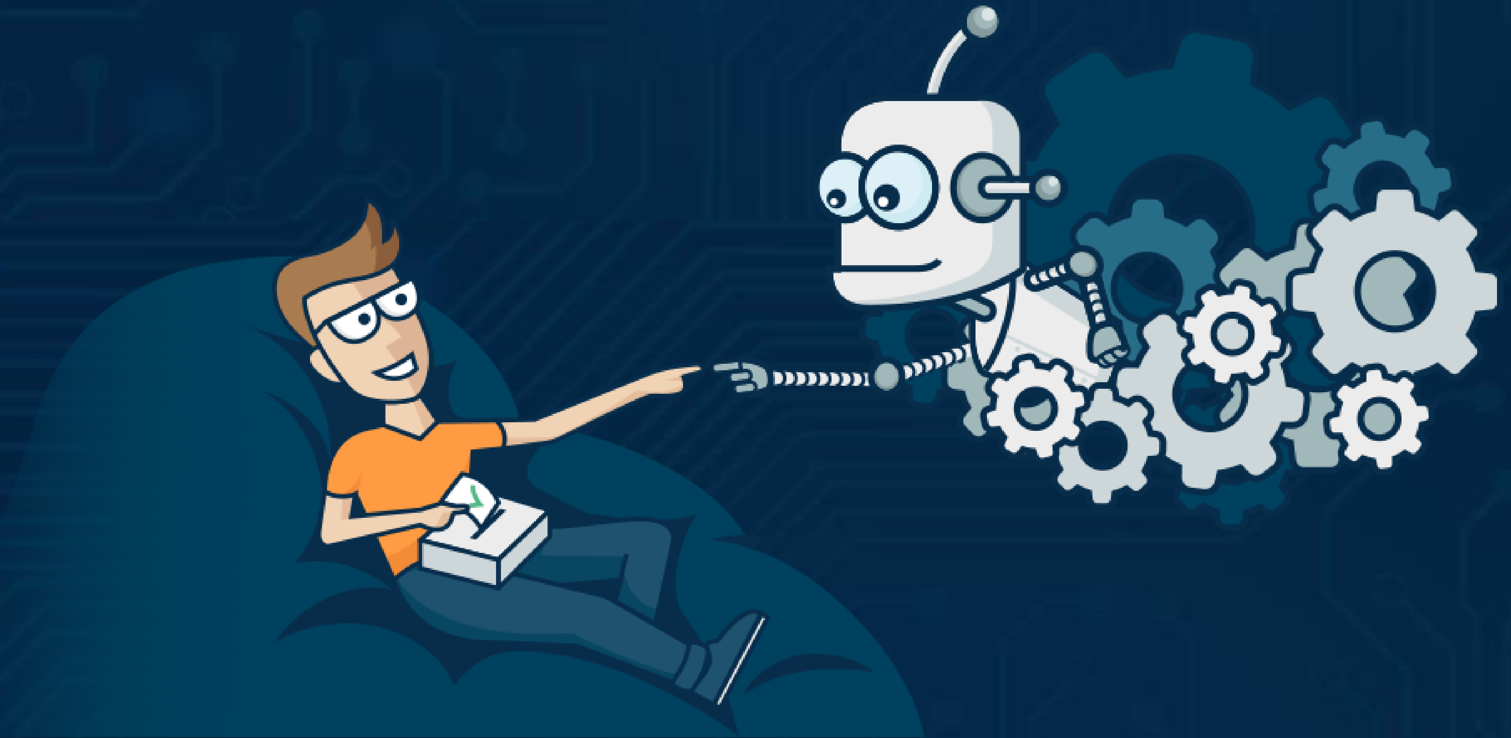
Great Ideas in Fair Division

Nisarg Shah

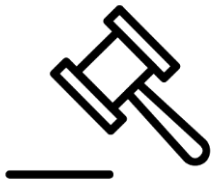
University of Toronto



Algorithms Making Decisions

Loans

Bails



Self-Driving Cars

Ads

Hiring

Organ Exchange

Computational Social Choice

Algorithms for aggregating individual preferences
towards collective decisions

Reasonable Collective Decisions

Cake Cutting

- Formally introduced by Steinhaus [1948]
- n agents
- Cake modeled as [0,1]
- Allocate the cake
  - A_i is the part given to agent i
  - Can be union of several disjoint intervals

Agent Valuations

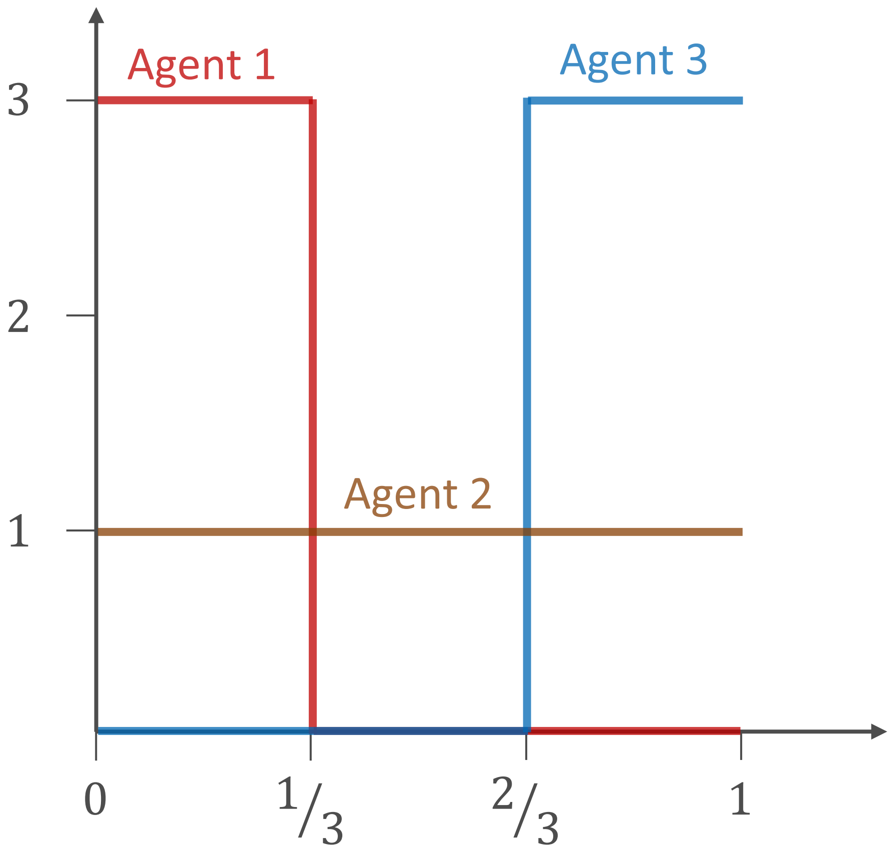
- Each agent i has an integrable density function $f_i: [0,1] \to \mathbb{R}_+$

- $v_i(X) = \int_{x \in X} f_i(x) dx$

- Normalization: $\int_0^1 f_i(x) dx = 1$
    - Without loss of generality

EXAMPLE

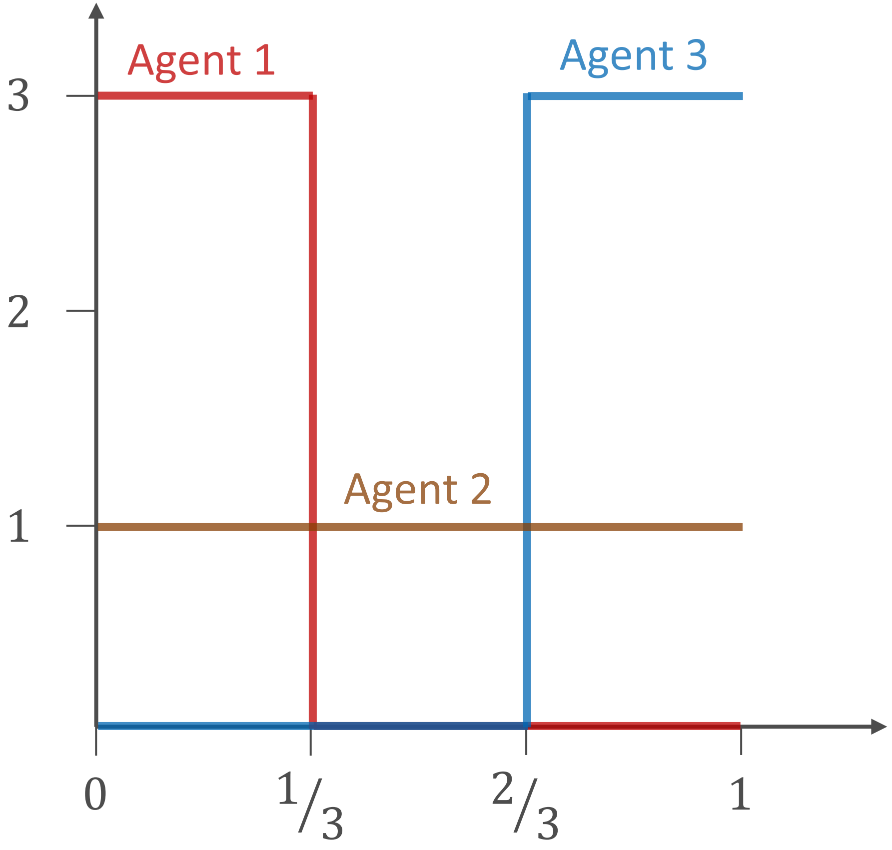
- Value density functions



- Agent 1 wants $[0, {}^1/_3]$ uniformly and does not want anything else
- Agent 2 wants the entire cake uniformly
- Agent 3 wants $[{}^2/_3\,, 1]$ uniformly and does not want anything else

Example

- Value density functions



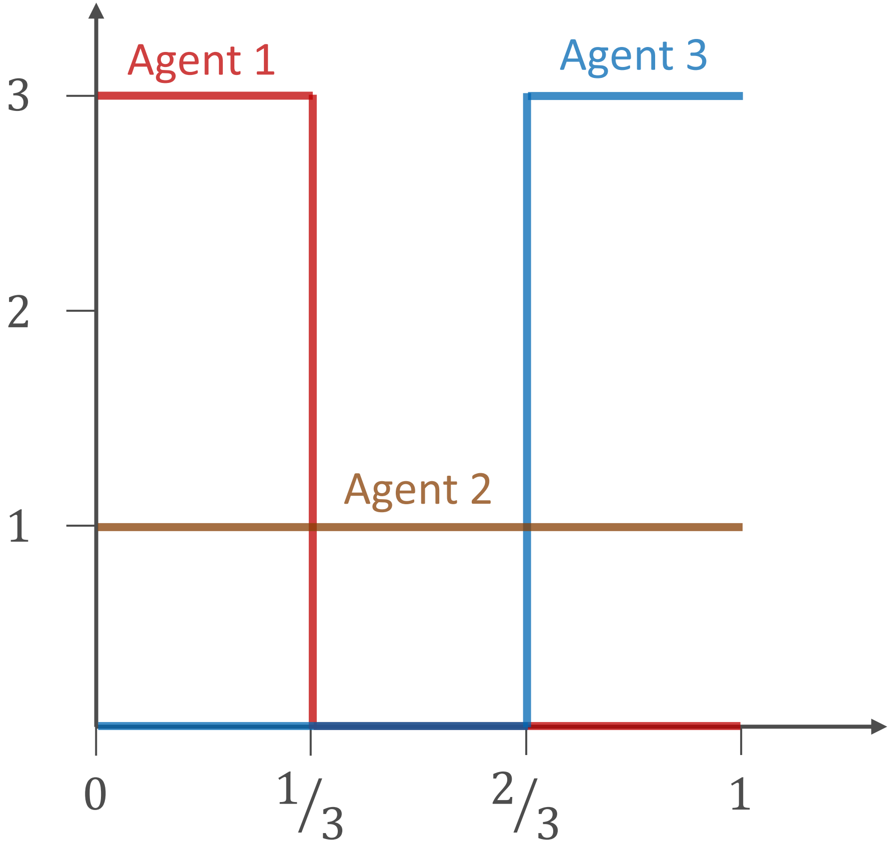
- Consider the following allocation

- $A_1 = [0, {}^1\!/_9] \Rightarrow v_1(A_1) = {}^1\!/_3$
- $A_2 = [{}^1\!/_9, {}^8\!/_9] \Rightarrow v_2(A_2) = {}^7\!/_9$
- $A_3 = [{}^8\!/_9, 1] \Rightarrow v_3(A_3) = {}^1\!/_3$

- Each of three agents is getting at least one-third of their value, which seems fair in some sense
- But agent 1 and 3 are envious of how well agent 2 is treated

EXAMPLE

- Value density functions



- Consider the following allocation

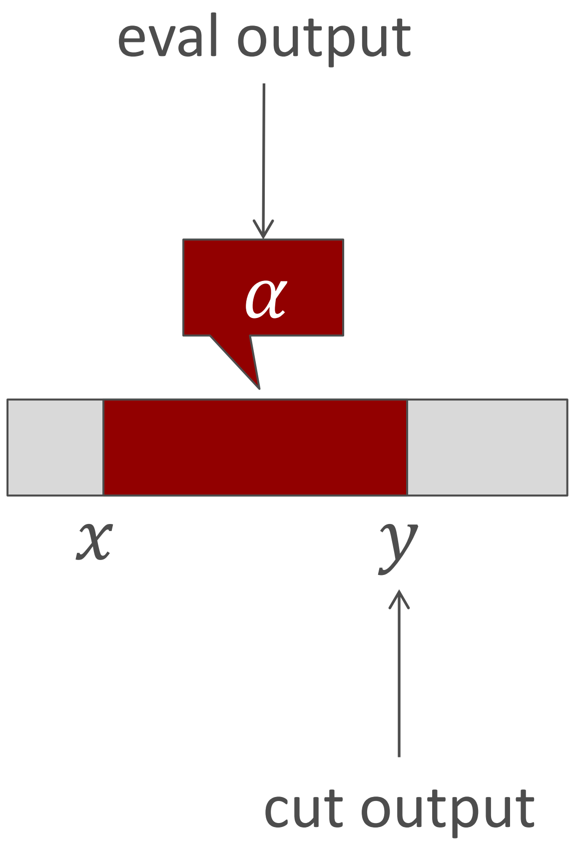
- $A_1 = [0, {}^1\!/_6] \Rightarrow v_1(A_1) = {}^1\!/_2$
- $A_2 = [{}^1\!/_6, {}^5\!/_6] \Rightarrow v_2(A_2) = {}^2\!/_3$
- $A_3 = [{}^5\!/_6, 1] \Rightarrow v_3(A_3) = {}^1\!/_2$

- Now agent 1 and 3 are not envious of what agent 2 is given, even though agent 2 has more utility than them

COMPLEXITY

- Inputs are functions
  - Infinitely many bits may be needed to fully represent the input
  - Query complexity is more useful

- Robertson-Webb Model
  - $\text{Eval}_i(x, y)$ returns $v_i([x, y])$
  - $\text{Cut}_i(x, \alpha)$ returns y such that $v_i([x, y]) = \alpha$



Three Classic Fairness Desiderata

- Proportionality (Prop): $\forall i \in N$: $v_i(A_i) \geq {}^1/_n$
  - Each agent should receive her "fair share" of the utility.
- Envy-Freeness (EF): $\forall i, j \in N$: $v_i(A_i) \geq v_i(A_j)$
  - No agent should wish to swap her allocation with another agent.
- Envy-freeness implies proportionality (Why?)

Proportionality

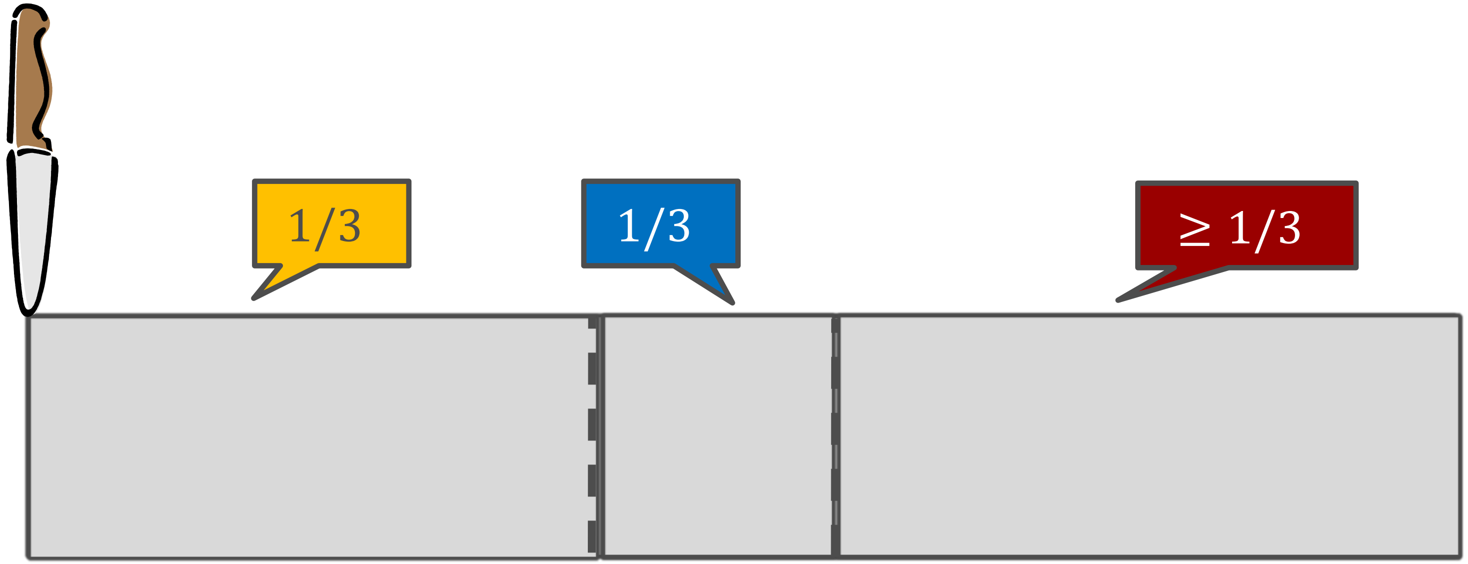
Proportionality : $n = 2$ agents

- **Cut-and-choose**
  - Agent 1 cuts the cake at x such that $v_1([0, x]) = v_1([x, 1]) = 1/2$
  - Agent 2 chooses the piece that she prefers.

- Elegant protocol
  - Envy-free for 2 agents
  - Needs only one cut and one eval query (optimal)

- More agents?

Proportionality: Dubins-Spanier



Animation Credit: Ariel Procaccia

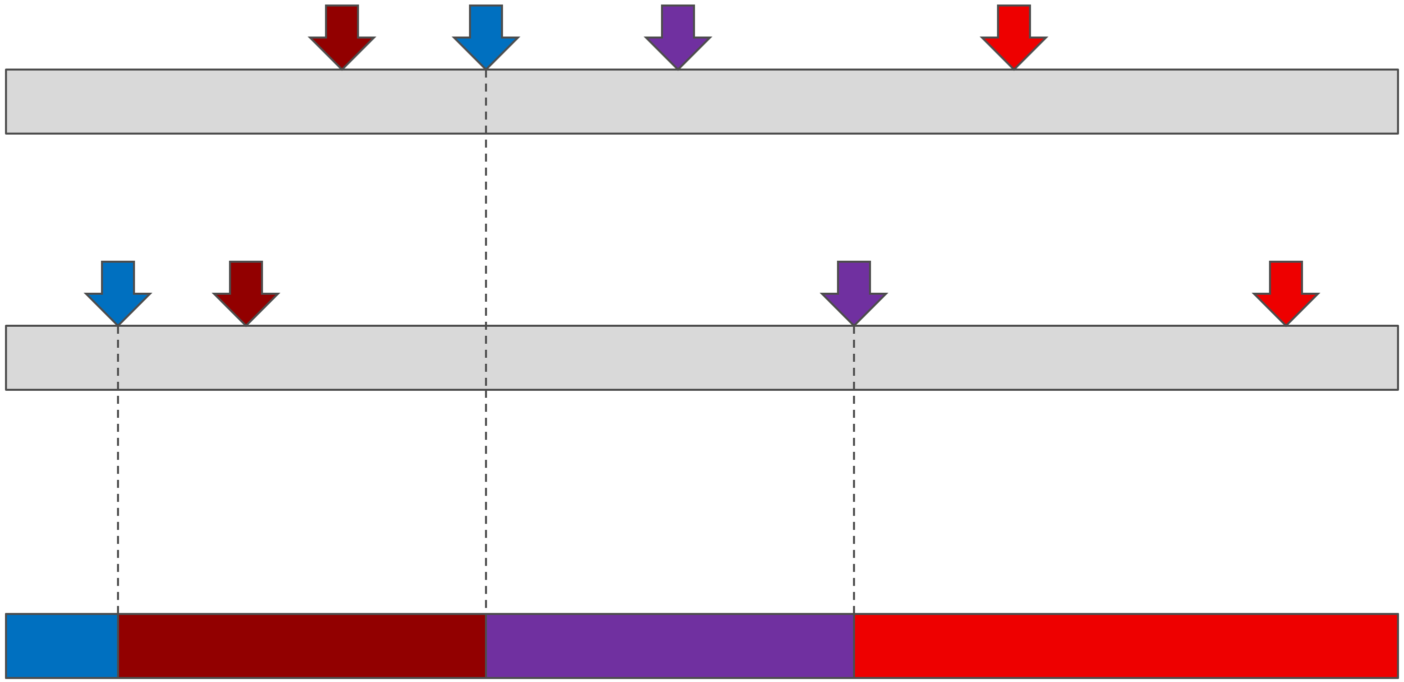
PROPORTIONALITY: DUBINS-SPANIER

- DUBINS-SPANIER
  - Referee starts a knife at 0 and moves the knife to the right.
  - Repeat: When the piece to the left of the knife is worth $1/n$ to an agent, the agent shouts "stop", receives the piece, and exits.
  - When only one agent remains, she gets the remaining piece.

- Can be implemented easily in Robertson-Webb model
  - When $[x, 1]$ is left, ask each remaining agent i to cut at y_i so that $v_i([x, y_i]) = 1/n$, and give agent $i^* \in \arg\min_i y_i$ the piece $[x, y_{i^*}]$

- Query complexity: $\Theta(n^2)$

EVEN-PAZ



Animation Credit: Ariel Procaccia

PROPORTIONALITY: EVEN-PAZ

- EVEN-PAZ
- Input:
  - Interval $[x, y]$, number of agents n (assume a power of 2 for simplicity)
- Recursive procedure:
  - If $n = 1$, give $[x, y]$ to the single agent.
  - Otherwise:
    - Each agent i marks z_i such that $v_i([x, z_i]) = v_i([z_i, y])$
    - $z^* = (n/2)^{\text{th}}$ mark from the left.
    - Recurse on $[x, z^*]$ with the left $n/2$ agents, and on $[z^*, y]$ with the right $n/2$ agents.

- Query complexity: $\Theta(n \log n)$

Complexity of Proportionality

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

- Hence, Even-Paz is provably (asymptotically) optimal!

Envy-Freeness

ENVY-FREENESS : FEW AGENTS

- $n = 2$ agents : CUT-AND-CHOOSE (2 queries)
- $n = 3$ agents : SELFRIDGE-CONWAY (14 queries)

Gets complex pretty quickly!

Suppose we have three players **P1**, **P2** and **P3**. Where the procedure gives a criterion for a decision it means that criterion gives an optimum choice for the player.

1. **P1** divides the cake into three pieces he considers of equal size.
2. Let's call **A** the largest piece according to **P2**.
3. **P2** cuts off a bit of **A** to make it the same size as the second largest. Now **A** is divided into: the trimmed piece **A1** and the trimmings **A2**. Leave the trimmings **A2** to the side for now.
   - If **P2** thinks that the two largest parts are equal (such that no trimming is needed), then each player chooses a part in this order: **P3**, **P2** and finally **P1**.
4. **P3** chooses a piece among **A1** and the two other pieces.
5. **P2** chooses a piece with the limitation that if **P3** didn't choose **A1**, **P2** must choose it.
6. **P1** chooses the last piece leaving just the trimmings **A2** to be divided.

It remains to divide the trimmings **A2**. The trimmed piece **A1** has been chosen by either **P2** or **P3**; let's call the player who chose it **PA** and the other player **PB**.

1. **PB** cuts **A2** into three equal pieces.
2. **PA** chooses a piece of **A2** - we name it **A21**.
3. **P1** chooses a piece of **A2** - we name it **A22**.
4. **PB** chooses the last remaining piece of **A2** - we name it **A23**.

Envy-Freeness : Few Agents

- [Brams and Taylor, 1995]
  - The first finite (but unbounded) protocol for any number of agents
- [Aziz and Mackenzie, 2016a]
  - The first bounded protocol for 4 agents (at most 203 queries)
- [Amanatidis et al., 2018]
  - A simplified version of the above protocol for 4 agents (at most 171 queries)

Envy-Freeness

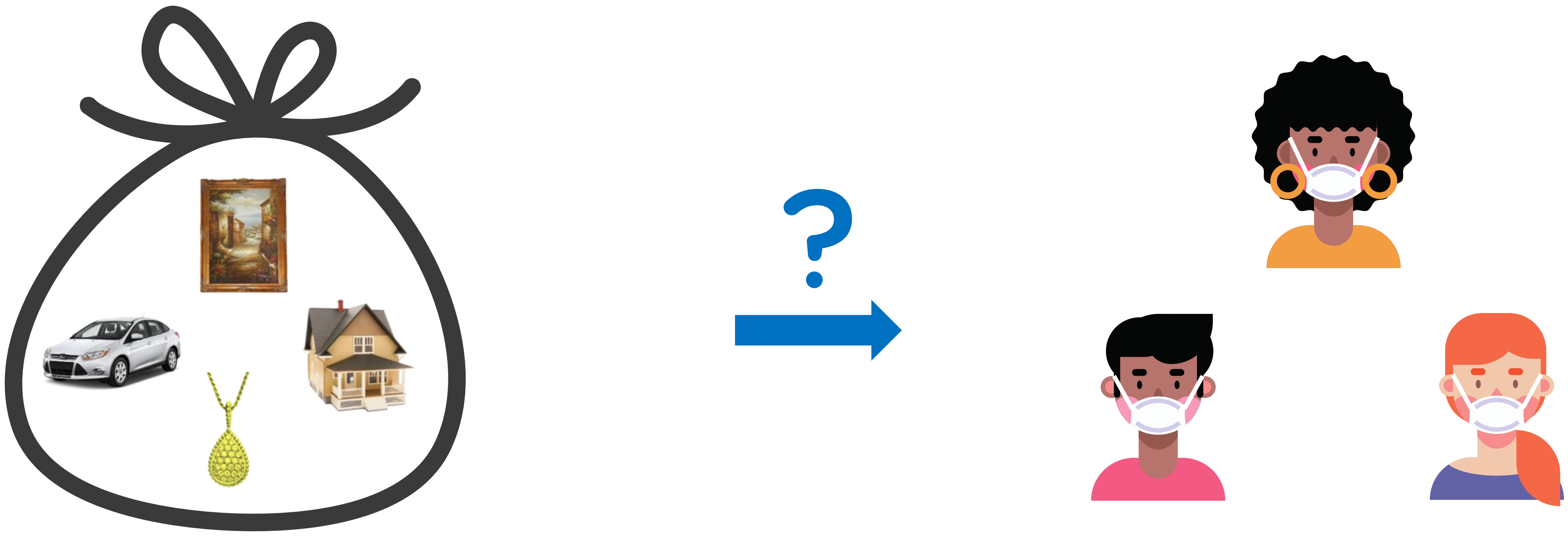
- Theorem [Aziz and Mackenzie, 2016b]
  - There exists a bounded protocol for computing an envy-free allocation with n agents, which requires $O(n^{n^{n^{n^{n^n}}}})$ queries

- Theorem [Procaccia, 2009]
  Any protocol for finding an envy-free allocation requires $\Omega(n^2)$ queries.

Open Problem

Bridge the gap between $O(n^{n^{n^{n^{n^n}}}})$ upper bound and $\Omega(n^2)$ lower bound for envy-free cake-cutting

Indivisible Goods



- Estate (inheritance) division
- Divorce settlement
- Friends splitting jointly purchased items
- …

splíddít

DIVIDE: RENT FARE CREDIT GOODS TASKS | ABOUT FEEDBACK

PROVABLY FAIR SOLUTIONS.

Spliddit offers quick, free solutions to everyday fair division problems, using methods that provide indisputable fairness guarantees and build on decades of research in economics, mathematics, and computer science.

Share Rent

Split Fare

Assign Credit

Divide Goods

Distribute Tasks

Suggest an App

Explainability

Envy-freeness

A division of goods is envy free if each participant believes that her bundle of goods is at least as valuable as every other participant's bundle, i.e., no participant envies any other participant. While our algorithm may often find an envy-free division, no algorithm can guarantee one.

Our algorithm guarantees a division that is envy free up to one good: A participant would never envy another participant if we removed a single good from the other participant's bundle. In fact, if the contested good is divisible, in the sense that it can be broken down into smaller pieces (e.g., cash, stocks), then we could eliminate envy by removing a hundredth (1%) of it.

Efficiency

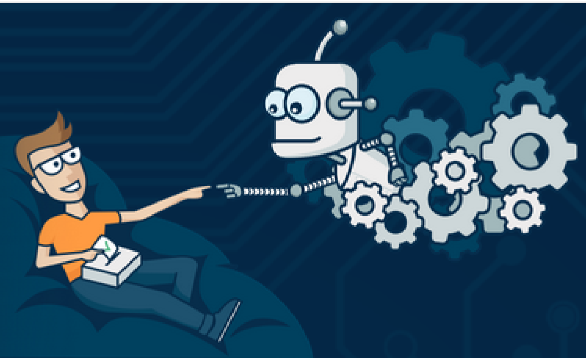
Our algorithm divides the goods in such a way that it would be impossible to find another division that benefits a participant without making another participant worse off.

ROBOVOTE

ROBOVOTE

AI-Driven Decisions

RoboVote is a free service that helps users combine their preferences or opinions into optimal decisions. To do so, RoboVote employs state-of-the-art voting methods developed in artificial intelligence research.
Learn More



Poll Types

RoboVote offers two types of polls, which are tailored to different scenarios; it is up to users to indicate to RoboVote which scenario best fits the problem at hand.

Objective Opinions

In this scenario, some alternatives are objectively better than others, and the opinion of a participant reflects an attempt to estimate the correct order. RoboVote's proposed outcome is guaranteed to be as close as possible — based on the available information — to the best outcome. Examples include deciding which product prototype to develop, or which company to invest in, based on a metric such as projected revenue or market share.

Subjective Preferences

In this scenario participants' preferences reflect their subjective taste; RoboVote proposes an outcome that mathematically makes participants as happy as possible overall. Common examples include deciding which restaurant or movie to go to as a group.

Ready to get started?

CREATE A POLL

Under the Hood

We use two fundamentally different algorithmic approaches, depending on whether the poll type corresponds to subjective preferences or objective opinions.

Subjective Preferences

For subjective preferences, the approach is known as *implicit utilitarian voting*. We assume that each participant has a (subjective) utility function that assigns an exact utility to each alternative. Our goal is to choose an outcome that maximizes utilitarian social welfare, which is the total utility assigned to the outcome by all participants. However, in order to minimize the cognitive burden imposed on participants, we only ask for a ranking of the alternatives. The algorithm selects the outcome whose utilitarian social welfare is closest to the optimum, in the worst case over all possible utility functions that are consistent with the reported rankings, in the sense that alternatives with higher utility are ranked higher.

Further Details

Optimal Social Choice Functions: A Utilitarian View, by Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel Procaccia, and Or Sheffet.

Subset Selection Via Implicit Utilitarian Voting, by Ioannis Caragiannis, Swaprava Nath, Ariel Procaccia, and Nisarg Shah.

Objective Opinions

For objective opinions, let us focus first on the case where the desired outcome is a ranking of the alternatives. We assume that there is a true ranking of the alternatives by relative quality, and our goal is to pinpoint a ranking that is as close as possible to the true ranking, given the available information. Specifically, the distance between two rankings is the number of disagreements between them on the relative ranking of pairs of alternatives (which is known as the *Kendall tau* distance). We implicitly compute the set of all feasible true rankings, under an assumption on the average number of times a participant fails to rank two alternatives in the correct order. Then, we select the ranking that minimizes the maximum distance to any ranking in the set of feasible true rankings. To select a single alternative, we again compute the set of feasible true rankings, and choose the alternative whose worst position in any of these rankings is as good as possible. To select a subset of alternatives, we choose the best alternatives according to the same criterion.

Further Details

Voting Rules As Error-Correcting Codes, by Ariel Procaccia, Nisarg Shah, and Yair Zick.

Thank You