7-1

Fair division

Examples:

divisible
- Division of spectrum resources
- Division of cost of renting a car / rideshare
- Allocation of budget into different portfolio

indivisible
- Division of inherited property among children
- Hostel room allocation
- Essential supplies, e.g., COVID vaccine, among different groups.

Divisible allocation of a single good : Cake Cutting

A cake is :

① heterogeneous : equal amounts of the good may have different values for an agent.

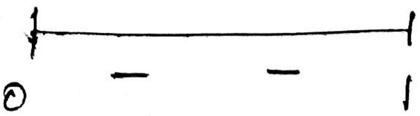
② divisible : arbitrarily fractional division is possible.

③ non-identically preferred : Same piece ~~on piece~~ can have different values for different agents.

④ A good : valuation for any piece is nonnegative

$\boxed{\text{Cake is an interval for our math model}}$

- Resource $= [0,1]$
- Agents $= \{1, 2, \dots, n\}$
- ~~Division~~ A piece of cake is a finite union of disjoint subintervals of $[0,1]$: $S_i \subseteq [0,1]$



$$S_i = I_{i1} \cup I_{i2} \cup \dots \cup I_{ik_i}, \quad I_{ik_1} \cap I_{ik_2} = \phi$$

Agent preferences:

Valuation function v_i: assigns a non-negative value to any piece of cake.

We will make two additional assumptions

① **Additivity:** $\forall X, Y \subseteq [0,1]$ s.t. $X \cap Y = \phi$

$$v_i(X \cup Y) = v_i(X) + v_i(Y)$$

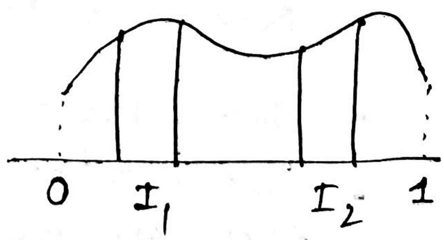
② **Divisibility:** For any $X \subseteq [0,1]$ and any $\lambda \in [0,1]$ there exists $Y \subseteq X$ s.t. $v_i(Y) = \lambda v_i(X)$

- Note that divisibility rules out atomic valuations. i.e., $v_i([x,x]) = 0 \quad \forall x, \forall i \in N$.
- Because of additive valuations and that cake is a good "more is always weakly better"
- divisibility says "an (arbitrarily small) trimming" is always possible - that will match any valuation.

③ **Normalization:** for each $i \in N$, $v_i([0,1]) = 1$.

Analog: valuation is a probability of falling within the piece. The heterogeneity is coming from a density function.

$$X = I_1 \cup I_2, \quad v_i(X) = \int_{x \in X} f_i(x)\,dx$$



Allocation / Division (of a cake)

A partition $(A_1, A_2, \ldots, A_n)$ of the cake $[0,1]$ where each A_i is a piece of cake assigned to agent i, and $\bigcup_{i \in N} A_i = [0,1]$.

7-3

$\boxed{\text{Desirable Fairness ideas}}$ ~~(examples)~~

① Proportionality: For each $i \in N$, $v_i(A_i) \geq \frac{1}{n}$ then $A = (A_1, \ldots, A_n)$ is proportional (PROP).

② Envy-freeness: $\forall\, i, j \in N$, $v_i(A_i) \geq v_i(A_j)$ then A is Envy-free (EF).

Which notion of fairness is stronger?

- Add the inequalities of EF for all $j \in N$.

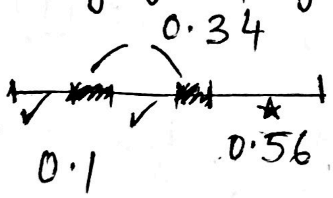
$$n\, v_i(A_i) \geq \sum_{j \in N} v_i(A_j) = 1 \Rightarrow v_i(A_i) \geq \frac{1}{n}.$$

EF ⇒ PROP for any number of agents

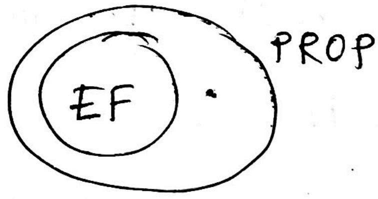
- PROP ⇒ EF for 2 agents

$$v_1(A_1) \geq \frac{1}{2} \Rightarrow 1 - v_1(A_2) \geq \frac{1}{2} \Rightarrow v_1(A_1) \geq \frac{1}{2} \geq v_1(A_2)$$

similarly for player 2 as well.



← PROP but not EF



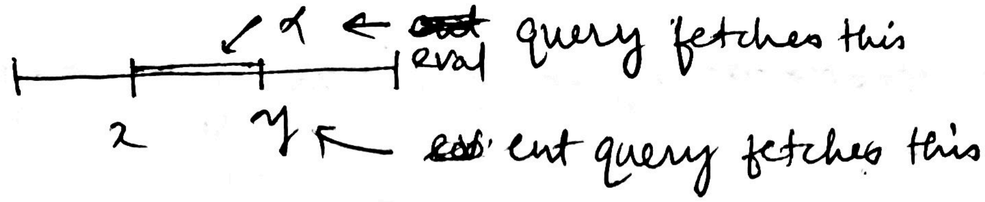
$\boxed{\text{Robertson-Webb Query model}}$ (RW 1998)

- What will be a good metric for complexity calculation?
- The input as valuation function is not appropriate since there are uncountably many possible valuation functions.
- Evaluation of an algorithm for cake cutting is done w.r.t. the number of queries it does to an oracle.

$\text{eval}_i(x, y)$: returns $v_i([x, y])$

$\text{cut}_i(x, \alpha)$: returns y s.t. $v_i([x, y]) = \alpha$

can return null if no such y exists.



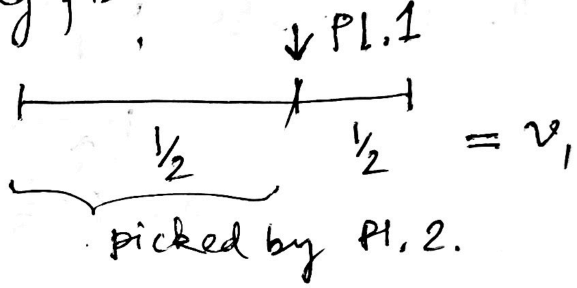
Cake-cutting algorithms!

Proportionality question first

Cut and Choose [I cut you choose]

1. Agent 1 cuts the cake into two equal valued pieces (as per v_1).
2. Agent 2 picks its preferred piece (as per v_2).

- Agent 1 gets the remaining piece.



PROP?

Yes. Player 2's value is at least $1/2$. Players 1's value is exactly $1/2$

EF? PROP $\Rightarrow$ EF for 2 agents.

What is the complexity according to RW query model?

Call $cut_1(0, 1/2)$ → gives the point where the value is $1/2$ for agent 1. Say it is y^*

Call $eval_2(0, y^*)$ → if $\geq 1/2$ give $[0, y^*]$ to agent 2
else give $[y^*, 1]$ to agent 2.

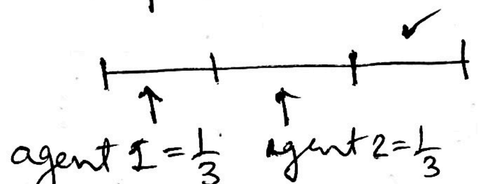
Takes 2 queries in RW query model.

Constant time irrespective of valuations for 2 agents.

A PROP cake-cutting algorithm for any number of agents.

$\boxed{\text{Dubins–Spanier (1961) Algorithm}}$

1. Mechanism designer gradually moves a knife from left → right. [gradually = continuously].
2. As soon as the piece on the left is worth $1/n$ to an agent, it shouts "stop".
3. The agent is assigned that piece and removed.
4. Procedure repeats with the remaining agents.



Why PROP? All agents ~~upto~~ $1, \dots, n-1$ get exactly $\frac{1}{n}$ of their valuation of the entire cake.

The last agent hasn't shouted because all the previous $(n-1)$ pieces were $< \frac{1}{n}$ for her, hence the last piece must be $\geq 1 - (n-1)\cdot\frac{1}{n} = \frac{1}{n}$ to her.

Implement this using RW query model.

LEFT = 0

- Ask each agent $\text{cut}_i(\overset{\text{LEFT}}{\cancel{0}}, 1/n) = x_i$, $i \in N$
- Pick the left most x_i, say x'_{i^*}
- Assign $[\overset{\text{LEFT}}{\cancel{0}}, x'_{i^*}]$ to i^* and remove both
- Continue $\text{cut}_i\left(x'_{i^*}, \frac{1}{n}\right) = x_i$, $\forall i \in N \setminus \{i^*\}$

LEFT $= x'_{i^*}$ Repeat steps above (with left most point being the current cut position with the remaining agents)

- Assign the last piece to the final remaining agent.

$O(n^2)$ cut queries and zero eval queries.

This algorithm ~~sever~~ serves as an upper bound of complexity according to RW model.

Is this the most efficient one?

query complexity

$O(n^2)$ — DS Algo (1961)

$O(n \log n)$ — ?

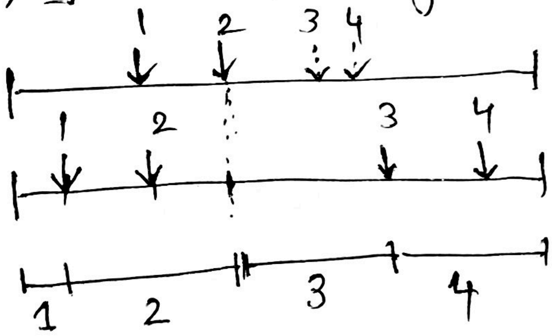
$\Omega(n \log n)$ — Edmonds & Pruhs (2011)

cut and choose

2 queries for ~~n=~~ $n = 2$

Recursive cake-cutting Algorithm (Even-Paz 1984)

- Assume $n = 2^k$ for ($n \geq 2$) ease of exposition.
- Given piece $[x, y]$
- Each player marks z_i s.t. $v_i([x, z_i]) = \frac{1}{2} v_i([x, y])$
- Let z^* be the $n/2$ mark from left.
- Recurse on $[x, z^*]$ with the left $n/2$ players and $[z^*, 1]$ with the right $n/2$ players



Find the worst-case complexity of this algo acc to RW model (exercise)

Why is this algo PROP?

- At stage 0, each agent values the cake 1.
- At each subsequent stage, the players who share a piece $[x, y]$ values it at least $v_i([x, y])/2$.
- Hence, if at stage k each player has value at least $\frac{1}{2^k}$, then at stage $(k+1)$ each player has value at least $\frac{1}{2^{k+1}}$
- The binary tree of division has $\log n$ stages.