Graduate AI

Lecture 24: Fair division

April 18, 2012

Teachers:
Martial Hebert
Ariel Procaccia (this time)

Cake cutting

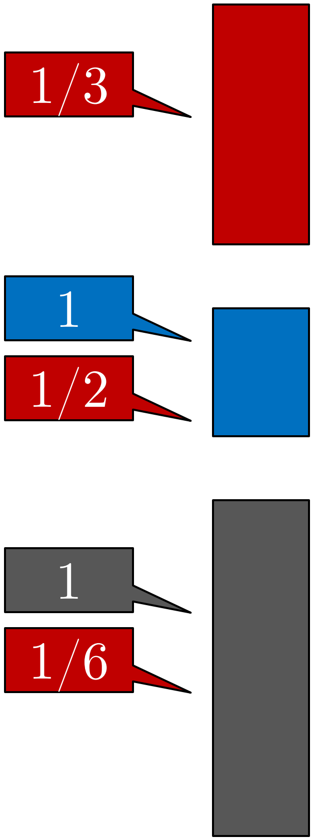
- A cake must be divided between several children
- The cake is heterogeneous
- Each child has different value for same piece of cake
- How can we divide the cake fairly?
- What is “fairly”?
- A metaphor for land disputes, time using shared resources, etc.

The model

- Cake is interval [0,1]
- Set of *agents/players* {1,...,n}
- Piece of cake $X \subseteq [0,1]$: finite union of disjoint intervals
- Each agent has valuation v_i over pieces of cake
  - Additive, value of whole cake is 1
  - Think probability measure
- Find allocation $X_1, \ldots, X_n$
  - Not necessarily connected pieces

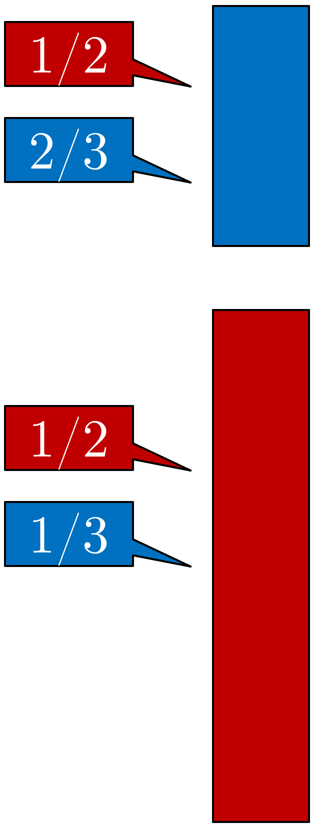
Fairness properties

- *Proportionality*: $\forall i,\ v_i(X_i) \geq 1/n$
- *Envy-Freeness*: $\forall i,j,\ v_i(X_i) \geq v_i(X_j)$
- For $n = 2$ which is stronger?
  - Envy-freeness $\Leftrightarrow$ proportionality
- For $n \geq 3$ which is stronger?
  - Envy-freeness $\Rightarrow$ proportionality
  - Proportionality does not imply EF



Cut-and-Choose

- Algorithm for n=2
- Agent 1 divides into two pieces X,Y s.t. $v_1(X)=1/2$, $v_1(Y)=1/2$
- Agent 2 chooses preferred piece
- This is EF (hence proportional)



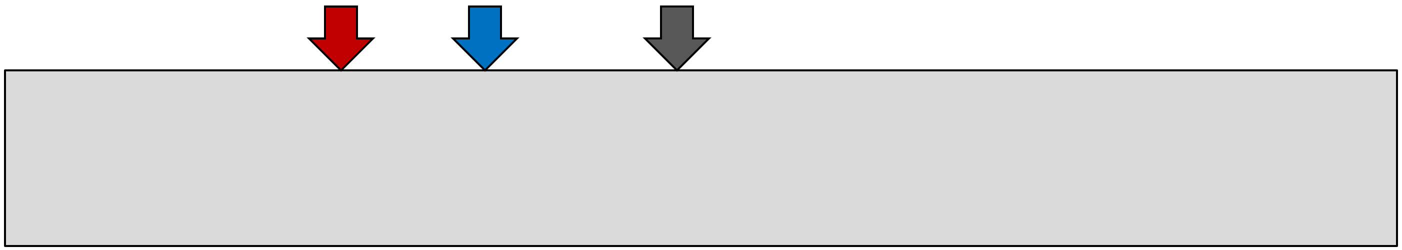
Dubins-Spanier

- Referee continuously moves knife
- Repeat: when piece left of knife is worth 1/n to agent, agent shouts "stop" and gets piece
- That agent is removed
- Last agent gets remaining piece
- Protocol is proportional

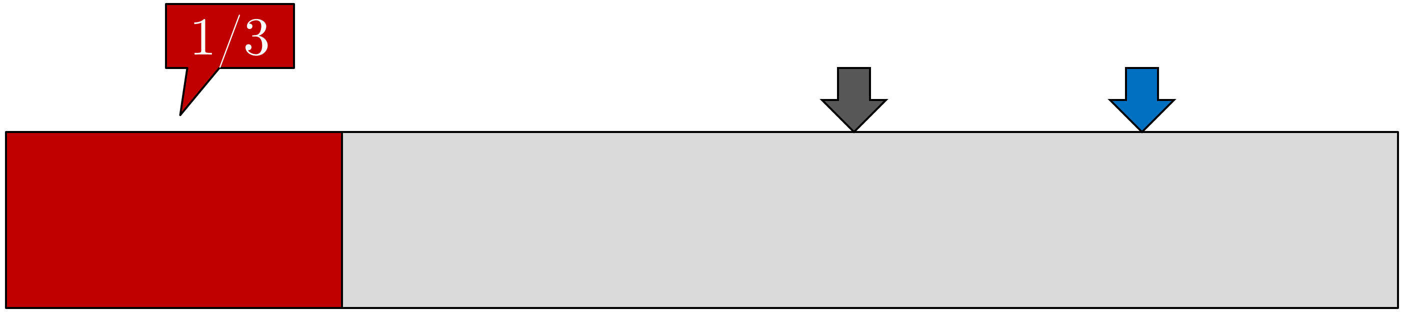
Discrete Dubins-Spanier

- Moving knife is not really needed
- Repeat: each agent makes a mark at his 1/n point, leftmost agent gets piece up to its mark

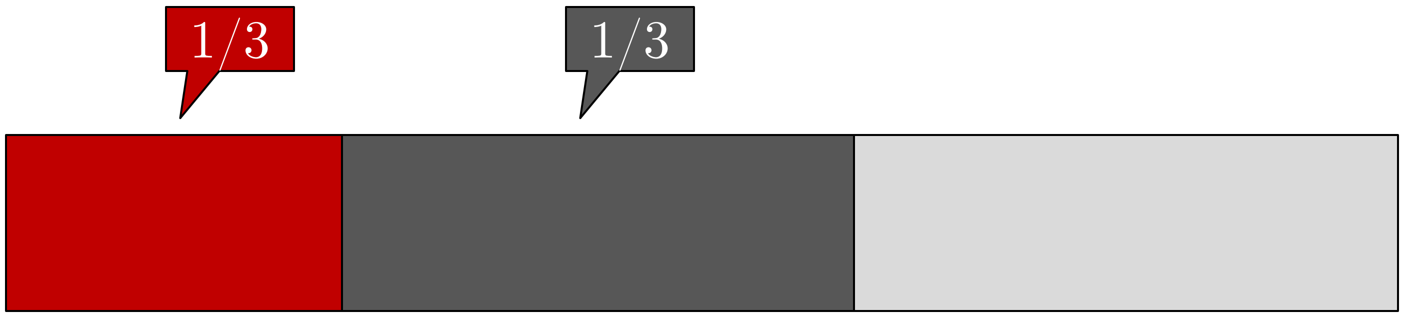
Example



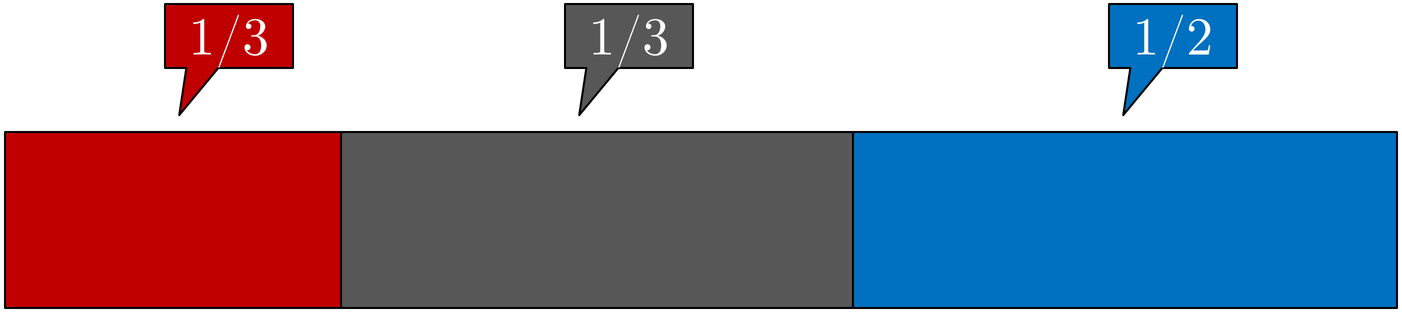
Example



Example



Example



Selfridge-Conway

- **Stage 0**
  - Agent 1 divides the cake into three equal pieces according to v_1
  - Agent 2 trims the largest piece s.t. there is a tie between the two largest pieces according to v_2
  - Cake 1 = cake w/o trimmings, Cake 2 = trimmings
- **Stage 1**
  - Agent 3 chooses one of the three pieces of Cake 1
  - If agent 3 did not choose the trimmed piece, agent 2 is allocated the trimmed piece
  - Otherwise, agent 2 chooses one of the two remaining pieces
  - Agent 1 gets the remaining piece
  - Denote the agent $i \in \{2, 3\}$ that received the trimmed piece by T, and the other by T'
- **Stage 2**
  - T' divides Cake 2 into three equal pieces according to $v_{T'}$
  - Agents T, 1, and T' choose the pieces of Cake 2, in that order

The Robertson-Webb model

- A concrete complexity model
- Two types of queries
  - $\text{Eval}_i(x,y) = v_i([x,y])$
  - $\text{Cut}_i(x,\alpha) = y \text{ s.t. } v_i([x,y]) = \alpha$
- Can simulate all known discrete protocols

Bounds in RW model

- Proportional
  - Recursive protocol that requires O(nlogn) queries [Even and Paz, 1984]
  - Lower bound of $\Omega(n\log n)$ [Edmonds and Pruhs, SODA 2006]
- Envy free (always exists)
  - $n = 2$: Cut and Choose
  - $n = 3$: "good" protocol [Selfridge and Conway]
  - $n \geq 4$: known protocol requires unbounded number of queries
  - Lower bound of $\Omega(n^2)$ [P, IJCAI 2009], unbounded with contiguous pieces [Stromquist, 2009]

Strategyproofness

- We discussed *strategyproofness (SP)* in social choice and auctions
- All the cake cutting algorithms that we discussed are not SP: agents can gain from manipulation
    - Cut and choose: player 1 can manipulate
    - Dubins Spanier: shout later
- Assumption: agents report their full valuation functions (which are typically assumed to be concisely representable)
- Deterministic EF and SP algs exist in some cases, but they are rather involved [Chen et al., AAAI 2010]

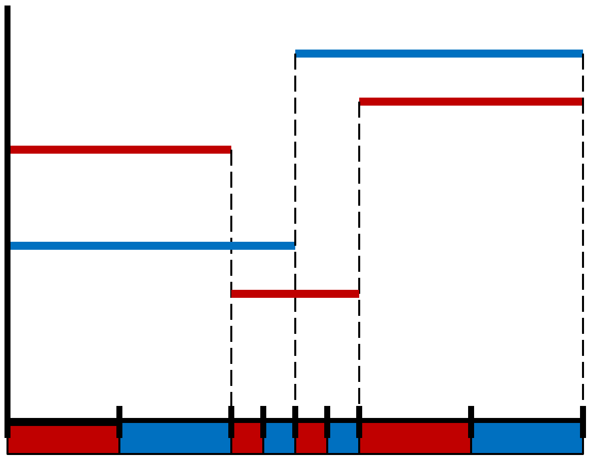
A randomized algorithm

- $X_1,\ldots,X_n$ is a *perfect partition* if $v_i(X_j)=1/n$ for all i,j
- Algorithm
  - Compute a perfect partition
  - Draw a permutation π over $\{1,\ldots,n\}$
  - Allocate to agent i the piece $X_{\pi(i)}$
- **Theorem** [Chen et al., AAAI 2010; Mossel&Tamuz, SAGT 2010]: the algorithm is SP in expectation and always produces an EF allocation
- **Proof:** if an agent lies the algorithm may compute a different partition, but for any partition:

$$\sum_{j\in N}\frac{1}{n}\cdot V_i(X'_j)=\frac{1}{n}\left(\sum_{j\in N}V_i(X'_j)\right)=\frac{1}{n}$$

Computing a perfect partition

- **Theorem** [Alon, 1986]: a perfect partition always exists, needs polynomially many cuts
- Proof is nonconstructive
- Can find perfect partitions for special valuation functions



Applications of fair division

- Setting: allocating multiple homogeneous resources to agents with different requirements
- Running example: cloud computing
- State-of-the-art systems employ a single resource abstraction
- Assumption: agents have proportional demands for their resources
- Example:
  - Agent has requirement (2 CPU,1 RAM) for each copy of task
  - Indifferent between allocations (4,2) and (5,2)

Dominant resource fairness

- *Dominant resource* of an agent = resource that requires highest fraction of total
- *Dominent share* = fraction of dominant resource
- **Dominant Resource Fairness (DRF)** [Ghodsi et al, NSDI 2011]: allocate max number of tasks s.t. dominant shares are equalized

DRF EXAMPLE

- System has 9 CPU, 18 RAM
- Agent 1 task needs (1 CPU, 4 RAM)
- Agent 2 task needs (3,1)
- y and z = number of tasks allocated to agents 1 and 2, resp.
- y+3z CPU and 4y+z RAM are allocated
- max (y,z) s.t. $y+3z \leq 9$, $4y+z \leq 18$, $2y/9=z/3$
- Solution: y=3, z=2 $\Rightarrow$ (3,12) to agent 1, (6,2) to agent 2

Properties of DRF

- **Theorem** [Ghodsi et al., NSDI 2011]: DRF is "proportional", envy free, and strategyproof (and Pareto optimal)