated.



Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

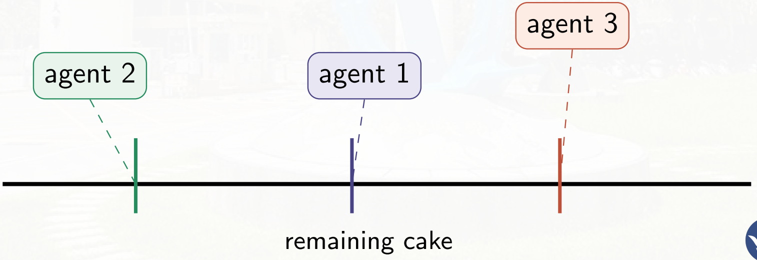
The leftmost mark determines the prefix to be allocated.



Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

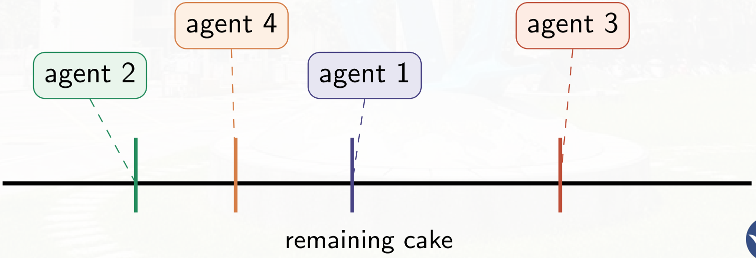
The leftmost mark determines the prefix to be allocated.



Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

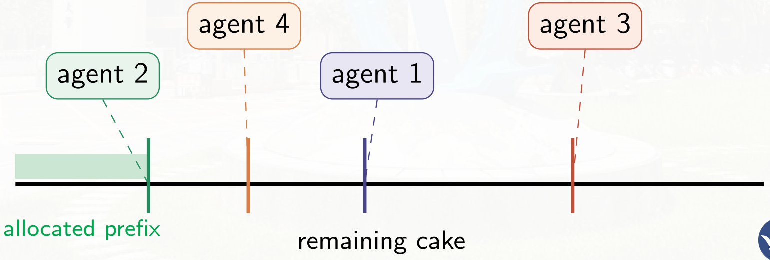
The leftmost mark determines the prefix to be allocated.



Dubins–Spanier (1961): Proportionality for Any n

1. Each remaining agent marks a point making the left prefix worth $1/n$ of the original cake.
2. The leftmost mark wins that prefix and exits.
3. Repeat with the remaining cake and agents.

The leftmost mark determines the prefix to be allocated.



Why Dubins–Spanier Is Proportional?

Suppose agent i^* has the leftmost mark and receives $[0, x_{i^*}]$.

- By construction, $V_{i^*}(0, x_{i^*}) = 1/n$.
- For any other remaining agent j, leftmost means $x_{i^*} \leq x_j$, so

$$V_j(0, x_{i^*}) \leq V_j(0, x_j) = 1/n.$$

- Thus no remaining agent loses more than $1/n$ value when this piece is removed.

Inductively, everyone gets value at least $1/n$ of the original cake.

Why Dubins–Spanier Is Proportional?

Suppose agent i^* has the leftmost mark and receives $[0, x_{i^*}]$.

- By construction, $V_{i^*}(0, x_{i^*}) = 1/n$.
- For any other remaining agent j, leftmost means $x_{i^*} \leq x_j$, so

$$V_j(0, x_{i^*}) \leq V_j(0, x_j) = 1/n.$$

- Thus no remaining agent loses more than $1/n$ value when this piece is removed.

Inductively, everyone gets value at least $1/n$ of the original cake.

- For the last agent j, $V_j(A_j) \geq 1 - (n-1)/n = 1/n$.

Dubins–Spanier Is Not Envy-Free

agents exiting later may receive pieces
that earlier agents prefer

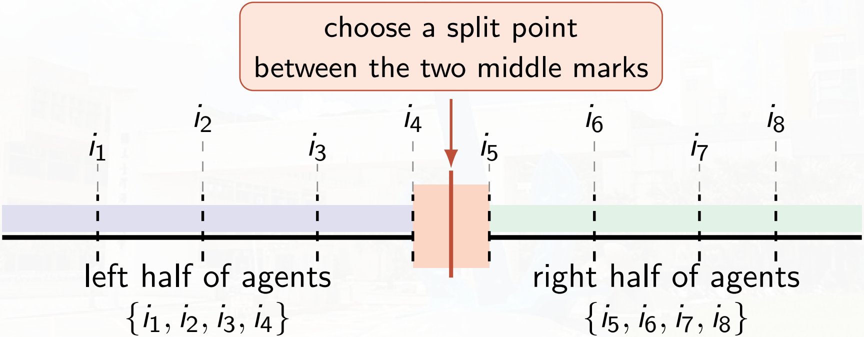
- Earlier exiting agents secure value $1/n$.
- An agent would never envy earlier agents, but they may still envy a later agent's piece.

Even–Paz (1984): Divide-and-Conquer

For simplicity assume n is a power of 2.

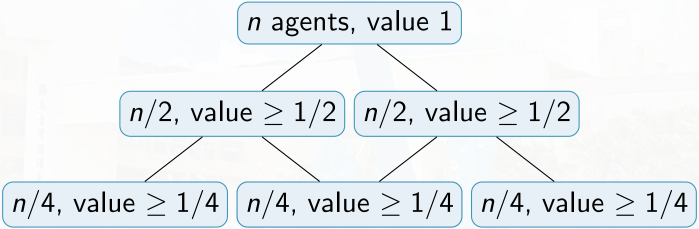
1. Given agents S and interval $[y, z]$, ask each $i \in S$ to mark x_i with
$$V_i(y, x_i) = \frac{1}{2} V_i(y, z).$$
2. Sort the marks from left to right.
3. Recurse on the left half of agents with the left subcake, and on the right half with the right subcake.

Even–Paz: One Recursive Split



The key invariant: each group of agents receives a subcake that each of them values sufficiently highly.

Even–Paz Recursion Tree



At depth k: each active agent values its current subcake at least $1/2^k$. At depth $\lg n$: each leaf agent gets value at least $1/n$.

Even–Paz: Why Proportional?

At recursive depth k:

- each active group has $n/2^k$ agents;
- every agent in that group values the group's cake at least $1/2^k$.

At depth $\lg n$, each group has one agent, and that agent receives a piece worth at least

$$\frac{1}{2^{\lg n}} = \frac{1}{n}.$$

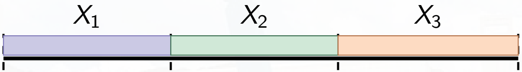
Thus the allocation is proportional.

Selfridge–Conway: Envy-Freeness for $n = 3$

- First bounded envy-free algorithm for three agents.
  - Bounded number of cake-cutting operations.
- Core idea: trim the most valuable piece for agent 2 so it ties with the second-best piece.
- Allocate the main cake carefully, then allocate the trimming in a compensating order.

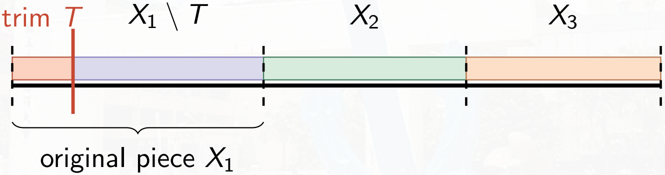
Selfridge–Conway: Initialization

1. Agent 1 cuts the cake into three pieces it values equally: X_1, X_2, X_3.



Selfridge–Conway: Initialization

1. Agent 1 cuts the cake into three pieces it values equally: X_1, X_2, X_3.
2. Agent 2 trims its favorite piece (say X_1) so that it ties with its 2nd favorite.



Agent 2 trims her favorite piece X_1 so that $V_2(X_1 \setminus T) = V_2(X_2)$, assuming X_2 is her second-best piece.

Now the trimmed piece and the 2nd-best piece are tied according to agent 2.

Selfridge–Conway: Allocate the Main Cake

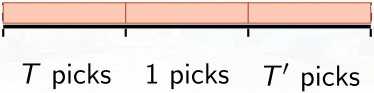
3. Agent 3 chooses first from the three main pieces.
4. If agent 3 did not choose the trimmed piece, agent 2 gets the trimmed piece (i.e., $X_1 \setminus T$ here).
5. Otherwise agent 2 chooses from the two remaining pieces.
6. Agent 1 receives the last main piece.

3 chooses first → 2 protected by trim → 1 gets untrimmed remainder

Selfridge–Conway: Allocate the Trimming

Let T be the agent among $\{2, 3\}$ who received the trimmed main piece, and let T' be the other.

6. Agent T' cuts the trimming into three equal-value pieces.
7. Selection order: T, then agent 1, then T'.



Selfridge–Conway: Proof Sketch

- Main cake is envy-free:
  - agent 3 chooses first;
  - agent 2 receives a piece tied for best in its valuation;
  - agent 1 receives an untrimmed piece, which it values as one of the original thirds.
- Trimming allocation protects T and T' by the cut-and-choose logic.
- Agent 1 cannot use the trimming to envy T overall: even all the trimming would only reconstruct one original piece for agent 1.

Classic Algorithms: Comparison

Algorithm	Agents	Guarantee	Pieces
Cut and choose	2	Prop. + EF	contiguous
Dubins–Spanier	any n	Prop.	contiguous
Even–Paz	any n	Prop.	contiguous
Selfridge–Conway	3	EF	≤ 2 intervals per agent

EF = envy-free; Prop. = proportional.

Outline

Why a New Complexity Model?

Standard input-size analysis is not directly applicable:

- a valuation function over $[0, 1]$ may not have a finite discrete representation;
- algorithms often interact with agents through cuts and value reports;
- complexity is measured by the number of permitted queries.

Question

How many valuation queries are necessary and sufficient to guarantee fairness?

The Robertson–Webb Model

Two query types are allowed.

Evaluation query

$$\mathrm{eval}_i(x, y) = V_i(x, y).$$

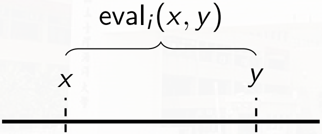
Agent i reports its value for interval $[x, y]$.

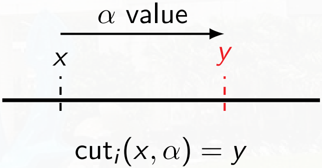
Cut query

$$\mathrm{cut}_i(x, \alpha) = y,$$

where y is the leftmost point such that $V_i(x, y) = \alpha$.

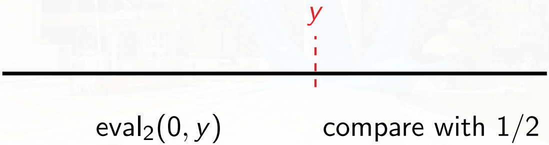
Query Types





Simulating Cut and Choose

- Ask agent 1: $\mathrm{cut}_1(0, 1/2) = y$.
- Ask agent 2: $\mathrm{eval}_2(0, y)$.
- If agent 2 values $[0, y]$ at least $1/2$, give it $[0, y]$; otherwise give it $[y, 1]$.



Query Complexity: Dubins–Spanier

At a stage with k remaining agents, ask each remaining agent one cut query.

$$k = n, n-1, \ldots, 2.$$

Thus the number of queries is

$$\sum_{k=2}^{n} k = \Theta(n^2).$$

n queries $\longrightarrow$ $n-1$ $\longrightarrow$ $\cdots$ $\longrightarrow$ 2

Query Complexity: Even–Paz

At each recursive level, every active agent makes one cut query.

- Level 0: n queries.
- Level 1: $2 \cdot (n/2) = n$ queries.
- Level k: $2^k \cdot (n/2^k) = n$ queries.

There are $\lg n$ levels, so the total is

$$n \lg n = \Theta(n \log n).$$

Optimality of Even–Paz

Theorem (Edmonds–Pruhs, SODA 2006)

Any proportional cake-cutting algorithm requires

$$\Omega(n \log n)$$

queries in the Robertson–Webb model.

Consequence

Even–Paz is asymptotically optimal for proportional cake cutting in this query model.

Lower Bound Strategy: Thin-Rich Pieces

Fix an agent i.

A piece X is:

- **thin** if $\ell(X) \leq 2/n$;
- **rich** if $V_i(X) \geq 1/n$;
- **thin-rich** if both conditions hold.

Lemma

If finding a thin-rich piece needs $T(n)$ queries, then proportional cake cutting needs $\Omega(nT(n))$ queries.

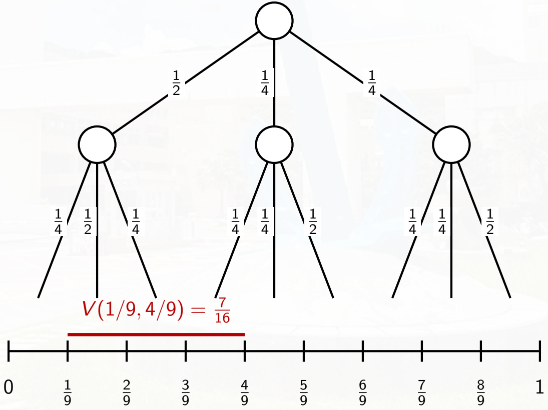
Why Thin-Rich Pieces Matter?

- In any proportional allocation, every allocated piece is rich for its owner.
- At most $n/2$ allocated pieces can have length more than $2/n$.
- Therefore, in any proportional allocation, at least $n/2$ agents receive thin-rich pieces.

A proportional allocation must locate many small high-value pieces.

Value Tree Idea

The lower bound uses adversarial valuation functions represented by a ternary tree.



Value Tree: Rich Leaves Require Heavy Edges

Let the height be $H = \log_3(n/2)$. If a leaf u has $q(u)$ heavy edges on its root-to-leaf path, then

$$V(u) = \left(\frac{1}{2}\right)^{q(u)} \left(\frac{1}{4}\right)^{H-q(u)} = \left(\frac{1}{4}\right)^{H-\frac{q(u)}{2}} .$$

Value Tree: Rich Leaves Require Heavy Edges

Let the height be $H = \log_3(n/2)$. If a leaf u has $q(u)$ heavy edges on its root-to-leaf path, then

$$\frac{1}{n} \leq V(u) = \left(\frac{1}{2}\right)^{q(u)} \left(\frac{1}{4}\right)^{H-q(u)} = \left(\frac{1}{4}\right)^{H-\frac{q(u)}{2}}.$$

To be rich, $V(u) \geq 1/n$.

Key intuition

A rich leaf must have a constant fraction of heavy edges on its path. Discovering that information costs $\Omega(\log n)$ queries.

By $2H - \lg n = \Omega(\lg n)$:

$$\begin{aligned}
\left(\frac{1}{4}\right)^{H-\frac{q(u)}{2}} &\geq \frac{1}{n} \\
\Rightarrow \quad 4^{H-\frac{q(u)}{2}} &\leq n \\
\Rightarrow \quad 2^{2H-q(u)} &\leq n \\
\Rightarrow \quad q(u) &= \Omega(\lg n).
\end{aligned}$$

From Tree Queries to Robertson–Webb Queries

The proof answers each Robertson–Webb query by revealing enough of the value tree.

- An $\mathrm{eval}(x, y)$ query reveals paths to the leaves containing x and y.
- A $\mathrm{cut}(x, \alpha)$ query reveals paths sufficient to identify the cut point.
- The adversary maintains that only $O(k)$ heavy edges are known after k queries.

RW query → Reveal tree paths → Few heavy edges known

From Tree Queries to Robertson–Webb Queries

The proof answers each Robertson–Webb query by revealing enough of the value tree.

- An $\mathrm{eval}(x, y)$ query reveals paths to the leaves containing x and y.
- A $\mathrm{cut}(x, \alpha)$ query reveals paths sufficient to identify the cut point.
- The adversary maintains that only $O(k)$ heavy edges are known after k queries.

$$\boxed{\text{RW query}} \longrightarrow \boxed{\text{Reveal tree paths}} \longrightarrow \boxed{\text{Few heavy edges known}}$$

Envy-Free Cake Cutting Is Harder

- Selfridge–Conway gives a bounded envy-free algorithm for $n = 3$.
- Brams–Taylor (1995) gives a *finite* envy-free algorithm for any n.
  - The algorithm is guaranteed to *terminate*.
- But for $n \geq 4$, the number of queries of Brams–Taylor is unbounded as a function of n alone.

Open-ended difficulty

Is there a bounded envy-free algorithm for arbitrary n?

Known Lower Bounds for Envy-Freeness

Theorem (Stromquist, Electron. J. Comb. 2008)

For $n \geq 3$, no finite envy-free cake-cutting algorithm outputs contiguous allocations.

Theorem (Procaccia, AAAI 2009)

Any envy-free cake-cutting algorithm requires $\Omega(n^2)$ queries in the Robertson–Webb model.

This separates envy-freeness from proportionality:

$$\Theta(n \log n) \text{ for proportionality} \quad \text{vs.} \quad \Omega(n^2) \text{ for envy-freeness.}$$

Piecewise Uniform Valuations Do Not Avoid the Difficulty

Theorem (Kurokawa–Procaccia–Wang, AAAI 2013)

Let $\mathcal{A}$ be an envy-free algorithm for piecewise uniform valuations using at most $f(n)$ Robertson–Webb queries. Then $\mathcal{A}$ can compute an envy-free algorithm for **general** *valuations using at most $f(n)$ queries.*

Interpretation (on query complexity)

Piecewise uniform valuations are expressive enough to capture the hard part of envy-free cake cutting.

ε-Envy-Freeness

Relax envy-freeness to

$$V_i(A_i) \geq V_i(A_j) - \varepsilon \qquad \forall i, j.$$

1. Ask each agent to cut the cake into $\lceil 1/\varepsilon \rceil$ disjoint intervals worth of value about ε.
2. Sort all $O(n/\varepsilon)$ cut points.
3. Treat intervals between adjacent cut points as indivisible goods.
4. Allocate these small goods by round-robin selection.

This gives a simple and efficient approximate envy-free allocation.

- $O(n/\varepsilon)$ cut queries and $O(n^2/\varepsilon)$ eval queries.

Round-Robin Argument for ε-EF

1 →	2 →	3 →	1 →	2 →	3
g_1	g_2	g_3	g_4	g_5	g_6

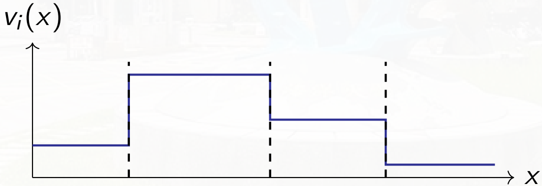
- After an agent first picks, it picks before each other agent in every subsequent round.
- The only possible envy comes from goods selected earlier, each worth at most ε.

Outline

From Query Model to Explicit Representation

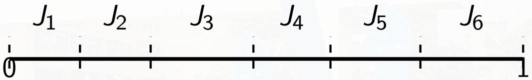
For valuations that are assumed to be explicitly represented:

- Example: piecewise constant densities.
- The input lists intervals and their density values.
- Now the computational task is to optimize over allocations, not to elicit valuations.
  - The notion of efficiency: Social Welfare.



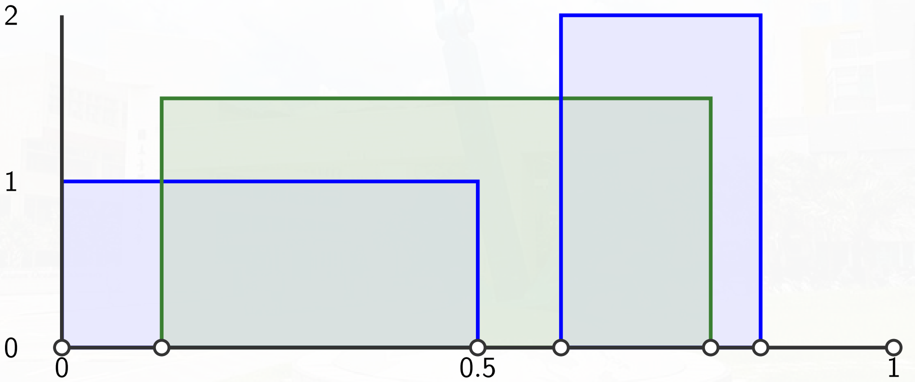
Common Refinement of Segments

Collect all boundaries reported by all agents, plus 0 and 1. Let $\mathcal{J}$ be the set of intervals between consecutive marks.



Crucial property

For each $J \in \mathcal{J}$ and each agent i, the density v_i is constant on J.


2
1
0
0
0.5
1

Computing Equitable and Envy-Free Allocations

If each refined interval J is split into n equal-length subintervals and each agent gets one subinterval from every J, then every agent values every final bundle equally:

$$V_i(A_j) = \frac{1}{n} \quad \forall i, j.$$

This is constructive under explicit piecewise constant valuations.

Social Welfare

The utilitarian social welfare of allocation A is

$$\mathrm{sw}(A) = \sum_{i \in N} V_i(A_i).$$

- Once welfare is introduced, fairness becomes a constraint in an optimization problem.

Optimization target

Maximize social welfare subject to proportionality, envy-freeness, or equitability.

LP for Optimal Proportional Allocation

Let f_i^J be the fraction of refined interval J allocated to agent i.

$$
\begin{aligned}
\max \quad & \sum_{i \in N} \sum_{J \in \mathcal{J}} f_i^J V_i(J) \\
\text{s.t.} \quad & \sum_{i \in N} f_i^J \leq 1 \qquad \forall J \in \mathcal{J}, \\
& \sum_{J \in \mathcal{J}} f_i^J V_i(J) \geq \frac{1}{n} \qquad \forall i \in N, \\
& f_i^J \geq 0 \qquad \forall i, J.
\end{aligned}
$$

- A polynomial-size linear program for explicitly represented piecewise constant valuations.

LP Constraints for Other Fairness Notions

Using the same variables f_i^J, other constraints are linear.

Envy-freeness

$$\sum_J f_i^J V_i(J) \geq \sum_J f_j^J V_i(J) \qquad \forall i, j.$$

Equitability

$$\sum_J f_i^J V_i(J) = \sum_J f_j^J V_j(J) \qquad \forall i, j.$$

Thus LPs can compute optimal fair allocations under non-contiguous pieces.

Contiguity Makes Optimization Hard

Theorem (Bei et al., AAAI 2012)

Given explicit piecewise constant valuations, if allocations must be both proportional and contiguous, then maximizing social welfare is NP-hard to approximate within a factor $\Omega(\sqrt{n})$.

Non-contiguous LP tractable	$\xrightarrow{\text{add restriction}}$	Contiguous hard to approximate

Outline

Price of Fairness: Definition

For a fairness property $\mathcal{F}$, define

$$\mathrm{PoF}(\mathcal{F}) = \sup_{V_1, V_2, \ldots, V_n} \frac{\max_A \mathrm{sw}(A)}{\max_{A \in \mathcal{F}} \mathrm{sw}(A)}.$$

- Numerator: unconstrained welfare optimum.
- Denominator: best welfare among fair allocations.
- Larger price means greater worst-case welfare loss due to fairness.

Price of Proportionality

Theorem (Caragiannis et al., WINE 2009)

The price of proportionality is

$$\Theta(\sqrt{n}).$$

Interpretation

A proportionality constraint can reduce optimal social welfare by a factor on the order of $\sqrt{n}$, and this bound is tight.

Upper Bound Idea for Proportionality

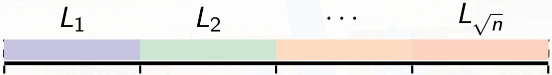
Let $\mathbf{A}^*$ be the unconstrained optimal allocation and define

$$L = \{i : V_i(A_i^*) \geq 1/\sqrt{n}\}, \qquad S = N \setminus L.$$

- L: the set of "large" agents; S: the set of "small" agents;
- If many agents are large $|L| \geq \sqrt{n}$, carefully redistribute pieces (e.g., Even-Paz algorithm) while losing at most a $\sqrt{n}$ factor. (skipped)
- If few agents are large, then the unconstrained social welfare is already $O(\sqrt{n})$, while any proportional allocation has welfare at least 1.
  - $\mathrm{sw}(\mathbf{A}^*) \leq |L| + |S|/\sqrt{N} < 2\sqrt{n}$, while for any proportional $\mathbf{A}$, $\mathrm{sw}(\mathbf{A}) \geq \sum_{i \in N} 1/n = 1$.

Lower Bound Idea for Proportionality

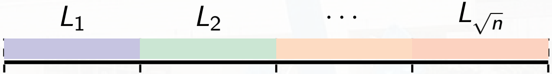
- Say the large agent set L has size exactly $\sqrt{n}$, each uniformly interested in a disjoint interval of length $1/\sqrt{n}$.
- The remaining agents (S) value the entire cake uniformly.



- Unconstrained welfare serves the large agents $\Rightarrow$ $\text{sw}(\mathbf{A}^*) = \sqrt{n}$.
- Any proportional $\mathbf{A}$ gives an interval of length $1/n$ to each small agent, leaving only $(1 - (n - \sqrt{n})/n) = 1/\sqrt{n}$ to agents in L.

Lower Bound Idea for Proportionality

- Say the large agent set L has size exactly $\sqrt{n}$, each uniformly interested in a disjoint interval of length $1/\sqrt{n}$.
- The remaining agents (S) value the entire cake uniformly.



- Unconstrained welfare serves the large agents $\Rightarrow$ $\mathrm{sw}(\mathbf{A}^*) = \sqrt{n}$.
- Any proportional $\mathbf{A}$ gives an interval of length $1/n$ to each small agent, leaving only $(1 - (n - \sqrt{n})/n) = 1/\sqrt{n}$ to agents in L.

$$\sum_{i \in L} V_i(A_i) + \sum_{j \in S} V_j(A_j) \leq \sqrt{n}/\sqrt{n} + 1 \leq 2.$$

Price of Envy-Freeness and Equitability

- Since envy-freeness implies proportionality, the $\Omega(\sqrt{n})$ lower bound also applies to envy-freeness.
- no nontrivial $o(n)$ upper bound on the price of envy-freeness is known.

Price of Envy-Freeness and Equitability

- Since envy-freeness implies proportionality, the $\Omega(\sqrt{n})$ lower bound also applies to envy-freeness.
- no nontrivial $o(n)$ upper bound on the price of envy-freeness is known.
- Equitability can be much more expensive.

Theorem (Caragiannis et al., WINE 2009)

The price of equitability is $\Theta(n)$.

Equitable vs. Envy-Free Welfare

Theorem (Brams et al., AAAI 2012)

For piecewise constant valuations, the optimal envy-free allocation has social welfare at least that of the optimal equitable allocation.

$$\text{Optimal equitable allocation} \xrightarrow{\text{sw(EF)} \geq \text{sw(EQ)}} \text{Optimal envy-free allocation}$$

Proof Tool: Linear Fisher Markets

A linear Fisher market has:

- divisible goods G;
- utilities u_{ij} for agent i and good j;
- budgets e_i;
- equilibrium prices p_j
- allocations f_{ij} (giving agent i a fraction f_{ij} of good j).
- utility of agent i for an allocation: $\sum_{j \in G} f_{ij} u_{ij}$.
- For all $j \in G$, $\sum_{i \in N} f_{ij} \leq 1$.

Fisher-Market View of Cake Cutting

Start from an equitable welfare-maximizing cake allocation

$$\mathbf{A}^* = (A_1^*, A_2^*, \ldots, A_n^*).$$

Construct a linear Fisher market

- Good j corresponds to cake piece A_j^*.
- Agent i's utility for good j is $u_{ij} = V_i(A_j^*)$.
- Budgets are identical: $e_i = \dfrac{1}{n} \quad \forall i \in N.$

$$\text{cake pieces } A_j^* \quad \longleftrightarrow \quad \text{divisible Fisher goods } j.$$

From Fisher Allocation Back to Cake

Given a Fisher allocation $\mathbf{f} = (f_{ij})_{i \in N, j \in G}$, we want a cake allocation A such that

$$V_i(A_j) = \sum_{k \in G} f_{jk} u_{ik} \quad \forall i, j \in N.$$

Idea

Each original cake piece A_k^* is subdivided into small intervals on which all agents' densities are uniform. Then an f_{jk}-fraction of each such interval is assigned to agent j.

Thus the utilities in the Fisher market can be replicated in the cake-cutting instance.

Why the Fisher Allocation Is Envy-Free?

Let $p = (p_1, p_2, \ldots, p_n)$ be equilibrium prices, and define $r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j}$.

By the Fisher-market property (Vazirani 2007), agent i receives only goods maximizing utility per price (bang-per-buck):

$$f_{ij} > 0 \quad \Longrightarrow \quad \frac{u_{ij}}{p_j} = r_i^*.$$

Why the Fisher Allocation Is Envy-Free?

Let $p = (p_1, p_2, \ldots, p_n)$ be equilibrium prices, and define $r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j}$.

By the Fisher-market property (Vazirani 2007), agent i receives only goods maximizing utility per price (bang-per-buck):

$$f_{ij} > 0 \quad \Longrightarrow \quad \frac{u_{ij}}{p_j} = r_i^*.$$

Hence (note: $\sum_{j \in G} f_{ij} p_j = e_i$; agents spend their entire budgets)

$$V_i(A_i) = \sum_{j \in G} f_{ij} u_{ij} = \sum_{j \in G} f_{ij} \frac{u_{ij}}{p_j} p_j = \sum_{j \in G} f_{ij} r_i^* p_j = \frac{r_i^*}{n}.$$

Why the Fisher Allocation Is Envy-Free?

Let $p = (p_1, p_2, \ldots, p_n)$ be equilibrium prices, and define $r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j}$.

By the Fisher-market property (Vazirani 2007), agent i receives only goods maximizing utility per price (bang-per-buck):

$$f_{ij} > 0 \quad \Longrightarrow \quad \frac{u_{ij}}{p_j} = r_i^*.$$

Hence (note: $\sum_{j \in G} f_{ij} p_j = e_i$; agents spend their entire budgets)

$$V_i(A_i) = \sum_{j \in G} f_{ij} u_{ij} = \sum_{j \in G} f_{ij} \frac{u_{ij}}{p_j} p_j = \sum_{j \in G} f_{ij} r_i^* p_j = \frac{r_i^*}{n}.$$

For another agent k,

$$V_i(A_k) = \sum_{j \in G} f_{kj} u_{ij} \leq \sum_{j \in G} f_{kj} r_i^* p_j = \frac{r_i^*}{n}.$$

Why the Fisher Allocation Is Envy-Free?

Let $p = (p_1, p_2, \ldots, p_n)$ be equilibrium prices, and define $r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j}$.

By the Fisher-market property (Vazirani 2007), agent i receives only goods maximizing utility per price (bang-per-buck):

$$f_{ij} > 0 \quad \Longrightarrow \quad \frac{u_{ij}}{p_j} = r_i^*.$$

Hence (note: $\sum_{j \in G} f_{ij} p_j = e_i$; agents spend their entire budgets)

$$V_i(A_i) = \sum_{j \in G} f_{ij} u_{ij} = \sum_{j \in G} f_{ij} \frac{u_{ij}}{p_j} p_j = \sum_{j \in G} f_{ij} r_i^* p_j = \frac{r_i^*}{n}.$$

For another agent k,

$$V_i(A_k) = \sum_{j \in G} f_{kj} u_{ij} \leq \sum_{j \in G} f_{kj} r_i^* p_j = \frac{r_i^*}{n}.$$

Therefore $V_i(A_i) \geq V_i(A_k)$, so **A** is envy-free.

Why Welfare Does Not Decrease

Since $\mathbf{A}^*$ is equitable, there exists $\alpha > 0$ such that $V_i(A_i^*) = \alpha \;\; \forall i \in N$.

Thus $\mathrm{sw}(\mathbf{A}^*) = n\alpha$.

Why Welfare Does Not Decrease

Since $\mathbf{A}^*$ is equitable, there exists $\alpha > 0$ such that $V_i(A_i^*) = \alpha \ \ \forall i \in N$.

Thus $\mathrm{sw}(\mathbf{A}^*) = n\alpha$.

From the previous frame, $\mathrm{sw}(\mathbf{A}) = \dfrac{1}{n}\sum_{i \in N} r_i^*$.

Why Welfare Does Not Decrease

Since $\mathbf{A}^*$ is equitable, there exists $\alpha > 0$ such that $V_i(A_i^*) = \alpha \ \ \forall i \in N$.

Thus $\mathrm{sw}(\mathbf{A}^*) = n\alpha$.

From the previous frame, $\mathrm{sw}(\mathbf{A}) = \dfrac{1}{n} \sum_{i \in N} r_i^*$.

Since

$$r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j} \geq \frac{u_{ii}}{p_i} = \frac{\alpha}{p_i},$$

we get

$$\mathrm{sw}(\mathbf{A}) \geq \frac{1}{n} \sum_{i \in N} \frac{\alpha}{p_i} = \frac{\alpha}{n} \sum_{i \in N} \frac{1}{p_i}.$$

Why Welfare Does Not Decrease

Since $\mathbf{A}^*$ is equitable, there exists $\alpha > 0$ such that $V_i(A_i^*) = \alpha \ \ \forall i \in N$.

Thus $\mathrm{sw}(\mathbf{A}^*) = n\alpha$.

From the previous frame, $\mathrm{sw}(\mathbf{A}) = \dfrac{1}{n} \displaystyle\sum_{i \in N} r_i^*$.

Since

$$r_i^* = \max_{j \in G} \frac{u_{ij}}{p_j} \geq \frac{u_{ii}}{p_i} = \frac{\alpha}{p_i},$$

we get

$$\mathrm{sw}(\mathbf{A}) \geq \frac{1}{n} \sum_{i \in N} \frac{\alpha}{p_i} = \frac{\alpha}{n} \sum_{i \in N} \frac{1}{p_i}.$$

Because $\sum_{i \in N} p_i = 1$, $\displaystyle\sum_{i \in N} \frac{1}{p_i} \geq n^2$ (Cauchy–Schwarz (Exercise)).

Therefore, $\mathrm{sw}(\mathbf{A}) \geq \dfrac{\alpha}{n} \cdot n^2 = n\alpha = \mathrm{sw}(\mathbf{A}^*)$.

Applications Beyond Literal Cake

- Time: scheduling access to facilities or backup servers.
- Land or physical resources: heterogeneous divisible goods.
- Advertising slots: different intervals can have different audiences.
- Computing resources: related models allocate CPU, RAM, and other resources under structured preferences.

Cake cutting is best understood as a mathematical abstraction for heterogeneous divisible resources.

Takeaways

1. Proportionality is algorithmically well understood in the Robertson–Webb model: $\Theta(n \log n)$ queries.
2. Envy-freeness is strictly harder; exact bounded algorithms are much more delicate.
3. Explicit piecewise constant valuations enable LP-based computation of optimal fair allocations.
4. Fairness can significantly reduce social welfare; the price depends strongly on the fairness notion.