CSC2556

Lecture 6

Fair Division 1: Cake-Cutting

[Some illustrations due to: Ariel Procaccia]

Announcements

- Reminder
  - Project proposal due by March 1st by 12:59PM
  - If you want to run your idea by me, this is a good time to approach me.

- Remember to use office hours (drop me an email) if you’re having any difficulty with homework questions.

Fair Division

Cake-Cutting

- A heterogeneous, divisible good
  - Heterogeneous: it may be valued differently by different individuals
  - Divisible: we can share/divide it between individuals
- Represented as $[0,1]$
  - Almost without loss of generality
- Set of players $N = \{1, \dots, n\}$
- Piece of cake $X \subseteq [0,1]$
  - A finite union of disjoint intervals

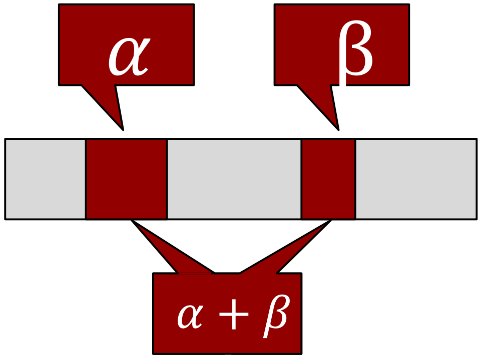
Agent Valuations

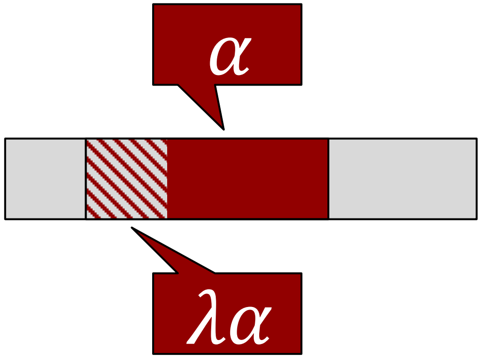
- Each player i has a valuation V_i that is very much like a probability distribution over $[0,1]$

- **Additive:** For $X \cap Y = \emptyset$, $V_i(X) + V_i(Y) = V_i(X \cup Y)$

- **Normalized:** $V_i([0,1]) = 1$

- **Divisible:** $\forall \lambda \in [0,1]$ and X, $\exists Y \subseteq X$ s.t. $V_i(Y) = \lambda V_i(X)$





Fairness Goals

- An allocation is a disjoint partition $A = (A_1, \ldots, A_n)$ of the cake
- We desire the following fairness properties from our allocation A:

- Proportionality (Prop):

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

- Envy-Freeness (EF):

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

Fairness Goals

- Prop: $\forall i \in N: V_i(A_i) \geq 1/n$
- EF: $\forall i, j \in N: V_i(A_i) \geq V_i(A_j)$

- Question: What is the relation between proportionality and EF?
  1. Prop ⇒ EF
  [✓ 2.] EF ⇒ Prop
  3. Equivalent
  4. Incomparable

CUT-AND-CHOOSE

- Algorithm for $n = 2$ players

- Player 1 divides the cake into two pieces X, Y s.t.

$$V_1(X) = V_1(Y) = 1/2$$

- Player 2 chooses the piece she prefers.

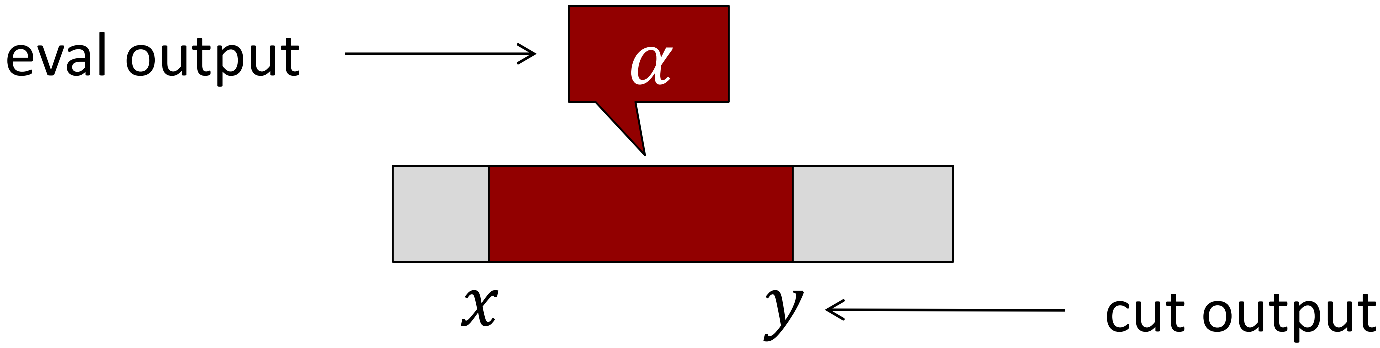
- This is EF and therefore proportional.
  - Why?

Input Model

- How do we measure the “time complexity” of a cake-cutting algorithm for n players?
- Typically, time complexity is a function of the length of input encoded as binary.
- Our input consists of functions V_i, which requires infinite bits to encode.
- We want running time just as a function of n.

Robertson-Webb Model

- We restrict access to valuations V_i's through two types of queries:
  - $\text{Eval}_i(x, y)$ returns $V_i([x, y])$
  - $\text{Cut}_i(x, \alpha)$ returns y such that $V_i([x, y]) = \alpha$



Robertson-Webb Model

- Two types of queries:
  - $\text{Eval}_i(x, y) = V_i([x, y])$
  - $\text{Cut}_i(x, \alpha) = y$ s.t. $V_i([x, y]) = \alpha$

- Question: How many queries are needed to find an EF allocation when $n = 2$?

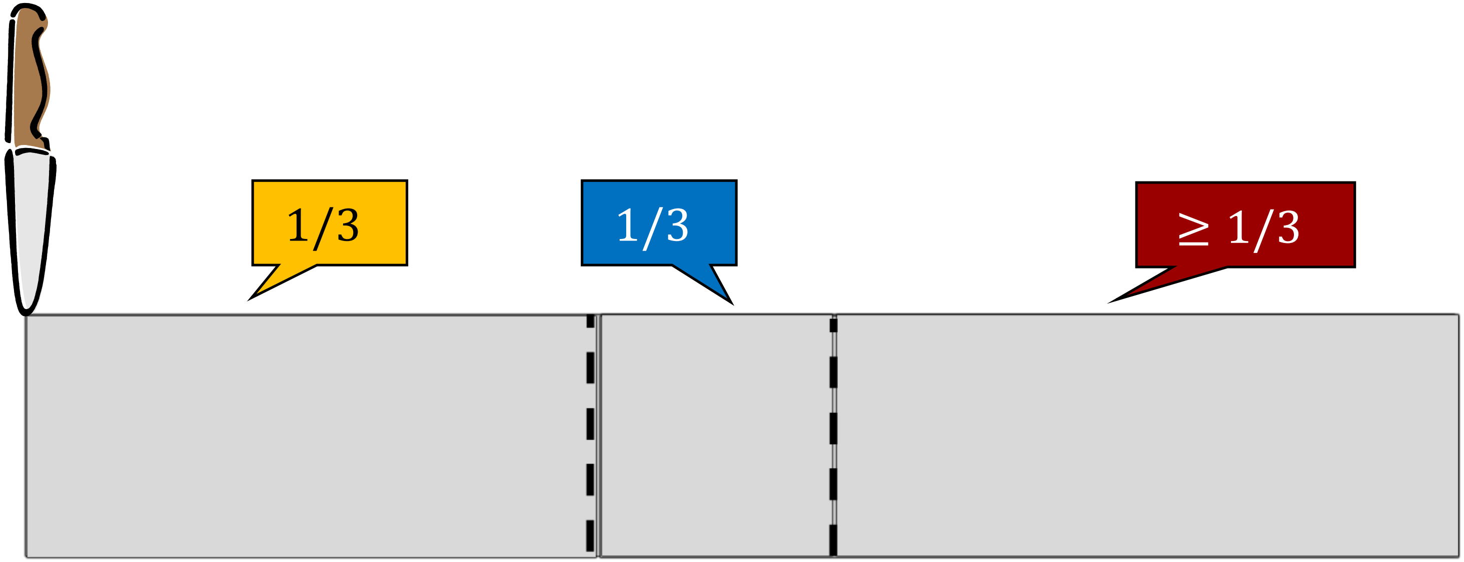
- Answer: 2
  - Why?

DUBINS-SPANIER

- Protocol for finding a proportional allocation for n players

- Referee starts at 0, and continuously moves knife to the right.
- Repeat: when piece to the left of knife is worth $1/n$ to a player, the player shouts "stop", gets the piece, and exits.
- The last player gets the remaining piece.

DUBINS-SPANIER



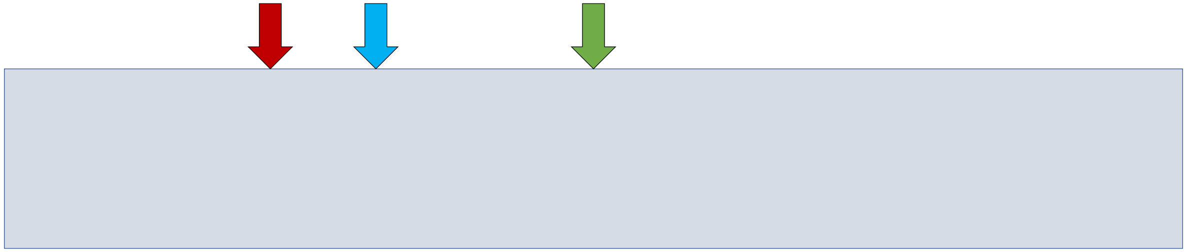
DUBINS-SPANIER

- Moving knife is not really needed.

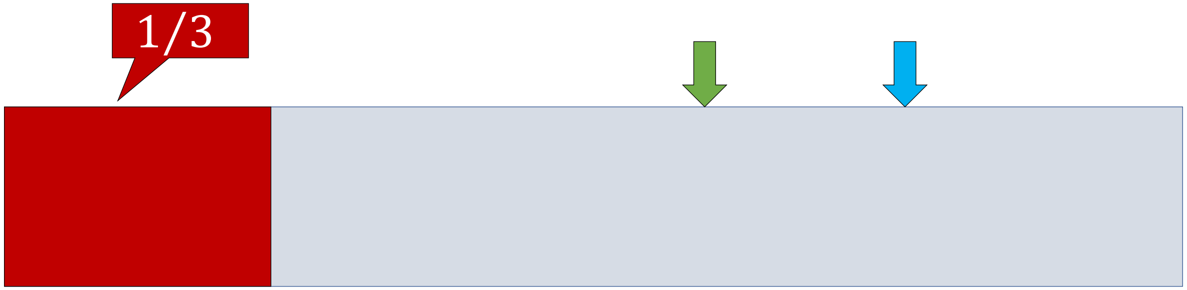
- At each stage, we can ask each remaining player a cut query to mark his $1/n$ point in the remaining cake.

- Move the knife to the leftmost mark.

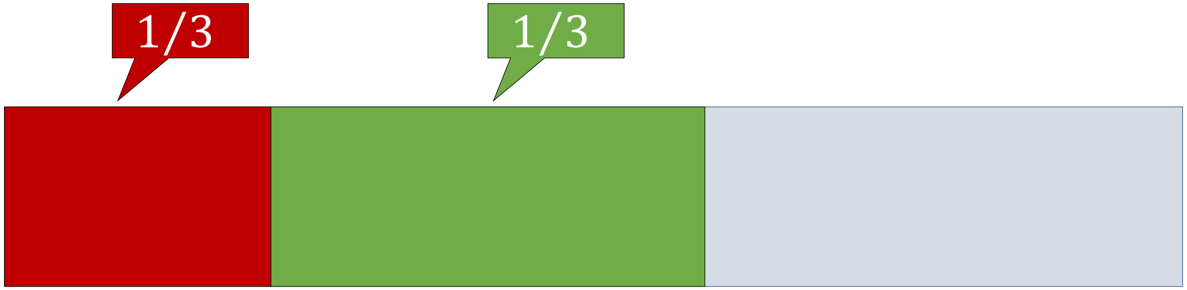
DUBINS-SPANIER



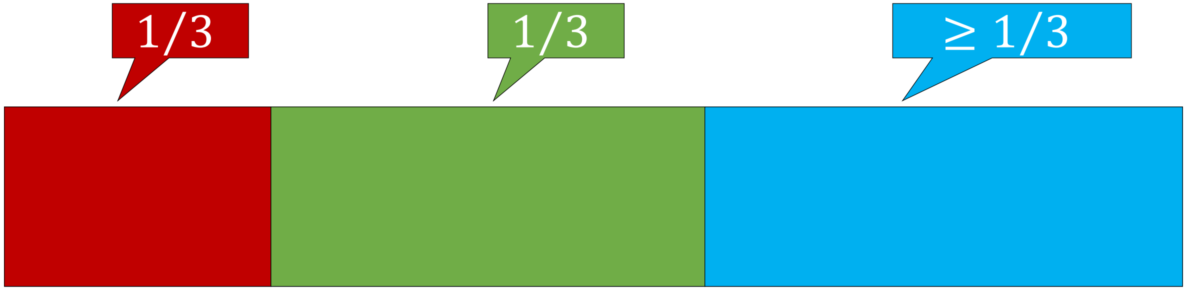
DUBINS-SPANIER



DUBINS-SPANIER



DUBINS-SPANIER



DUBINS-SPANIER

- Question: What is the complexity of the Dubins-Spanier protocol in the Robertson-Webb model?

1. $\Theta(n)$
2. $\Theta(n \log n)$
[✓ 3.] $\Theta(n^2)$
4. $\Theta(n^2 \log n)$

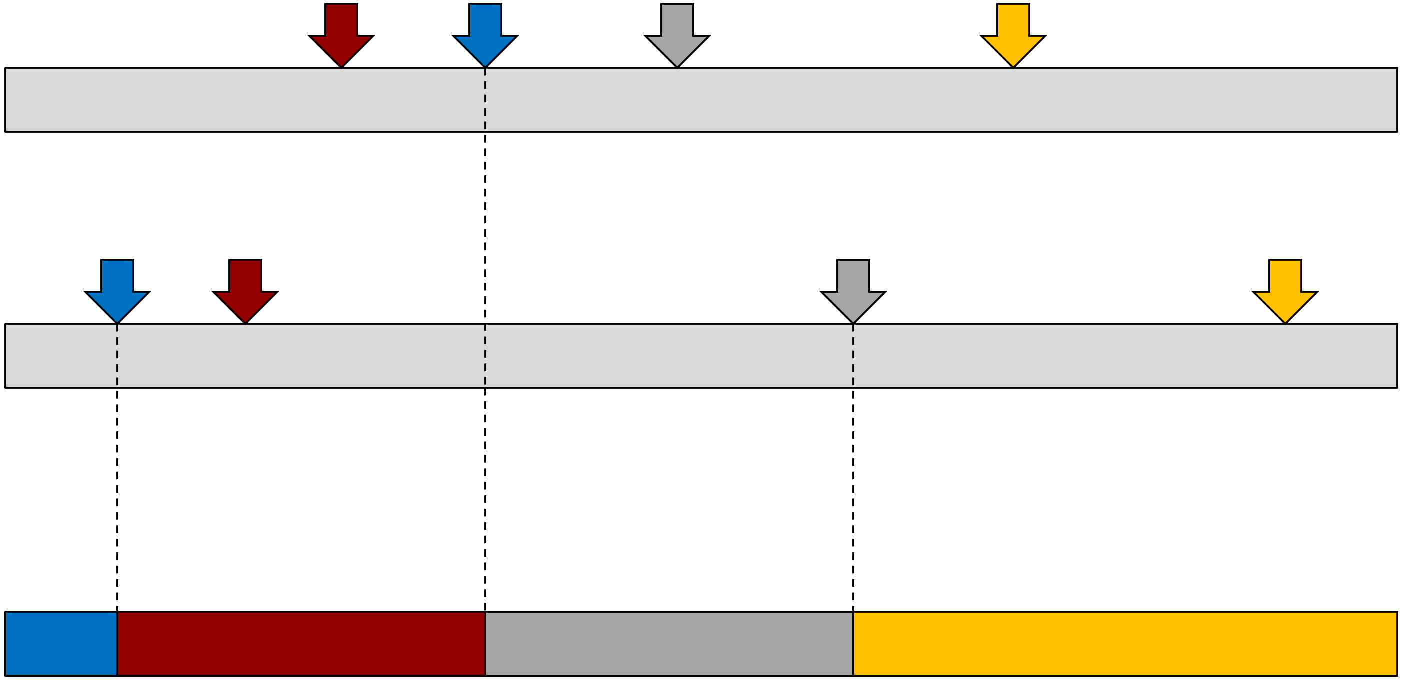
EVEN-PAZ

- Input: Interval $[x, y]$, number of players n
  - Assume $n = 2^k$ for some k
- If $n = 1$, give $[x, y]$ to the single player.
- Otherwise, let each player i mark z_i s.t.

$$V_i([x, z_i]) = \frac{1}{2} V_i([x, y])$$

- Let z^* be the $n/2$ mark from the left.
- Recurse on $[x, z^*]$ with the left $n/2$ players, and on $[z^*, y]$ with the right $n/2$ players.

EVEN-PAZ



EVEN-PAZ

- **Theorem:** EVEN-PAZ returns a Prop allocation.
- **Proof:**
  - Inductive proof. We want to prove that if player i is allocated piece A_i when $[x, y]$ is divided between n players, $V_i(A_i) \geq (1/n)V_i([x, y])$
    - Then Prop follows because initially $V_i([x, y]) = V_i([0,1]) = 1$
  - Base case: $n = 1$ is trivial.
  - Suppose it holds for $n = 2^{k-1}$. We prove for $n = 2^k$.
  - Take the 2^{k-1} left players.
    - Every left player i has $V_i([x, z^*]) \geq (1/2)\, V_i([x, y])$
    - If it gets A_i, by induction, $V_i(A_i) \geq \frac{1}{2^{k-1}}\, V_i([x, z^*]) \geq \frac{1}{2^k}\, V_i([x, y])$

EVEN-PAZ

- Question: What is the complexity of the Even-Paz protocol in the Robertson-Webb model?

1. $\Theta(n)$
[✓ 2.] $\Theta(n \log n)$
3. $\Theta(n^2)$
4. $\Theta(n^2 \log n)$

Complexity of Proportionality

- Theorem [Edmonds and Pruhs, 2006]: Any proportional protocol needs $\Omega(n \log n)$ operations in the Robertson-Webb model.

- Thus, the EVEN-PAZ protocol is (asymptotically) provably optimal!

Envy-Freeness?

- "I suppose you are also going to give such cute algorithms for finding envy-free allocations?"

- Bad luck. For n-player EF cake-cutting:
  - [Brams and Taylor, 1995] give an unbounded EF protocol.
  - [Procaccia 2009] shows $\Omega(n^2)$ lower bound for EF.
  - Last year, the long-standing major open question of "bounded EF protocol" was resolved!
  - [Aziz and Mackenzie, 2016]: $O(n^{n^{n^{n^{n^{n}}}}})$ protocol!
    - Not a typo!

Other Desiderata

- There are two more properties that we often desire from an allocation.

- Pareto optimality (PO)
  - Notion of efficiency
  - Informally, it says that there should be no “obviously better” allocation

- Strategyproofness (SP)
  - No player should be able to gain by misreporting her valuation

Strategyproofness (SP)

- For deterministic mechanisms
  - "Strategyproof": No player should be able to increase her *utility* by misreporting her valuation, irrespective of what other players report.
- For randomized mechanisms
  - "Strategyproof-in-expectation": No player should be able to increase her *expected utility* by misreporting.
  - For simplicity, we'll call this strategyproofness, and assume we mean "in expectation" if the mechanism is randomized.

Strategyproofness (SP)

- Deterministic
  - Bad news!
  - Theorem [Menon & Larson ‘17] : No deterministic SP mechanism is (even approximately) proportional.
- Randomized
  - Good news!
  - Theorem [Chen et al. ‘13, Mossel & Tamuz ‘10]: There is a randomized SP mechanism that always returns an envy-free allocation.

Perfect Partition

- Theorem [Lyapunov '40]:
  - There always exists a "perfect partition" $(B_1, \dots, B_n)$ of the cake such that $V_i\left(B_j\right) = {}^1/_n$ for every $i, j \in [n]$.
  - Every agent values every bundle equally.

- Theorem [Alon '87]:
  - There exists a perfect partition that only cuts the cake at $poly(n)$ points.
  - In contrast, Lyapunov's proof is non-constructive, and might need an unbounded number of cuts.

Perfect Partition

- Q: Can you use an algorithm for computing a perfect partition as a black-box to design a randomized SP+EF mechanism?
  - Yes! Compute a perfect partition, and assign the n bundles to the n players uniformly at random.
  - Why is this EF?
    - Every agent values every bundle at $1/n$.
  - Why is this SP-in-expectation?
    - Because an agent is assigned a random bundle, her expected utility is $1/n$, irrespective of what she reports.