Fair Division

Eklavya Sharma
PhD student at ISE, UIUC
Advised by Prof. Jugal Garg

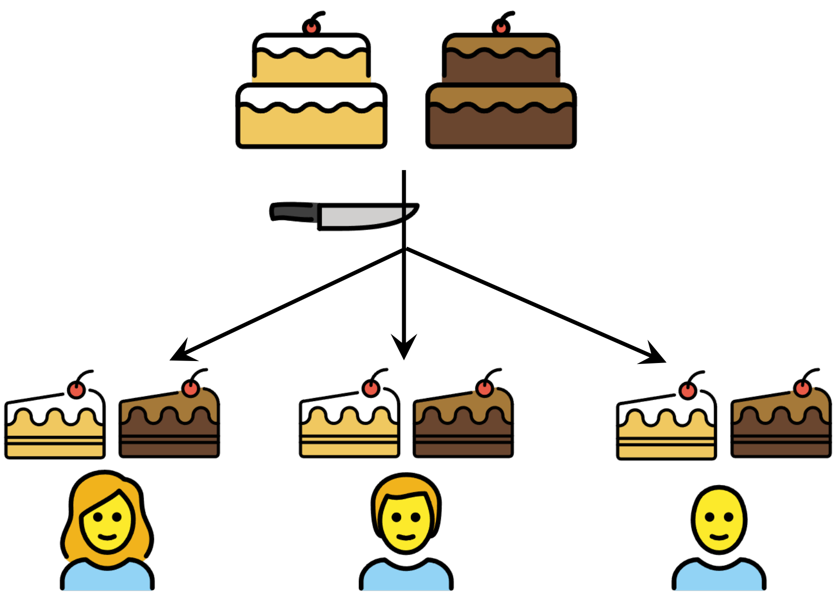
Before I begin

- Audience distribution.
- Please volunteer to talk at ISE student seminars.
- This is not a SoTA result talk. It's for a general audience. You should be able to follow along.
- If you have questions, interrupt me.

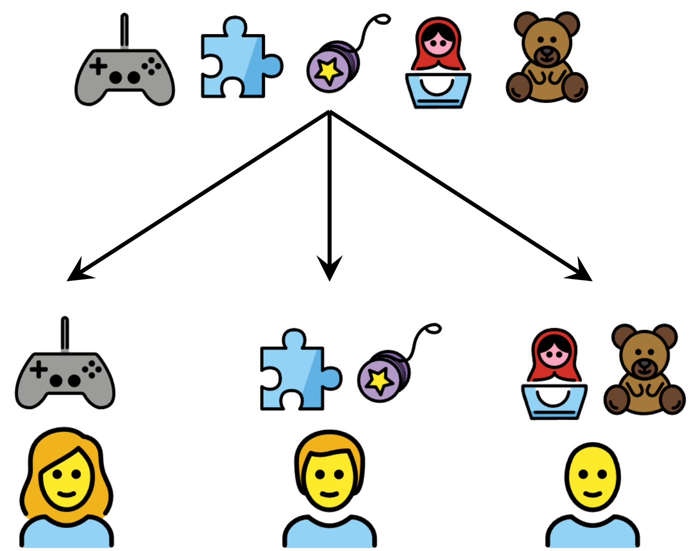
Fair Division of Goods

Divide goods among n people (called agents), who are all 'equally deserving'.

Divisible (continuous version):



Indivisible (discrete version):



Fair Division of Goods

- Trivial for divisible and homogeneous goods.
- Indivisible goods:
  - Toys, candy
  - Dividing Inheritance
  - Business partnership dissolution
  - Divorce settlements
- Divisible goods:
  - Cake
  - Land
  - Radio frequency spectrum
  - Airplane runway time

Formalizing the Problem

- Formalizing roughly means deciding input and output format.
- Set $N = \{1, 2, \dots, n\}$ of agents.
  Agents can be people, companies, countries, etc.
- Set M of goods:
  - Divisible: $M = [0, 1]$.
  - Indivisible: $M = \{1, 2, \dots, m\}$.

Valuation Functions

- Each agent i has a valuation function v_i that encodes their preferences:
  - Input: Subset $S \subseteq M$ of goods.
  - Output: A non-negative real number: how much they like S.
- Input: List $(v_1, v_2, \ldots, v_n)$ of valuation functions.
- $v_i(\emptyset) = 0$. Monotonicity: $S \subseteq T \Longrightarrow v_i(S) \leq v_i(T)$.

Additive valuations

- Additivity: $v_i(S \sqcup T) = v_i(S) + v_i(T)$.
  (All valuations in this talk are additive, except when I say otherwise.)
- Examples against additivity:
  pair of socks, pair of identical cars.
- For indivisible goods, additivity means
  $v_i(S) = \sum_{g \in S} v_i(\{g\})$.
- Indivisible input format: valuation matrix: $v_i(\{j\})$.

Output format

- An allocation X is a partition of M into n parts, i.e., $X = (X_1, X_2, \dots, X_n)$ is an n-tuple such that $\bigcup_{i \in N} X_i = M$ and $X_i \cap X_j = \emptyset$ for $i \neq j$.
- X_i is called agent i's *bundle* in allocation X.
- We need to find an allocation that is ***fair***.

Notions of Fairness

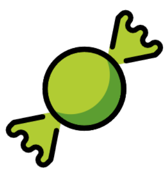
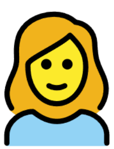
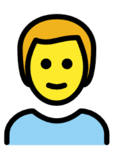
Envy-Freeness

- In allocation X, agent i envies agent j if $v_i(X_j) > v_i(X_i)$.
- X is envy-free (EF) if no one envies anyone else.

	🍬	🍪	🍭
👩	4	2	6
👨	5	15	25

Proportionality

- Allocation X is proportional (PROP) if $v_i(X_i) \geq v_i(M)/n$ for each agent i.
- This example is also PROP.

				avg
	4	2	6	6
	5	15	25	22.5

Dividing a cake among 2 agents

- Cut and choose protocol:
    1. Alice cuts cake into 2 pieces of equal value to her.
    2. Bob picks the piece which he prefers.
    3. Alice gets the remaining piece.
- The output is an envy-free allocation. (Why?)
- This works even for non-additive valuations.

Exercise: An EF allocation is PROP

- **Theorem**: If X is an EF allocation, then X is PROP.
  (only true for additive valuations.)

Exercise: An EF allocation is PROP

- **Theorem**: If X is an EF allocation, then X is PROP. (only true for additive valuations.)
- *Proof*. For each agent i,

$$v_i(X_i) \geq v_i(X_1)$$
$$v_i(X_i) \geq v_i(X_2)$$
$$\vdots$$
$$v_i(X_i) \geq v_i(X_n)$$

- Add all these inequalities together to get

$$n v_i(X_i) \geq v_i(M)$$

Dubins-Spanier Algorithm

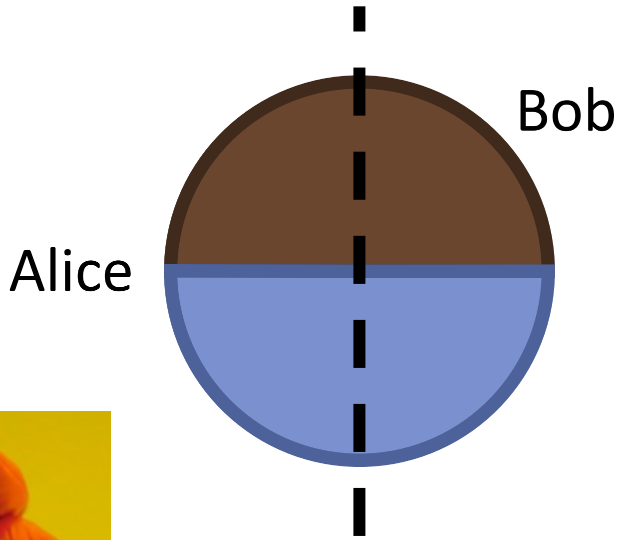
- Algorithm for PROP cake cutting for n agents.
- Cake is 1-dimensional interval $M = [a, b]$.
- Algorithm:
  - Each agent i tells a point $x_i \in [a, b]$ such that $v_i([a, x_i]) = v_i(M)/n$.
  - Suppose agent k has the smallest x_k.
  - Give $[a, x_k]$ to agent k and recurse.
- Key observation: $v_i(M \setminus [a, x_k]) \geq \frac{n-1}{n} v_i(M)\ \forall i$.
- Bonus: Pieces are connected.

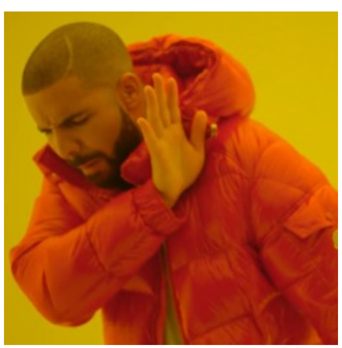
EF cake cutting

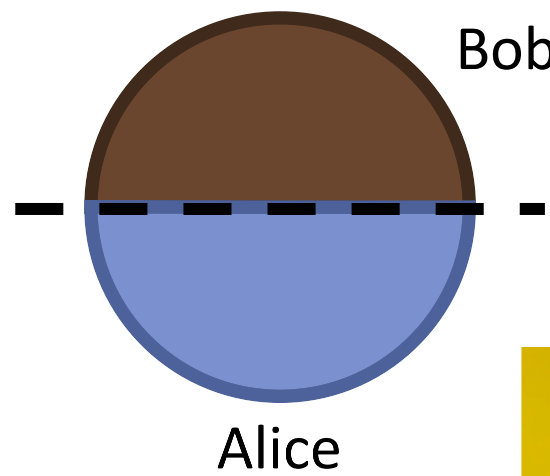
- EF allocations exist, even when we insist on connected pieces.
- EF algorithms for $n \geq 3$ are complicated and have a large running time.
- For piecewise-constant valuation functions, efficient EF algorithms exist.

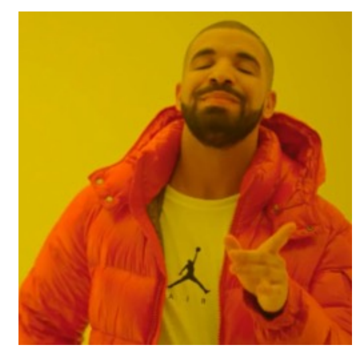
Efficiency

- Fairness isn't the only concern.
- Alice prefers blueberry and Bob prefers chocolate.
- Both allocations are fair. But one is better.









Pareto Optimality (PO)

- Intuitively, an allocation is pareto optimal (PO) if it’s impossible to make someone happier without making someone else sadder.
- Allocation X pareto-dominates allocation Y iff both of the following are true:
  - no one prefers Y: $\forall i, v_i(X_i) \geq v_i(Y_i)$.
  - someone prefers X: $\exists i, v_i(X_i) > v_i(Y_i)$.
- Allocation X is pareto-optimal (PO) if it is not pareto-dominated by any other allocation.

Nash Social Welfare (NSW)

- PO is a weak notion: maybe we can make someone a lot happier by making someone slightly sadder.
- NSW is the ‘average’ happiness of an allocation.
- $\mathrm{NSW}(X) = \sqrt[n]{v_1(X_1) v_2(X_2) \cdots v_n(X_n)}$.
- An alloc that maximizes NSW is Nash Optimal (NO).

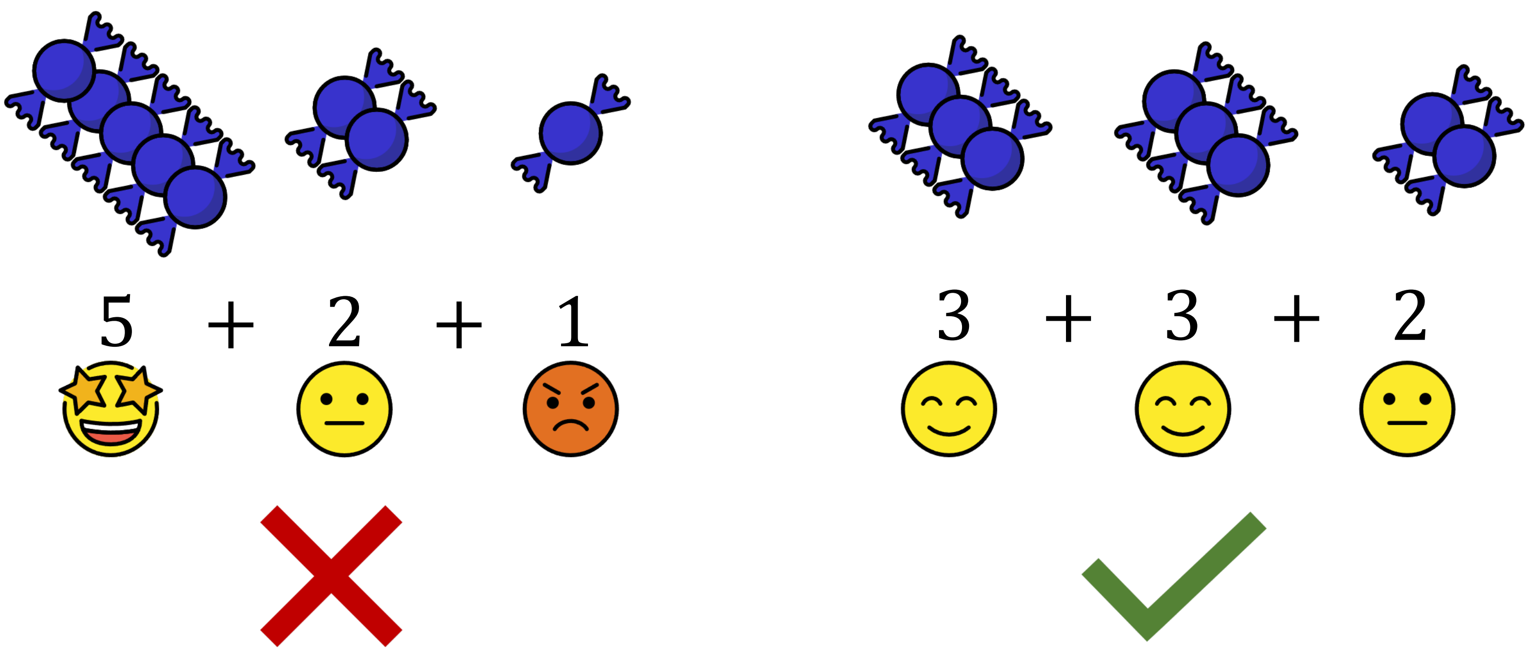
Nash Social Welfare (NSW)

- PO is a weak notion: maybe we can make someone a lot happier by making someone slightly sadder.
- NSW is the ‘average’ happiness of an allocation.
- $\mathrm{NSW}(X) = \sqrt[n]{v_1(X_1) v_2(X_2) \cdots v_n(X_n)}$.
- An alloc that maximizes NSW is Nash Optimal (NO).
- NO implies PO.
- In practice, NO allocations are fair.
- A Nash-optimal cake division is EF[1].

Fairness for Indivisible Goods

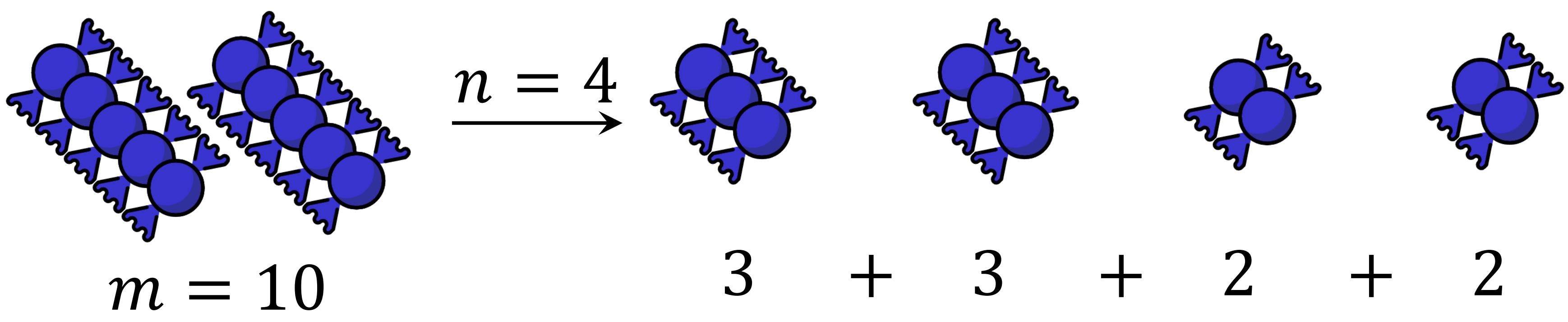
EF and PROP are not guaranteed

- For divisible goods, EF and PROP always exist.
- But not for indivisible goods: e.g., single good.
- We can't be fair, but we can be *approximately* fair.



Fairness for the Indivisible setting

- Suppose there are m identical goods and n agents.
- Each agent should get $\lfloor m/n \rfloor$ or $\lceil m/n \rceil$ goods.
- How do we generalize this idea?



EFX (EF up to any good)

- Observation: $\lceil m/n \rceil - \lfloor m/n \rfloor \leq 1$.
- In allocation X, agent i strongly envies agent j if $\exists g \in X_j$ s.t. $v_i(X_i) < v_i(X_j - \{g\})$.
  Equivalently, $v_i(X_i) < \max_{g \in X_j} v_i\big(X_j - \{g\}\big)$.
- X is EFX if no one strongly envies anyone else.

	🍭	🍪	🍬	🍩
👩	4	4	2	7
👨	14	14	12	17

	🍭	🍪	🍬	🍩
👩	8		9	
👨	28		29	

EFX: Existence and Computation

- Important problems:
  - Do EFX allocations always exist?
  - Can we efficiently compute EFX allocations?
- EFX exists when $n = 2$ or identical valuations (even for non-additive) [PR SODA'18].
- EFX exists for $n = 3$ [CGM EC'20].
- For $n \geq 4$, open problem since 2016.
- Relaxations of EFX:
  - EF1 [EC'04], α-EFX [TCS'20], EFX-with-charity [SODA'20], Epistemic EFX.

MaxiMin Share (MMS)

- MMS: relaxation of PROP for indivisible goods.
- Threshold based fairness: $v_i(X_i) \geq \mu_i$.
  - PROP: $\mu_i = v_i(M)/n$.

	🍬	🍬	🍬	🍩
👩	1	1	1	6
👨	1	1	1	6
🧑	10	10	10	60

$\mu_1 = \mu_2 = 1, \mu_3 = 10$

- MMS: μ_i is the max value so that $\exists X, v_i(X_j) \geq \mu_i \ \forall j$.

$$\mu_i = \max_X \min_{j \in N} v_i(X_j)$$

- X is MMS if $v_i(X_i) \geq \mu_i \ \forall i$.

MMS: Existence and Computation

- Important problems:
  - Do MMS allocations always exist?
  - Can we efficiently compute MMS allocations?
- MMS exists when idval. NP-hard even for $n = 2$.
- There is a known example with $n = 3$ for which an MMS allocation doesn't exist [EC'14].
- Relaxation: α-MMS: $v_i(X_i) \geq \alpha\mu_i \ \forall i \ (\alpha \in (0,1])$.
  - (3/4)-MMS in strong polytime [GT EC'20].

Summary

- Formalizing fair division.
- Divisible goods:
  - EF and PROP.
  - EF for 2 agents using cut-and-choose.
  - PROP using Dubins-Spanier.
  - Efficiency: PO and NSW.
- Indivisible goods:
  - EFX and MMS.

Social Choice Theory

How can multiple agents make a joint decision?

- Fair division of goods.
- Fair division of chores.
- Splitting rent.
- Which activities should a group of friends do together over the weekend?

Thank You

Questions?

Image credits: [openmoji.org]() ([CC BY-SA 4.0]()), imgflip.com

Homework

Fair cake cutting:

1. Show that when agents have identical valuations, a Nash optimal allocation is envy free.

Fair division of indivisible goods:

1. Find a PROP allocation that is not EF.
2. Show that MMS allocations exist when:
   i. there are only 2 agents (hint: cut-and-choose).
   ii. all agents have the same valuation function.
3. Give a fast algorithm to find an EFX allocation when:
   i. there are only 2 agents (hint: cut-and-choose).
   ii. all agents have the same valuation function (hint: greedy).