Lecture 12

Applications to Algorithmic Game Theory & Computational Social Choice

A little bit of game theory

Recap: Yao's Minimax Principle

- Let R and D denote randomized and deterministic algorithms.
- Let F denote a distribution over instances.
- Let $C(\cdot,\cdot)$ denote the running time of an algorithm on an input.

- Then Yao's principle says that:

$$\min_R \max_I E_R[C(R, I)] = \max_F \min_D E_{I\sim F}\,[D, I]$$

von-Neumann's Minimax Theorem

- Generalizes Yao's minimax principle
- A game between two players
  - There is a matrix A.
  - The row player (R) selects a row, and the column player (C) selects a column.
  - The result is the value of the chosen cell.
  - R wants to maximize the value, while C wants to minimize the value.
  - Each player's strategy can be a distribution over actions.
  - In a Nash equilibrium, each player is playing an optimal strategy given the strategy of the other player.

von-Neumann's Minimax Theorem

- Let x_R and x_C denote strategies of the players.
- Let v denote the final value.
- R can guarantee: $V \geq V_R = \max_{x_1} \min_{x_2} A(x_1, x_2)$.
- C can guarantee: $V \leq V_C = \min_{x_2} \max_{x_1} A(x_1, x_2)$.
- Then $V_R \leq V_C$ (WHY?)
  - Thus, we have that:

    $$\max_{x_1} \min_{x_2} A(x_1, x_2) \leq \min_{x_2} \max_{x_1} A(x_1, x_2)$$

  - Can it be that $V_R < V_C$?

von-Neumann's Minimax Theorem

- Theorem [von Neumann]:
  $\max_{x_1} \min_{x_2} A(x_1, x_2) = \min_{x_2} \max_{x_1} A(x_1, x_2)$
- Interpretations
  - "It does not matter which player goes first."
  - "Playing my safe strategy is optimal if the other player is also playing his safe strategy."
  - Just a statement that holds for any matrix A
- Yao's principle:
  - Columns = deterministic algorithms
  - Rows = problem instances
  - Cell values = running times

Minimax via Regret Learning

- We want to show:

$$V_R = \max_{x_1} \min_{x_2} A(x_1, x_2)$$
$$V_C = \min_{x_2} \max_{x_1} A(x_1, x_2)$$
$$V_R = V_C$$

- We know about Randomized Weighted Majority:

$$M^{(T)} \leq (1+\eta) \cdot m_i^{(T)} + 2 \cdot (\log n / \eta)$$

  - Setting $\eta = \sqrt{\log n / T}$

$$M^{(T)} \leq m_i^{(T)} + 2\sqrt{T \cdot \log n}$$

Minimax via Regret Learning

- Scale the values so that they're in $[0,1]$.

- Suppose for contradiction $V_R = V_C - \delta, \delta > 0$.
  - If C commits first, there is a row guaranteeing $V \geq V_C$.
  - If R commits first, there is a column guaranteeing $V \leq V_R$.
  - WHY?

- Suppose R uses RWM, and C responds optimally to the current mixed strategy.

Minimax via Regret Learning

- After T iterations:
  - $V \geq$ best row in hindsight $- 2\sqrt{T \cdot \log n}$
  - Best row in hindsight $\geq T \cdot V_C$
  - $V \leq T \cdot V_R$
- Thus: $T \cdot V_R \geq T \cdot V_C - 2\sqrt{T \cdot \log n}$
- $\delta T \leq 2\sqrt{T \cdot \log n}$, which false for large enough T.
- QED!

A little bit of fair division

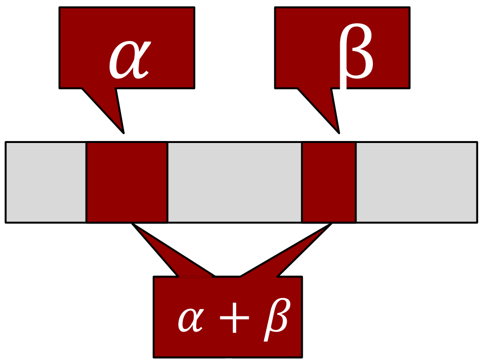
Cake-Cutting

- 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

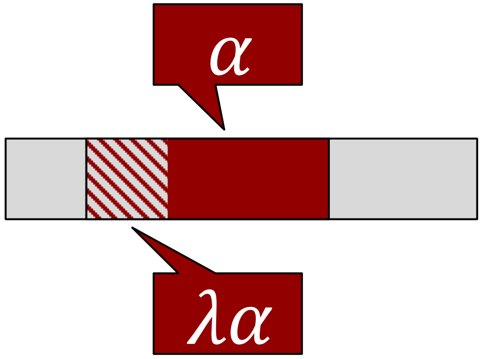
- Each player i has a valuation V_i that is very much like a probability distribution over $[0,1]$

- **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 $A = (A_1, \dots, A_n)$ of the cake
- We desire the following fairness properties from our allocation A:
- 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 EF and therefore proportional.
  - Why?

Input Model

- How do we measure the “complexity” of a cake-cutting protocol for n players?

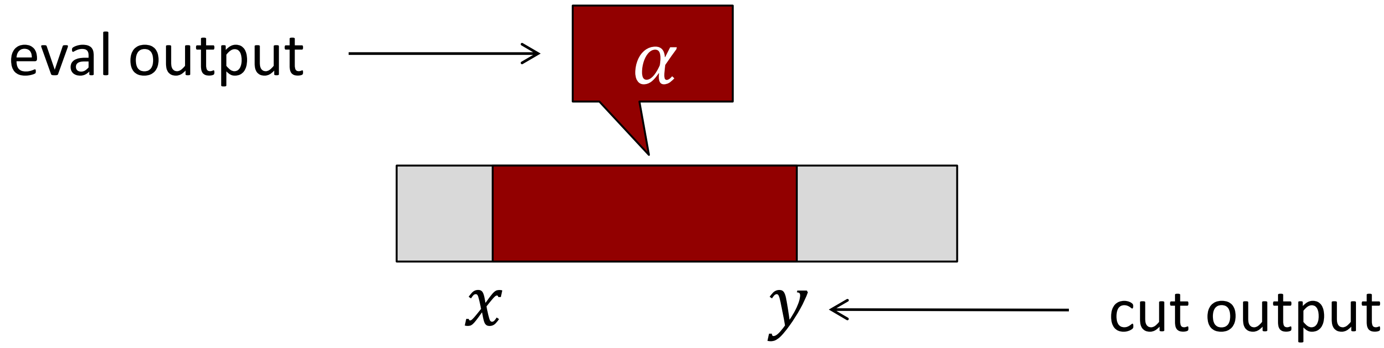
- Typically, running time is a function of the length of input encoded in binary.

- Our input consists of functions V_i, which need infinite bits of encoding.

- We need an oracle model.

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 $n = 2$?

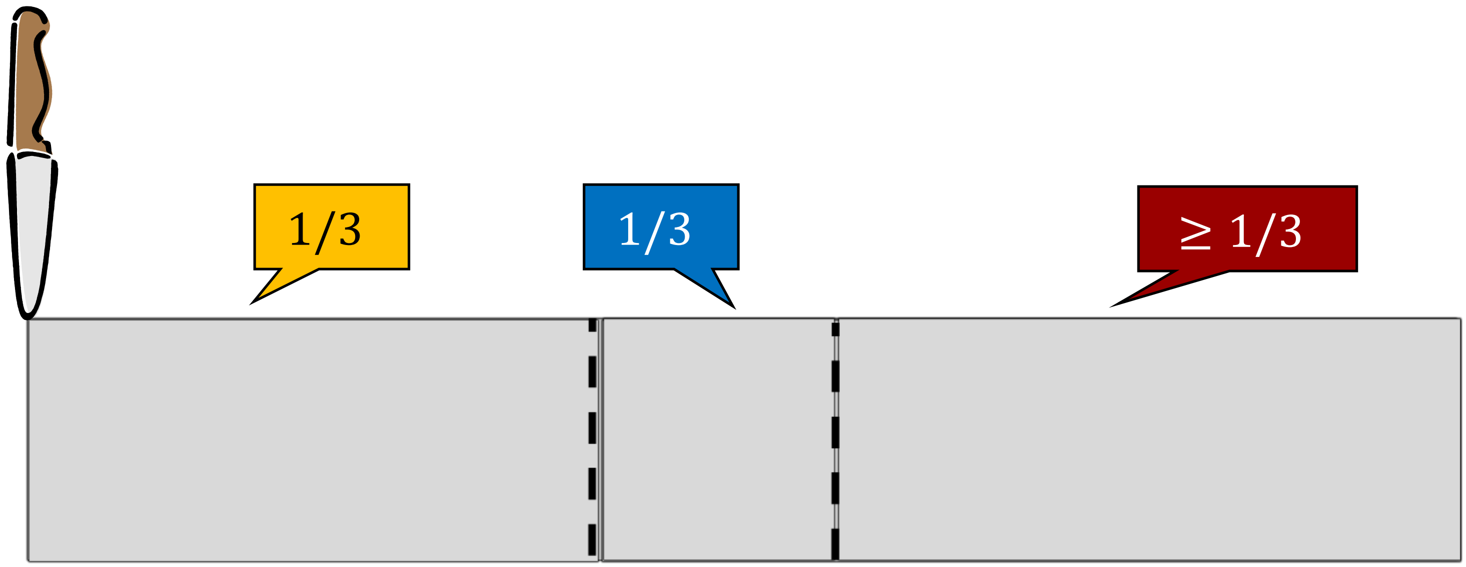
- Answer: 2
  - Why?

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



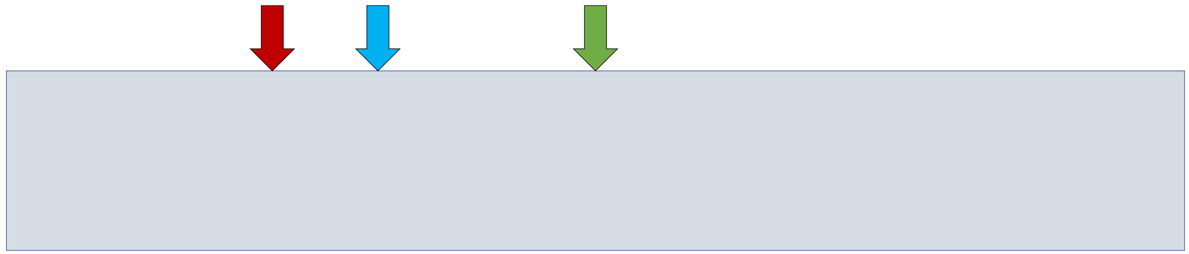
Dubins-Spanier

- Moving knife is not really needed.

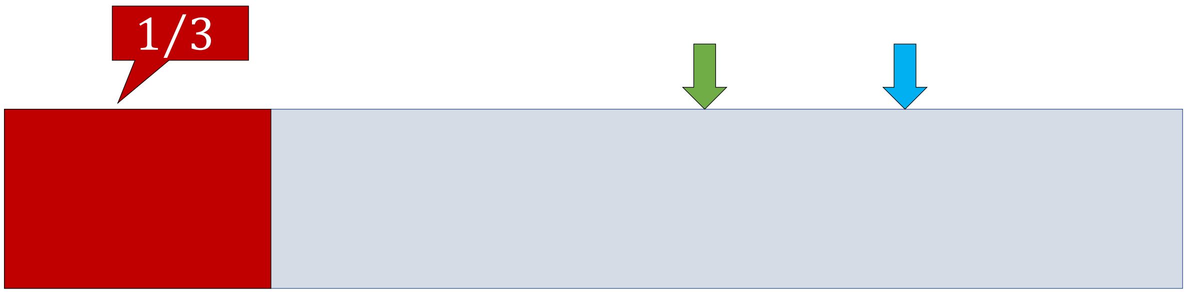
- At each stage, we can ask each remaining player a cut query to mark his $1/n$ point in the remaining cake.

- Move the knife to the leftmost mark.

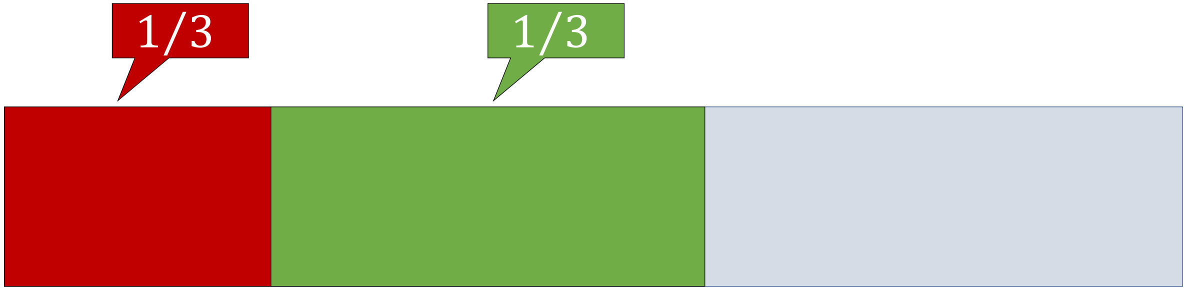
DUBINS-SPANIER



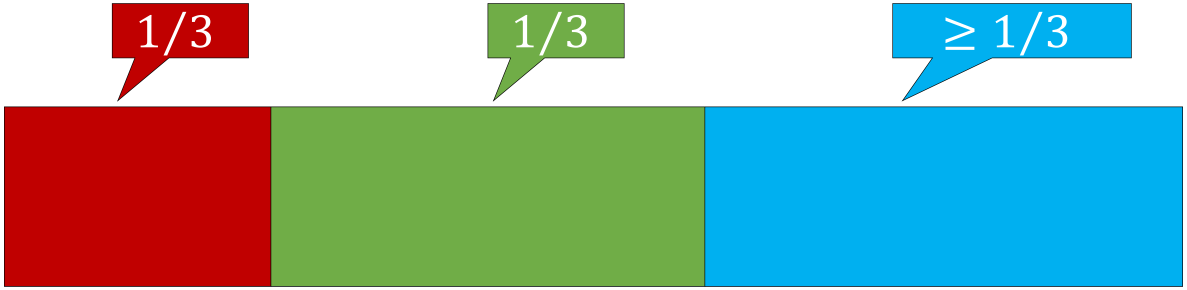
Dubins-Spanier



Dubins-Spanier



Dubins-Spanier



Dubins-Spanier

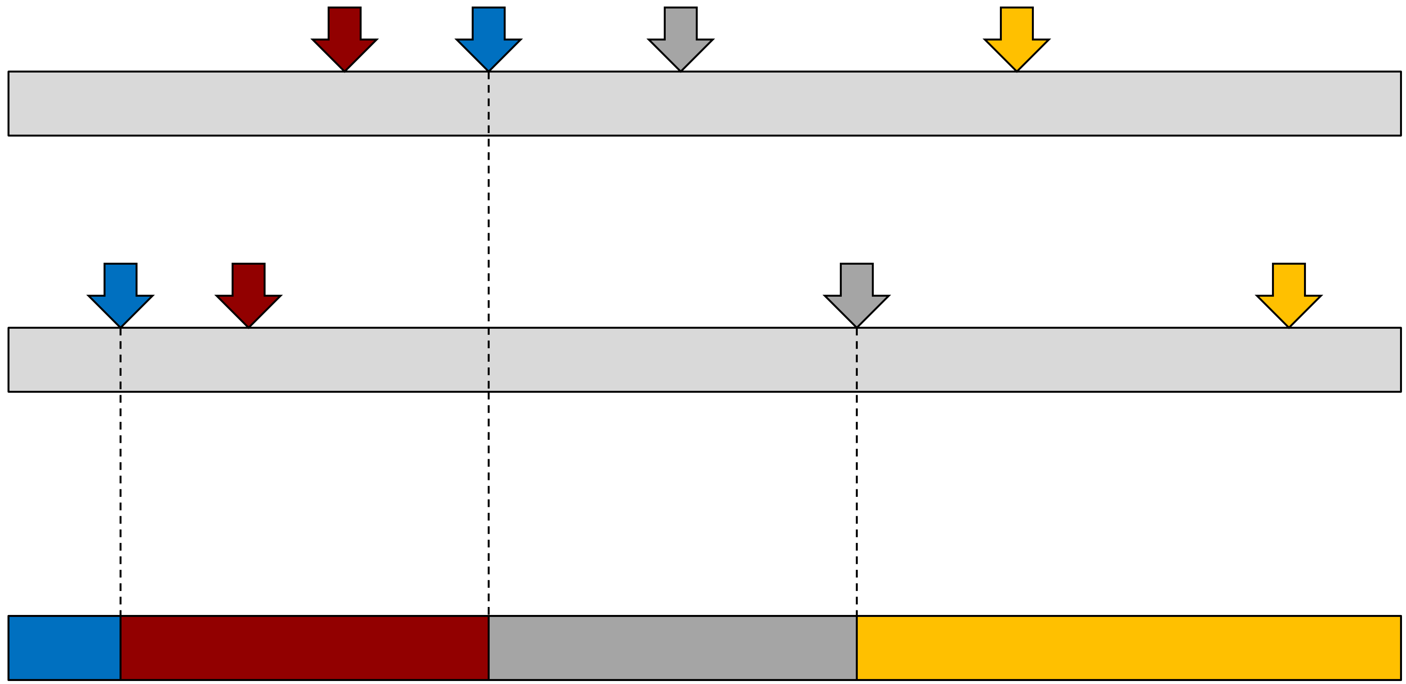
- Question: What is the complexity of the Dubins-Spanier protocol in the Robertson-Webb model?

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!

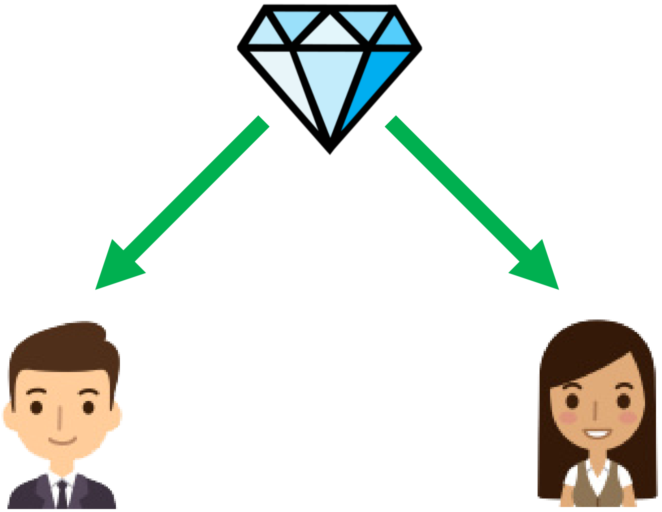
Envy-Freeness?

- "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!
    - Yes, it's not a typo. Go figure!

Indivisible Goods

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



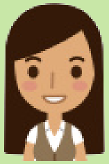
Indivisible Goods: Setting

				
	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$

Example Allocation

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

Indivisible Goods

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

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

  - If $A_j = \emptyset$, then we treat this to be true.
  - "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 cyclic order: $1,2,\ldots,n,1,2,\ldots,n,\ldots$
  - In her turn, an agent picks the good she likes the most among the goods still not picked by anyone.

- Observation: This always yields an EF1 allocation.
  - Informal proof on the board.

Efficient?

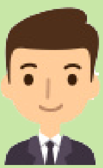
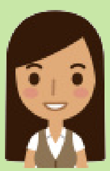
- Sadly, a round robin allocation can be suboptimal in the following strong sense:
  - It may be possible to redistribute the items to make everyone happier!

- Pareto optimality (PO):
  - We say that an allocation A is Pareto optimal if there exists no allocation B such 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^*.

EF1+PO?

- Does there always exist an allocation that is EF1+PO?
- Theorem [Caragiannis et al. '16]:
  - The MNW allocation $\text{argmax}_A \prod_{i \in N} V_i(A_i)$ is EF1 + PO.
  - Subtle note: If all allocations have zero Nash welfare:
    - Step 1: Call $S \subseteq N$ to be "feasible" if there is an allocation under which every player in S has a positive utility. Let S^* be a feasible set that has the maximum cardinality among all feasible sets.
    - Step 2: Choose $\text{argmax}_A \prod_{i \in S^*} V_i(A_i)$

MNW Allocation: the maximum product is $20 * (11+8) * 9 = 3420$

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

MNW Allocation

- MNW $\Rightarrow$ PO is trivial (WHY?)
- MNW $\Rightarrow$ EF1 (when $S^* = N$):
  - Transferring good $g \in A_j$ to A_i should not increase Nash welfare.

$$\forall g \in A_j \; : \; V_i(A_i + g) \cdot V_j\left(A_j - g\right) \leq V_i(A_i) \cdot V_j\left(A_j\right)$$

$$\Updownarrow$$

$$\forall g \in A_j : \frac{V_j(g)}{V_j\left(A_j\right)} \geq \frac{V_i(g)}{V_i(A_i + g)} \geq \frac{V_i(g)}{V_i(A_i + g^*)}$$

  - where $g^* = \text{argmax}_{g \in A_j} V_i(g)$
  - Now take sum over $g \in A_j$...

Computation

- Computing the MNW allocation is strongly NP-hard

- Open Question: Can we compute an EF1+PO allocation in polynomial time?
  - Not sure.
  - A recent paper gives a pseudo-polynomial time algorithm.
    - Polynomial time if the values are at most polynomial.

Stronger Fairness

- Open Question: Does there always exist an EFx allocation?
- EF1: $\forall i, j \in N, \exists g \in A_j : V_i(A_i) \geq V_i\left(A_j \setminus \{g\}\right)$
  - Intuitively, i doesn't envy j if she gets to remove her most valued item from j's bundle.
- EFx: $\forall i, j \in N, \forall g \in A_j : V_i(A_i) \geq V_i\left(A_j \setminus \{g\}\right)$
  - Subtle note: Either we need to assume strictly positive values, or change this to "$\forall g \in A_j$ s.t. $V_i(\{g\}) > 0$".
  - Intuitively, i should't envy j even if she removes her least positively valued item from j's bundle.

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.

Other Fairness Notions

- Maximin Share Guarantee
  - $MMS_i = \max_{(B_1,\ldots,B_n)} \min_k V_i(B_k)$
    - "If I divide the items into n bundles, but get the worst bundle, how much can I guarantee myself?"
  - α −MMS allocation: $V_i(A_i) \geq \alpha \cdot MMS_i$ for every i
- Theorem [Procaccia, Wang '14]:
  - There exists an instance on which no MMS allocation exists. ${}^2/_3$ −MMS always exists.
    - It can be computed in polynomial time [Amanatidis et al. '15]
- Theorem [Ghodsi et al. '17]:
  - ${}^3/_4$ −MMS always exists, and can be computed in polytime.
- MNW gives exactly $\frac{2}{1+\sqrt{4n-3}}$ −MMS.

CSC2556 : Algorithms for
Group Decision Making
(a.k.a. Computational Social Choice)