CSC304
Algorithmic Game Theory & Mechanism Design

Evi Micha

Cake-Cutting

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

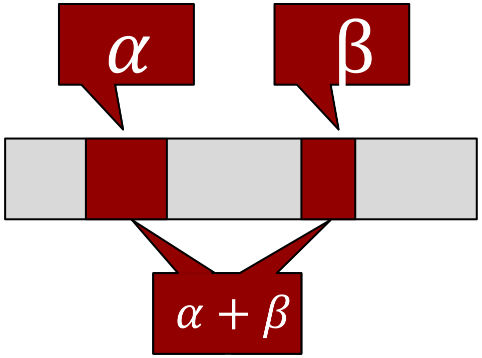
Agent Valuations

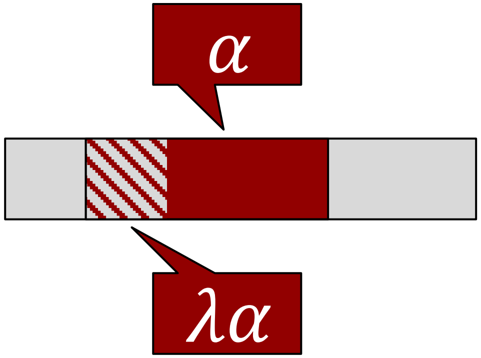
- Each player i has a valuation V_i that is very much like a probability distribution over $[0,1]$

- 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

- Allocation: disjoint partition $A = (A_1, \dots, A_n)$
  - A_i = piece of the cake given to player i

- Desired fairness properties:
  - Proportionality (Prop):

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

  - Envy-Freeness (EF):

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

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 $\Rightarrow$ EF
  2. EF $\Rightarrow$ 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 envy-free and therefore proportional.
  - Why?

Input Model

- How do we measure the "time complexity" of a cake-cutting algorithm for n players?

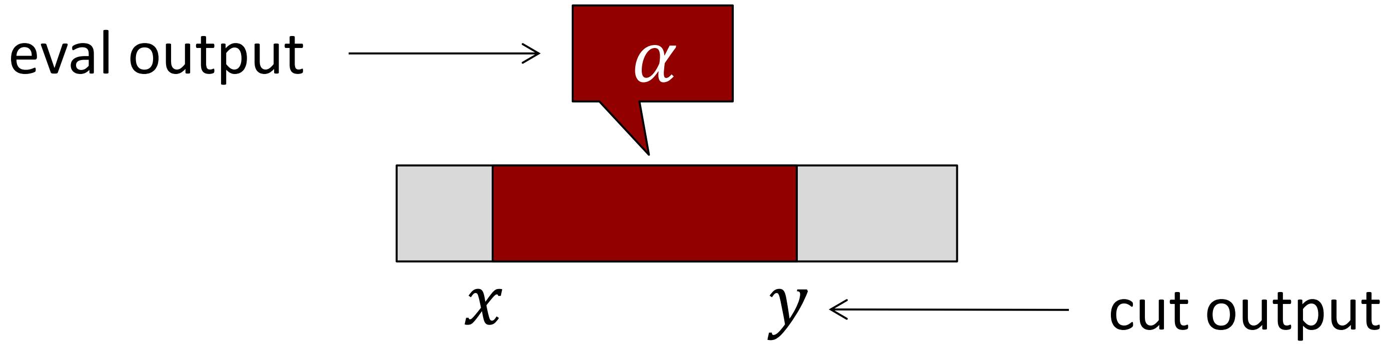
- Typically, time complexity is a function of the length of input encoded as binary.

- Our input consists of functions V_i, which require infinite bits to encode.

- We want running time as a function of n.

Robertson-Webb Model

- We restrict access to valuation V_i through two types of queries:
  - $\text{Eval}_i(x, y)$ returns $\alpha = V_i([x, y])$
  - $\text{Cut}_i(x, \alpha)$ returns any y such that $V_i([x, y]) = \alpha$
    - If $V_i([x, 1]) < \alpha$, return 1.



Robertson-Webb Model

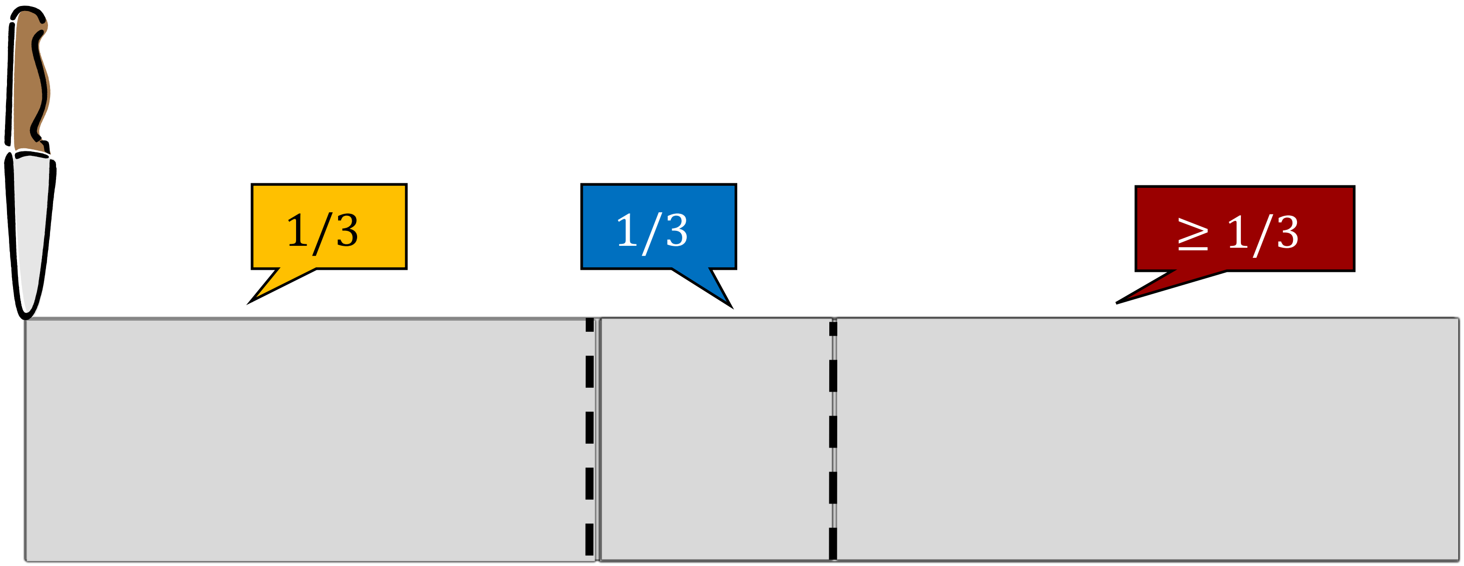
- Two types of queries:
  - $\text{Eval}_i(x, y) = V_i([x, y])$
  - $\text{Cut}_i(x, \alpha) = y$ s.t. $V_i([x, y]) = \alpha$
- Question: How many queries are needed to find an EF allocation when $n = 2$?
- Answer: 2

Dubins-Spanier

- Protocol for finding a proportional allocation for n players

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

DUBINS-SPANIER

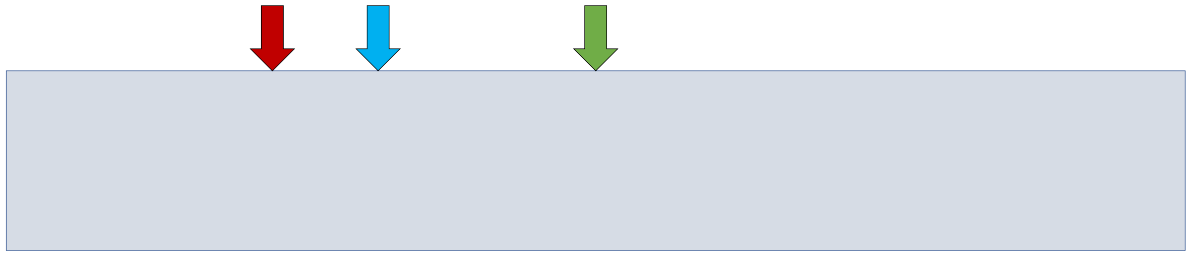


Dubins-Spanier

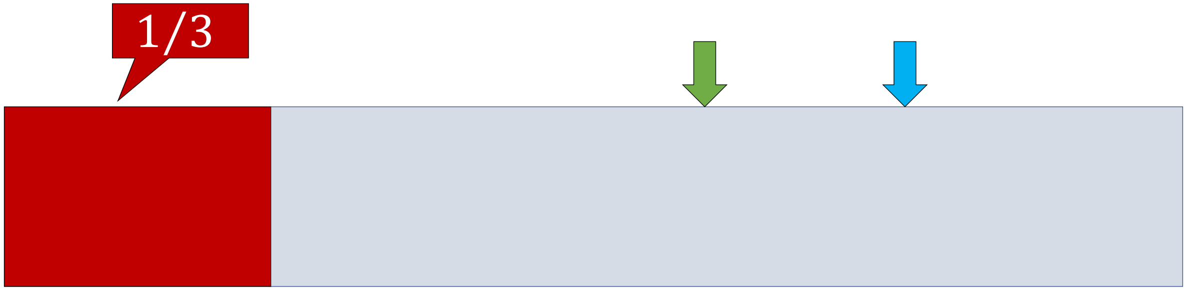
- Robertson-Webb model? Cut-Eval queries?
  - Moving knife is not really needed.

- At each stage, we want to find the remaining player that has value $1/n$ from the smallest next piece.
  - Ask each remaining player a cut query to mark a point where her value is $1/n$ from the current point.
  - Directly move the knife to the leftmost mark, and give that piece to that player.

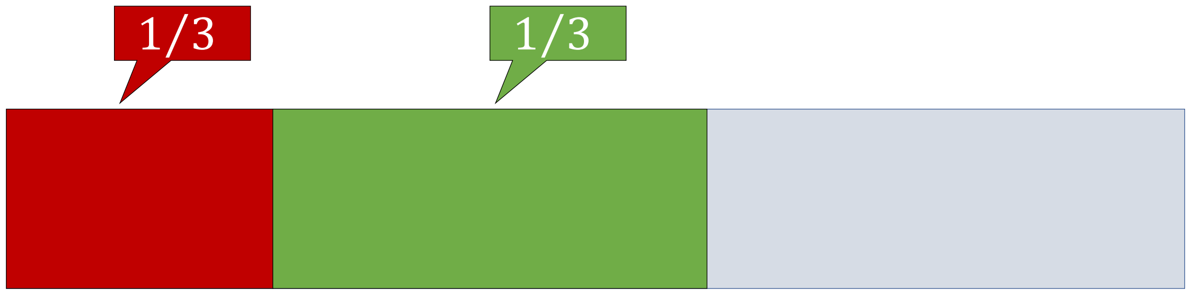
Visual Proof of Proportionality



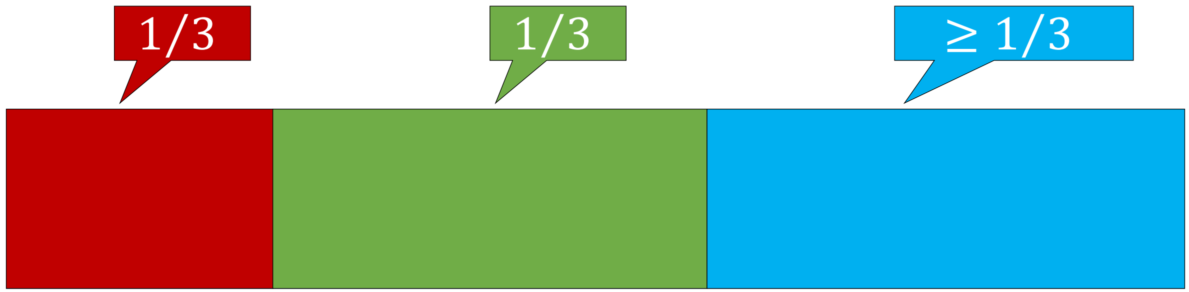
Visual Proof of Proportionality



Visual Proof of Proportionality



Visual Proof of Proportionality



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. $\Theta(n^2 \log n)$

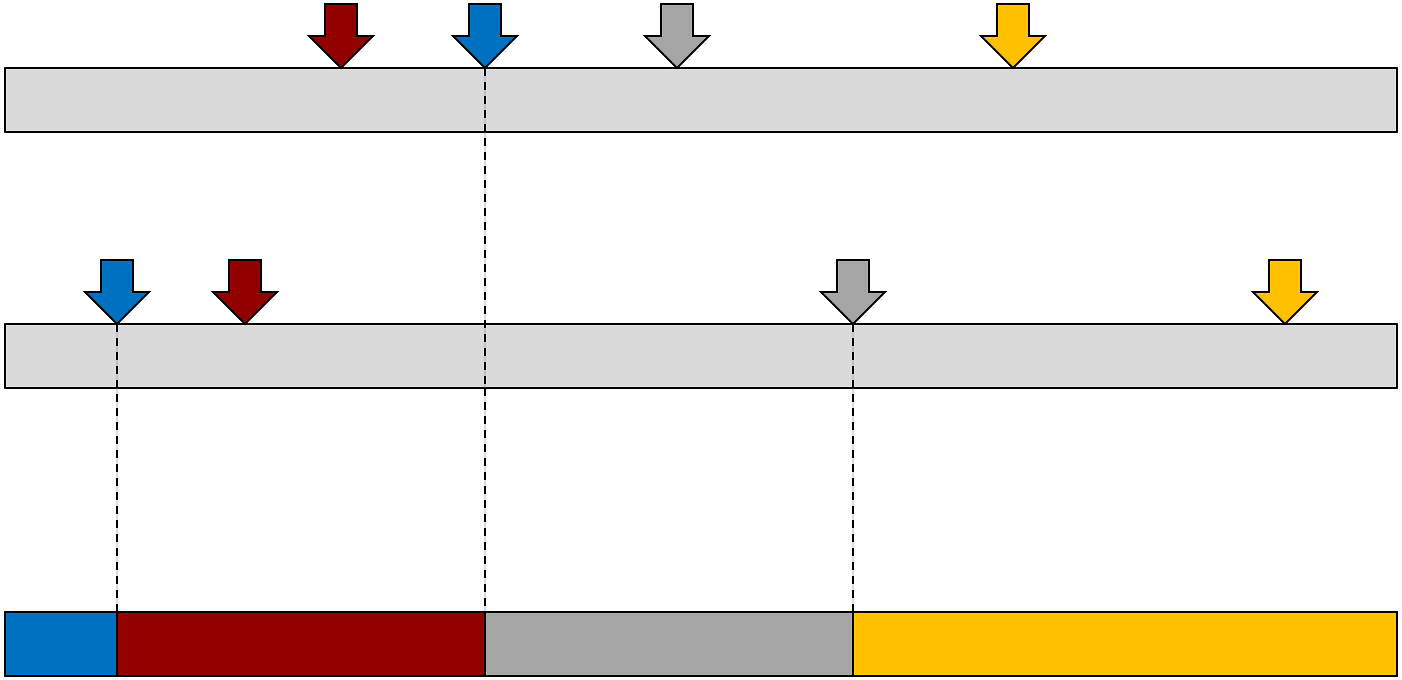
Even-Paz (Recursive)

- Input: Interval $[x, y]$, number of players n
  - For simplicity, 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 mark $n/2$ 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.
- Inductive Proof:
  - Hypothesis: With n players, EVEN-PAZ ensures that for each player i, $V_i(A_i) \geq (1/n) \cdot V_i([x, y])$
    - 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

- Theorem: EVEN-PAZ uses $O(n \log n)$ queries.
- Simple Proof:
  - Protocol runs for $\log n$ rounds.
  - In each round, each player is asked one cut query.
  - QED!

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!
    - Yes, it's not a typo!

Four More Desiderata

- Equitability
  - $V_i(A_i) = V_j(A_j)$ for all i, j.

- Perfect Partition
  - $V_i(A_k) = 1/n$ for all i, k.
  - Implies equitability.
  - Guaranteed to exist [Lyapunov '40] and can be found using only poly(n) cuts [Alon '87].

Four More Desiderata

- Pareto Optimality
  - We say that A is Pareto optimal if for any other allocation B, it cannot be that $V_i(B_i) \geq V_i(A_i)$ for all i and $V_i(B_i) > V_i(A_i)$ for some i.

- Strategyproofness
  - No agent can misreport her valuation and increase her (expected) value for her allocation.

Strategyproofness

- 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.

Strategyproofness

- Randomized SP Mechanism:
  - Compute a perfect partition, and assign the n bundles to the n players uniformly at random.
- Why is this EF?
  - Every agent has value $1/n$ for her own as well as for every other agent's allocation.
  - Note: We want EF in every realized allocation, not only in expectation.
- Why is this SP?
  - An agent is assigned a random bundle, so her expected utility is $1/n$, irrespective of what she reports.

Pareto Optimality (PO)

- Definition: We say that A is Pareto optimal if for any other allocation B, it cannot be that $V_i(B_i) \geq V_i(A_i)$ for all i and $V_i(B_i) > V_i(A_i)$ for some i.

- Q: Is it PO to give the entire cake to player 1?
- A: Not necessarily. But yes if player 1 values "every part of the cake positively".

PO + EF

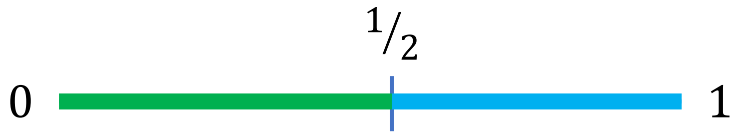
- Theorem [Weller '85]:
  - There always exists an allocation of the cake that is both envy-free and Pareto optimal.

- One way to achieve PO+EF:
  - Nash-optimal allocation: $\text{argmax}_A \prod_{i \in N} V_i(A_i)$
  - Obviously, this is PO. The fact that it is EF is non-trivial.
  - This is named after John Nash.
    - Nash social welfare = product of utilities
    - Different from utilitarian social welfare = sum of utilities

Nash-Optimal Allocation

- Example:
  - Green player has value 1 distributed evenly over $[0, {}^2/_3]$
  - Blue player has value 1 distributed evenly over $[0,1]$
  - Without loss of generality (why?) suppose:
    - Green player gets $[0, x]$ for $x \leq {}^2/_3$
    - Blue player gets $[x, {}^2/_3] \cup [{}^2/_3, 1] = [x, 1]$
  - Green's utility = $\frac{x}{{}^2/_3}$, blue's utility = $1 - x$
  - Maximize: $\frac{3}{2} x \cdot (1 - x) \Rightarrow x = {}^1/_2$

Allocation



Green has utility $\frac{3}{4}$

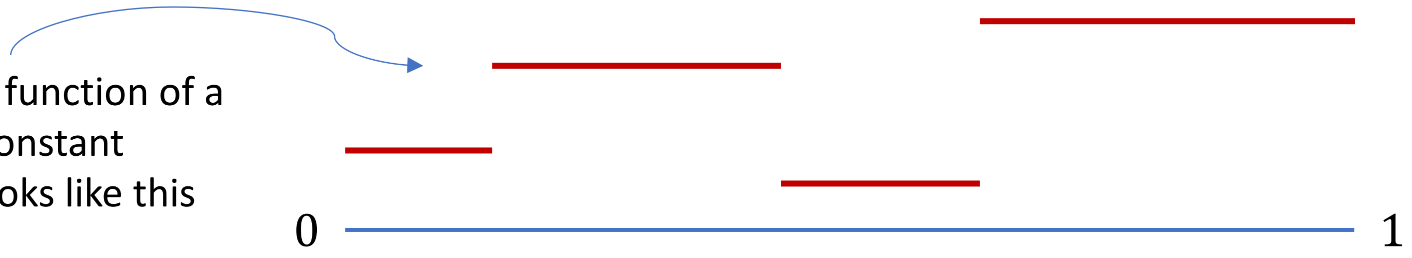
Blue has utility $\frac{1}{2}$

Problem

- Difficult to compute in general
  - I believe it should require an unbounded number of queries in the Robertson-Webb model. But I can't find such a result in the literature.

- Theorem [Aziz & Ye '14]:
  - For *piecewise constant* valuations, the Nash-optimal solution can be computed in polynomial time.

The density function of a piecewise constant valuation looks like this

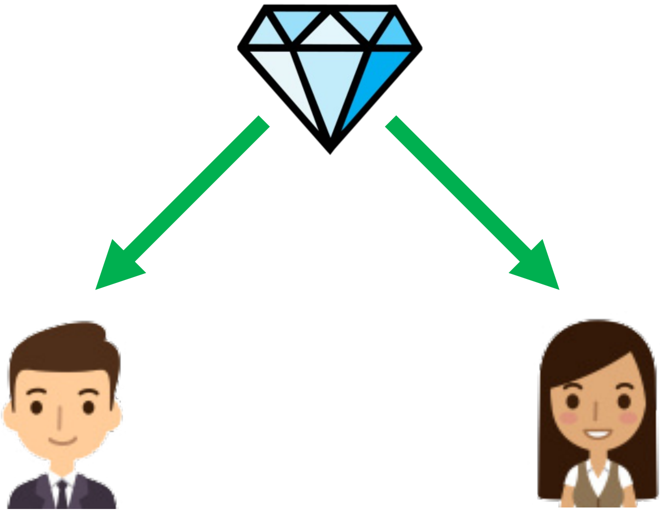


NOT IN SYLLABUS

Indivisible Goods
(Only if time permits)

Indivisible Goods

- Goods cannot be shared / divided among players
  - E.g., house, painting, car, jewelry, …
- Problem: Envy-free allocations may not exist!



Indivisible Goods: Setting

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

Given such a matrix of numbers, assign each good to a player.

We assume additive values. So, e.g., V$(\{$$,$$\}) = 8 + 7 = 15$

Indivisible Goods

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

Indivisible Goods

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

Indivisible Goods

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

Indivisible Goods

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

Indivisible Goods

- Envy-freeness up to one good (EF1):

$$\forall i, j \in N, \exists g \in A_j : V_i(A_i) \geq V_i\left(A_j \setminus \{g\}\right)$$

  - Technically, $\exists g \in A_j$ only applied if $A_j \neq \emptyset$.
  - "If i envies j, there must be some good in j's bundle such that removing it would make i envy-free of j."

- Does there always exist an EF1 allocation?

EF1

- Yes! We can use Round Robin.
  - Agents take turns in a cyclic order, say $1,2, \ldots, n, 1,2, \ldots, n, \ldots$
  - An agent, in her turn, picks the good that she likes the most among the goods still not picked by anyone.
  - [Assignment Problem] This yields an EF1 allocation regardless of how you order the agents.

- Sadly, the allocation returned may not be Pareto optimal.

EF1+PO?

- Nash welfare to the rescue!

- Theorem [Caragiannis et al. '16]:
  - Maximizing Nash welfare $\prod_i V_i(A_i)$ achieves both EF1 and PO.
  - A bit of subtlety required if the maximum Nash welfare is zero

Integral Nash Allocation

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

20 * 8 * (9+10) = 3040

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

$(8+7) * 8 * 18 = 2160$

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

$8 * (12+8) * 10 = 1600$

				
	8	7	20	5
	9	11	12	8
	9	10	18	3

20 * (11+8) * 9 = 3420

				
	8	7	**20**	5
	9	**11**	12	**8**
	**9**	10	18	3

Computation

- For indivisible goods, Nash-optimal solution is strongly NP-hard to compute
  - That is, remains NP-hard even if all values are bounded.

- Open Question: Can we find an allocation that is both EF1 and PO in polynomial time?
  - A recent paper provides a pseudo-polynomial time algorithm, i.e., its time is polynomial in n, m, and $\max_{i,g} V_i(\{g\})$.

Stronger Fairness Guarantees

- Envy-freeness up to the least valued good (EFx):
  - $\forall i, j \in N, \forall g \in A_j : V_i(A_i) \geq V_i(A_j \backslash \{g\})$
  - "If i envies j, then removing *any* good from j's bundle eliminates the envy."
  - Open question: Is there always an EFx allocation?

- Contrast this with EF1:
  - $\forall i, j \in N, \exists g \in A_j : V_i(A_i) \geq V_i(A_j \backslash \{g\})$
  - "If i envies j, then removing *some* good from j's bundle eliminates the envy."
  - We know there is always an EF1 allocation that is also PO.

Stronger Fairness

- To clarify the difference between EF1 and EFx:
  - Suppose there are two players and three goods with values as follows.

	A	B	C
P1	5	1	10
P2	0	1	10

  - If you give {A} → P1 and {B,C} → P2, it's EF1 but not EFx.
    - EF1 because if P1 removes C from P2's bundle, all is fine.
    - Not EFx because removing B doesn't eliminate envy.
  - Instead, {A,B} → P1 and {C} → P2 would be EFx.