MAX PLANCK INSTITUTE FOR INFORMATICS

SIC Saarland Informatics Campus

Mechanism Design Without Money

Kurt Mehlhorn, Javier Cembrano, Golnoosh Shahkarami

Max Planck Institute for Informatics

April 22, 2025

Introduction

Please Introduce yourself!

- What is your name?
- Where are you from?
- Which Program are you studying?
- Which Semester?
- How much were you exposed to Algorithms?

Organization

• Seminar	2+0, 7 CPS
• Organizers	Kurt Mehlhorn, Javier Cembrano, and Golnoosh Shahkarami
• Time	Every Tuesday 14:15 - 15:45
• Requirements	Basic algorithms lecture
• Structure	- First 4 lectures by us - One week break - Student presentation

Organization

- Your task
  1. Send us your preferred order of the papers — **April 29th**
  2. Give a practice talk — **One week** before your talk
  3. Present a paper (60-75 minutes) — May 27th – July 15th
  4. Write a summary of the paper — **July 22nd**

- Grade
  - 70% presentation
  - 20% summary
  - 10% engagement

Games

- Prisoner's Dilemma
  - Two suspects are arrested
  - Police has no evidence
  - Each have two options: confess or stay quiet
  - They cannot talk to each other
  - They are selfish

	NC	C
NC	2, 2	4, 0
C	0, 4	3, 3

- Dominant strategy Bob: C, Alice: C
- Nash equilibrium (C, C)
- Pareto optimal (NC, NC)

Mechanism Design

- How can we model strategic behavior?
  - Each agent wants to optimize their own utility rather than the general goal of making everyone happy
- What affects does strategic behavior have?
- How should we set rules of the games?

How to design systems that can cope with strategic behavior?

Using financial compensation: **mechanism design with money**

Using voting mechanism: **mechanism design without money**

Mechanism Design with Money - Auctions

- One item to sell 
- A: set of agents (player/bidders)
- v_i: value of agent i for winning the item
- b_i: reported bid of agent i
- $u_i = v_i - p$: utility of agent i in case of winning

  ($u_i = 0$ otherwise)
- Mechanism:
  - Who wins? 
  - How much should they pay? 

Value	Agents	Bids
v_1		b_1
v_2		b_2
v_3		b_3
v_4		b_4
v_5		b_5
v_6		b_6

Mechanism Design with Money - Auctions

- One item to sell 
- A: set of agents (player/bidders)
- v_i: value of agent i for winning the item
- b_i: reported bid of agent i
- $u_i = v_i - b_i$: utility of agent i in case of winning

  ($u_i = 0$ otherwise)
- **First-price Auction**:
  - Who wins? Bidder with highest bid
  - How much should they pay? Their bid

Not a Nash equilibrium!

Value	Agents	Bids
v_1		b_1
v_2		b_2
v_3		b_3
v_4		b_4
v_5		b_5
v_6		b_6

Mechanism Design with Money - Auctions

- One item to sell 
- $\boldsymbol{A}$: set of agents (player/bidders)
- $\boldsymbol{v_i}$: value of agent i for winning the item
- $\boldsymbol{b_i}$: reported bid of agent i
- $\boldsymbol{u_i = v_i - b_j}$: utility of agent i in case of winning
  ($\boldsymbol{u_i = 0}$ otherwise)
- **Second-price Auction**:
  - Who wins? Bidder with highest bid
  - How much should they pay? **2nd** highest bid b_j

Value	Agents	Bids
v_1		b_1
v_2		b_2
v_3		b_3
v_4		b_4
v_5		b_5
v_6		b_6

Mechanism Design with Money - Auctions

- A bid b_i is a (weakly) **dominant strategy** for player i with value v_i if for all possible bids b'_i by that player and all possible bids b_{-i} of the other players,

$$u_i((b_i\ ,\ b_{-i}\ ),\ v_i) \geq u_i((b'_i\ ,\ b_{-i}),\ v_i).$$

Theorem – Vickrey, 1961

In a Second-price auction, for each player it is a dominant strategy to bid **truthfully**.

Proof. Fix a player i, we need to show that u_i is maximized by $b_i = v_i$.

Let $b_j = max_{k \neq i}\ b_k$. Then $u_i = max\{0, v_i - \ b_j\}$.

1. If $v_i \leq b_j$:

   better to lose!

2. If $v_i > b_j$:

   you should bid higher than b_j to win and there is no difference if you bid less or higher than v_i

Mechanism Design without Money

1. Stable matchings
2. Kidney exchange
3. Facility location
4. Distortion
5. Impartial selection
6. Strategyproof mechanism with Withholding agents
7. Fair division
8. Budget aggregation

Mechanism Design without Money

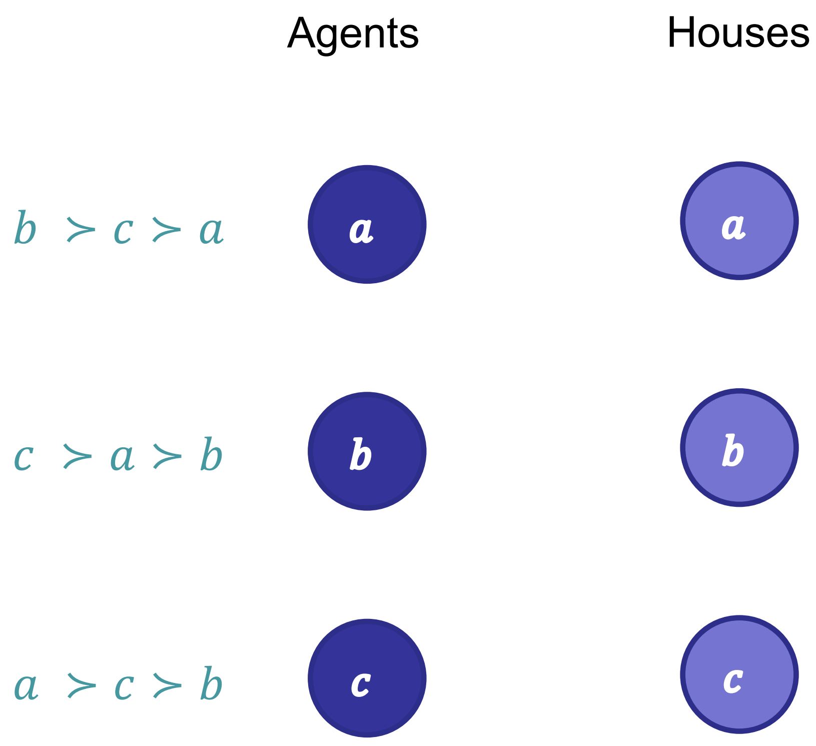
1. Stable matchings (Lecture 4)
2. Kidney exchange
3. Facility location (Lecture 3)
4. Distortion
5. Impartial selection
6. Strategyproof mechanism with Withholding agents
7. Fair division
8. Budget aggregation

House-Allocation Problem

Input
- A: set of agents
- Each agent brings an item to exchange
- Each agent has a preference list over houses

Output
- A permutation $\pi: A \to A$ between agents and houses

Goal
- Agents get happier



House-Allocation Problem

Input

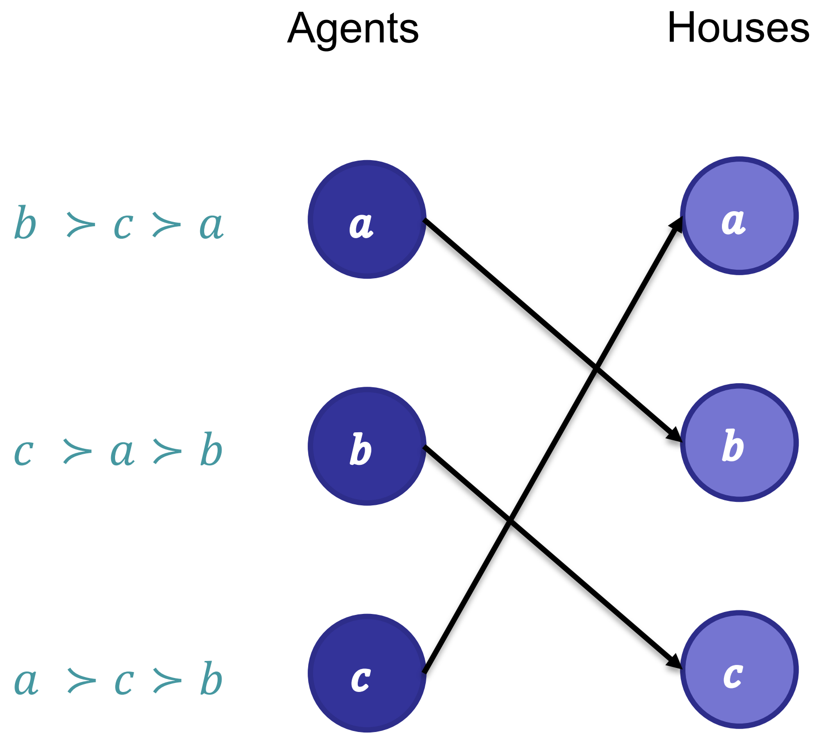
- A: set of agents
- Each agent brings an item to exchange
- Each agent has a preference list over houses

Output

- A permutation $\pi: A \rightarrow A$ between agents and houses

Goal

- Agents get happier



House-Allocation Problem

Input

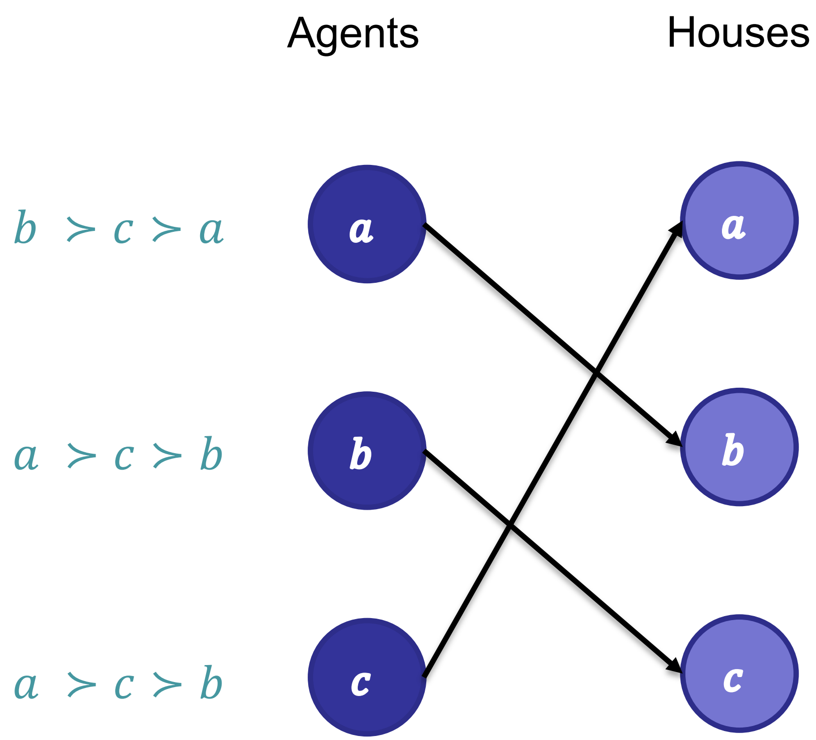
- A: set of agents
- Each agent brings an item to exchange
- Each agent has a preference list over houses

Output

- A permutation $\pi: A \to A$ between agents and houses

Goal

- Agents get happier



Unstable!

House-Allocation Problem

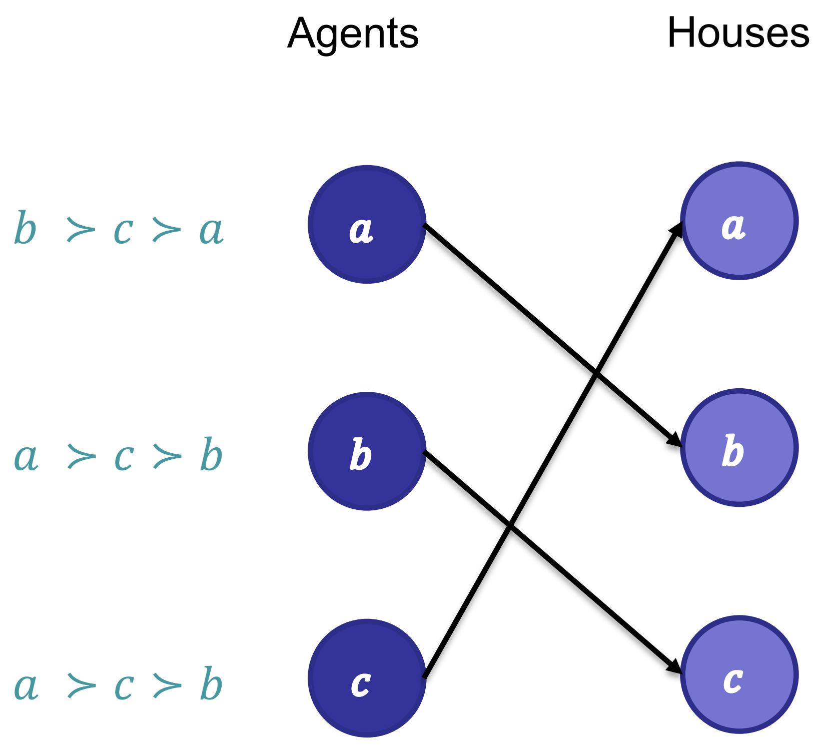
Blocking Coalition

- A subset of the agents $N \subseteq A$ is a blocking coalition of an allocation π, if there is an allocation $\sigma\colon N \to N$ such that

1. No $i \in N$ prefers $\pi(i)$ over $\sigma(i)$ and
2. At least one $i \in N$ prefers $\sigma(i)$ over $\pi(i)$.

Stable Allocation

- An allocation is stable if there is no blocking coalition.



House-Allocation Problem

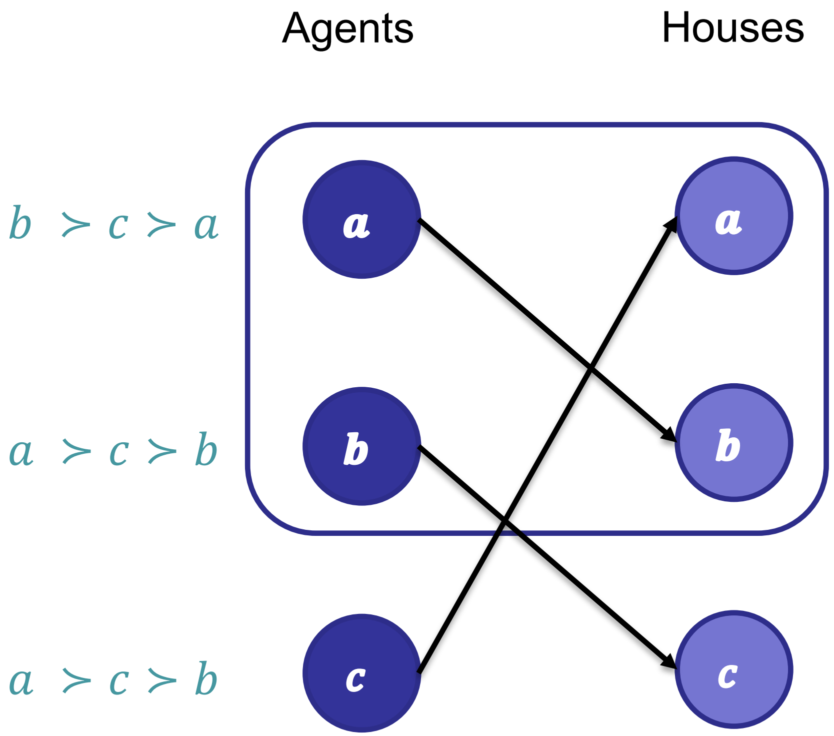
Blocking Coalition

- A subset of the agents $\boldsymbol{N} \subseteq \boldsymbol{A}$ is a blocking coalition of an allocation $\boldsymbol{\pi}$, if there is an allocation $\boldsymbol{\sigma}\colon \boldsymbol{N} \rightarrow \boldsymbol{N}$ such that

1. No $i \in N$ prefers $\boldsymbol{\pi}(\boldsymbol{i})$ over $\boldsymbol{\sigma}(\boldsymbol{i})$ and
2. At least one $i \in N$ prefers $\boldsymbol{\sigma}(\boldsymbol{i})$ over $\boldsymbol{\pi}(\boldsymbol{i})$.

Stable Allocation

- An allocation is stable if there is no blocking coalition.



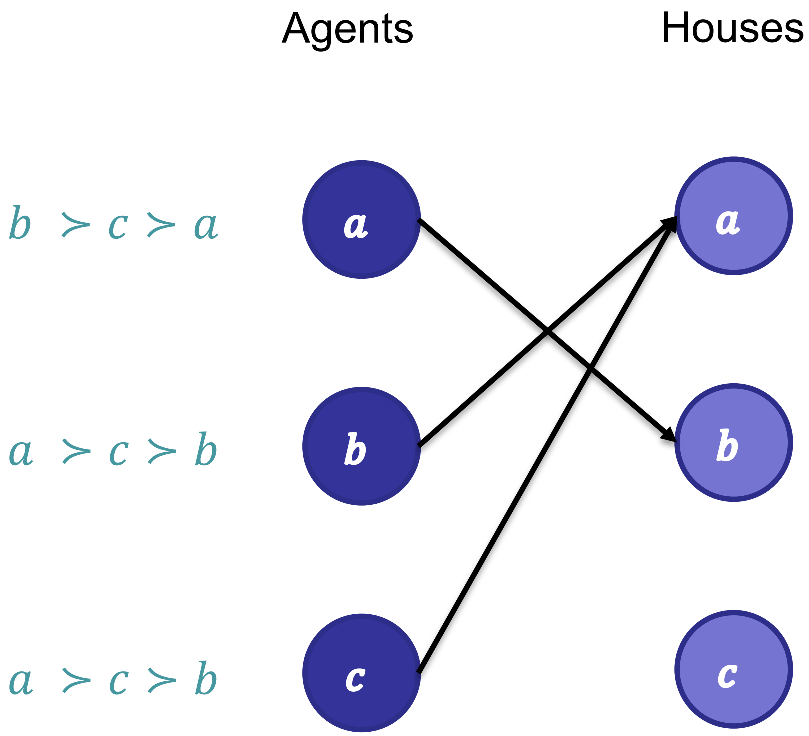
Unstable!

Top Trading Cycle Algorithm

There is always a stable allocation.

- Set $A' = A$
- While $A' \neq \emptyset$
  - Construct a directed graph on vertices A'.

    (Each vertex i has one outgoing edge to the owner of his preferred house among all houses owned by A'.)
  - Find an arbitrary **directed cycle** in this graph.

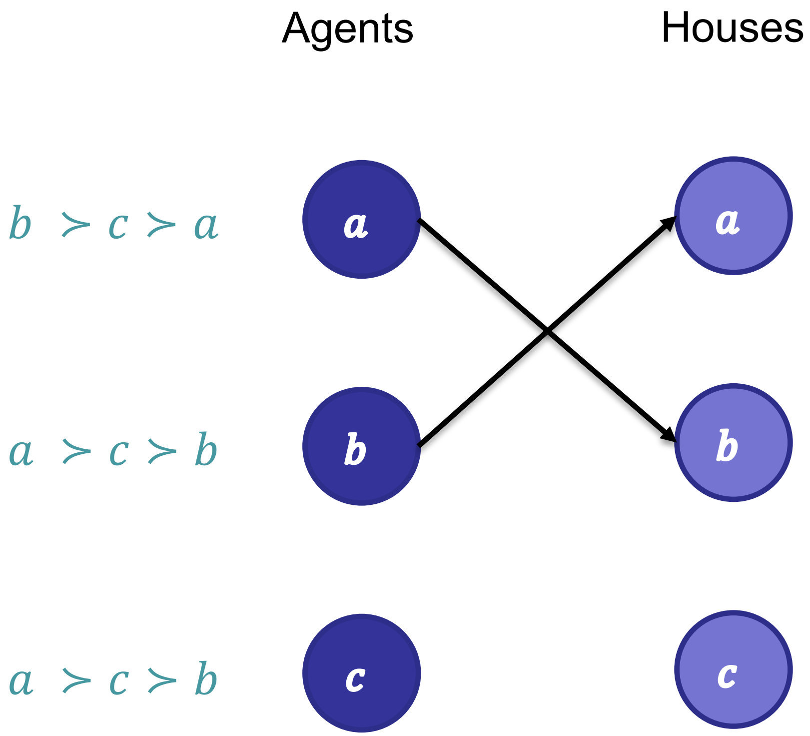


Top Trading Cycle Algorithm

There is always a stable allocation.

- Set $A' = A$
- While $A' \neq \emptyset$
  - Construct a directed graph on vertices A'.

  (Each vertex i has one outgoing edge to the owner of his preferred house among all houses owned by A'.)
  - Find an arbitrary **directed cycle** in this graph.

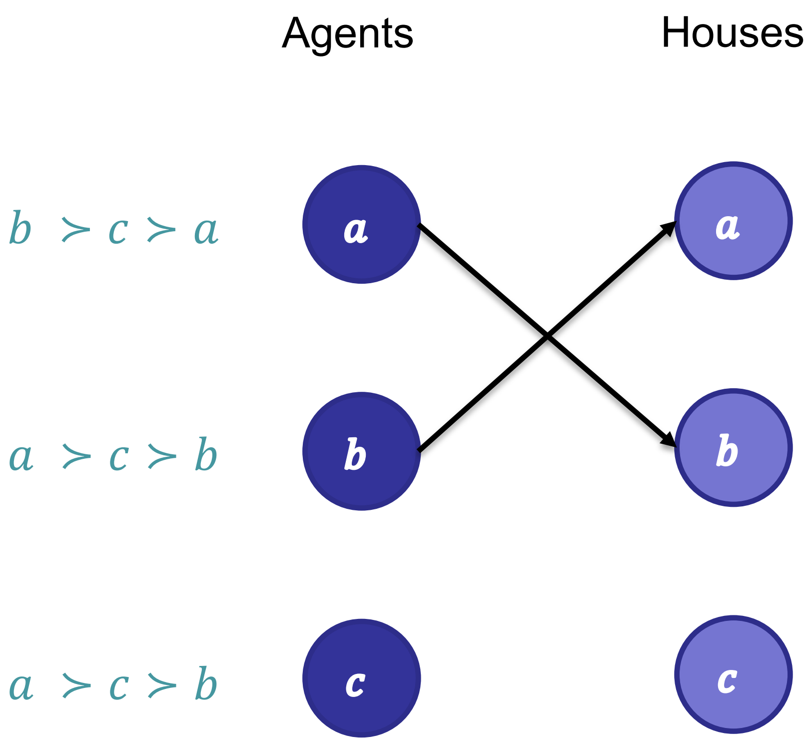


Top Trading Cycle Algorithm

There is always a stable allocation.

- Set $A' = A$
- While $A' \neq \emptyset$
  - Construct a directed graph on vertices A'.

    (Each vertex i has one outgoing edge to the owner of his preferred house among all houses owned by A'.)
  - Find an arbitrary **directed cycle** in this graph.
  - To each i in this cycle, assign the house owned by i's successor in the cycle.
  - Remove all agents in the cycle from A'.



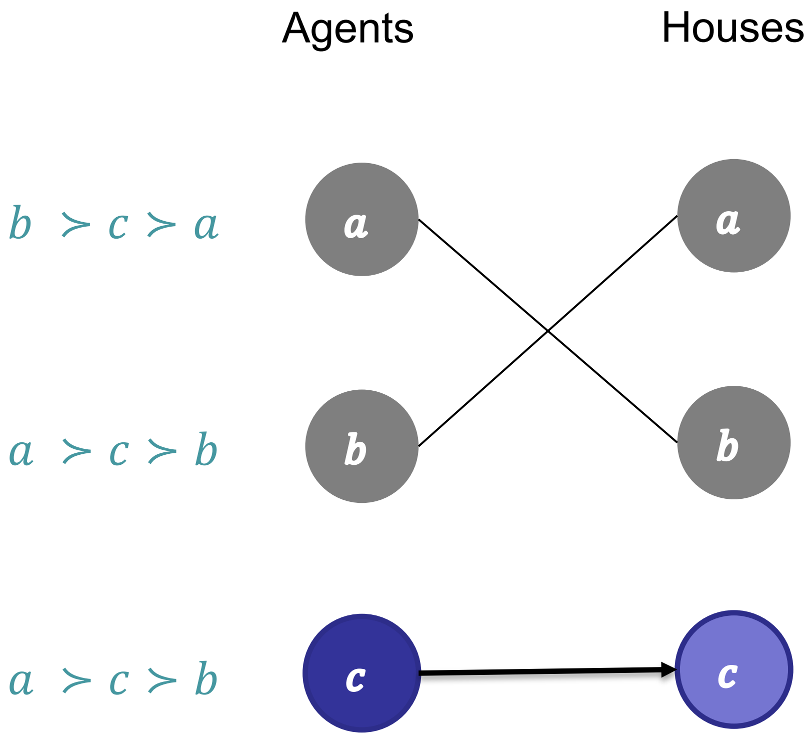
Top Trading Cycle Algorithm

There is always a stable allocation.

- Set $A' = A$
- While $A' \neq \emptyset$
  - Construct a directed graph on vertices A'.

    (Each vertex i has one outgoing edge to the owner of his preferred house among all houses owned by A'.)
  - Find an arbitrary **directed cycle** in this graph.
  - To each i in this cycle, assign the house owned by i's successor in the cycle.
  - Remove all agents in the cycle from A'.

* There always is a directed cycle in the constructed graph.



Top Trading Cycle Algorithm

The allocation outputted by the *Top Trading Cycles Algorithm* is

Stable, **Unique,** **Truthful**.

Kidney Exchange

Organ donations

- It is forbidden to ask for or pay money for these!
- Requires the tissues to be compatible
- What if all your relatives who would be willing and able to donate are incompatible?
- We might need chain of exchanges
- Top Trading Cycles Algorithm
- All surgeries have to take place simultaneously, close to each other
- We need short cycles

Related Papers

Stable matchings: Effective Affirmative Action in School Choice (27.05)

Kidney exchange: Mix and Match: A Strategyproof Mechanism for Multi-hospital Kidney Exchange (03.06)

Single Facility Location

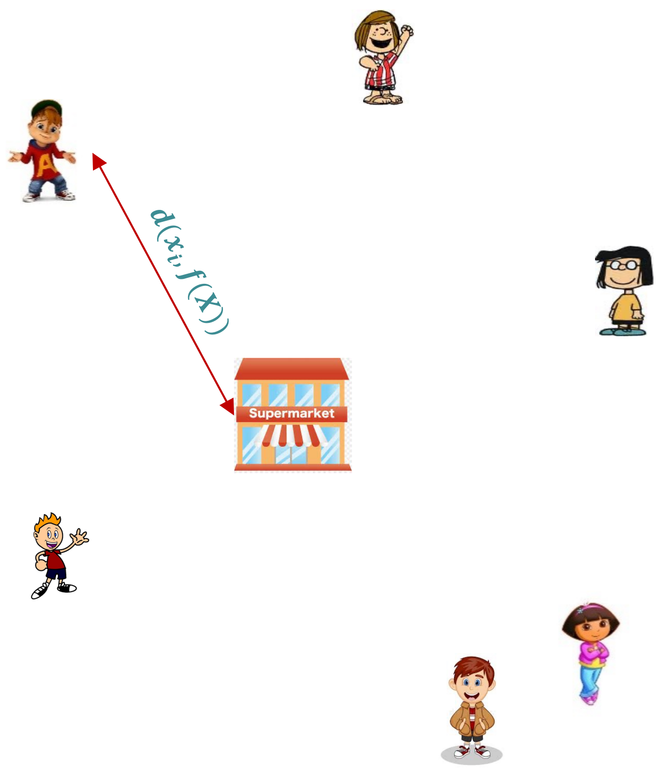
- $\boldsymbol{N}$: set of n strategic agents
- $\boldsymbol{X} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$: each agent i has a preferred location $\boldsymbol{x_i} \in \mathbb{R}^2$
- $\boldsymbol{f(X)} \in \mathbb{R}^2$: location of the facility
- $\boldsymbol{d(x_i, f(X)) \geq 0}$: Euclidean distance of agent i from the facility

- **Goal**: minimize some social cost objective
  - Egalitarian Social Cost:

$$\max_{i \in N} d(x_i, f(X))$$

  - Utilitarian Social Cost:

$$\sum_{i \in N} d(x_i, f(X))$$



Single Facility Location

- Goal: minimize $\max_{i \in N} d(x_i, f(X))$

OPT — D

Input:

$$X = \{x_1, \ldots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$

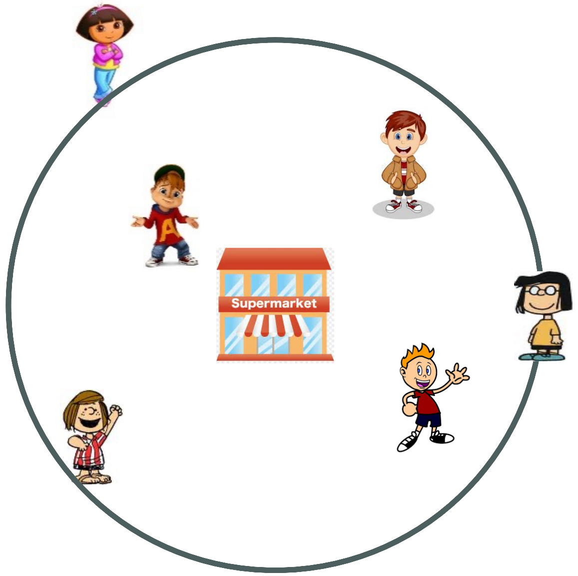
OPT □ D

Input:

$$X = \{x_1, \ldots, x_n\}$$

Output:

$$f(X) = \mathit{center\ of\ the\ minimum\ bounding\ circle}$$



Strategic Single Facility Location

- Strategyproof Mechanism:
  - Locations ($\boldsymbol{x_i} \in \mathbb{R}^2$) are **private** information
  - No agent has an incentive to misreport her location

OPT

D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$

Strategic Single Facility Location

- Strategyproof Mechanism:
  - Locations ($\boldsymbol{x_i} \in \mathbb{R}^2$) are **private** information
  - No agent has an incentive to misreport her location

OPT

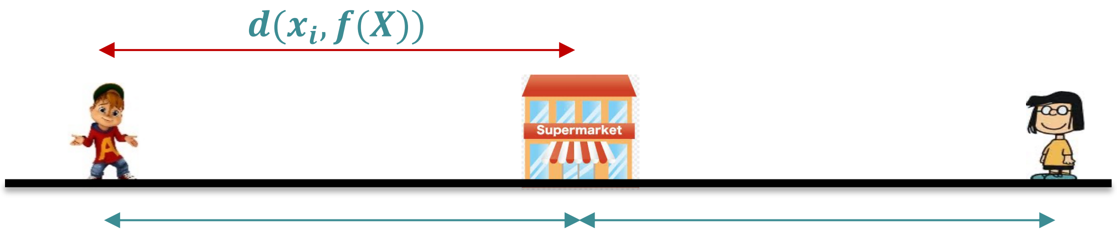
– D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$



Strategic Single Facility Location

- Strategyproof Mechanism:
  - Locations ($\boldsymbol{x_i} \in \mathbb{R}^2$) are **private** information
  - No agent has an incentive to misreport her location

OPT

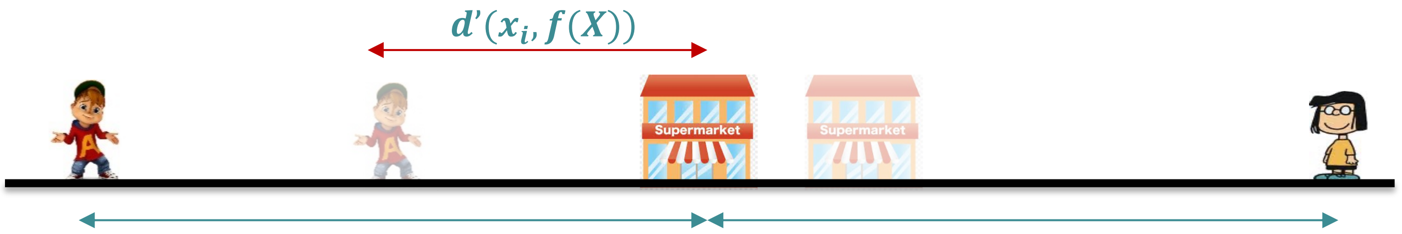
− D

Input:

$$X = \{x_1, \ldots, x_n\}$$

Output:

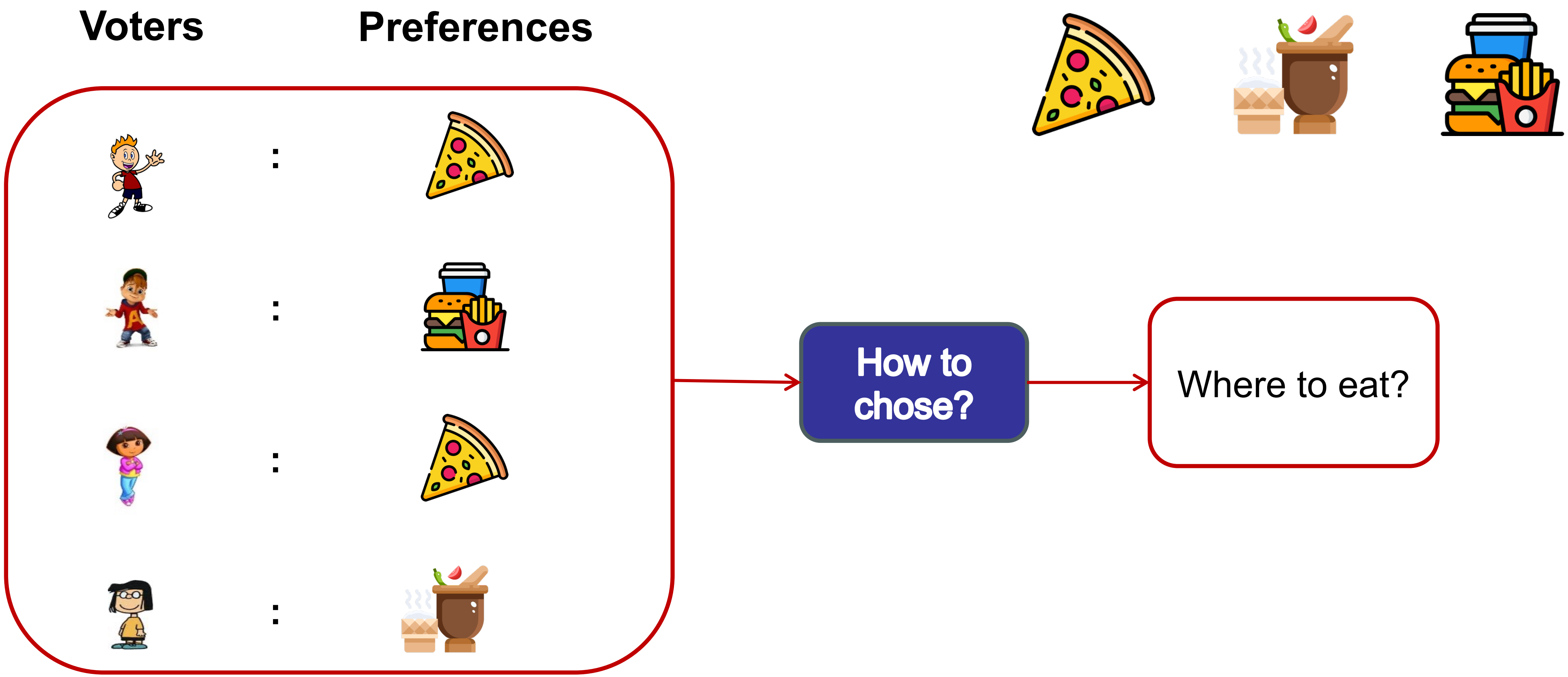
$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$



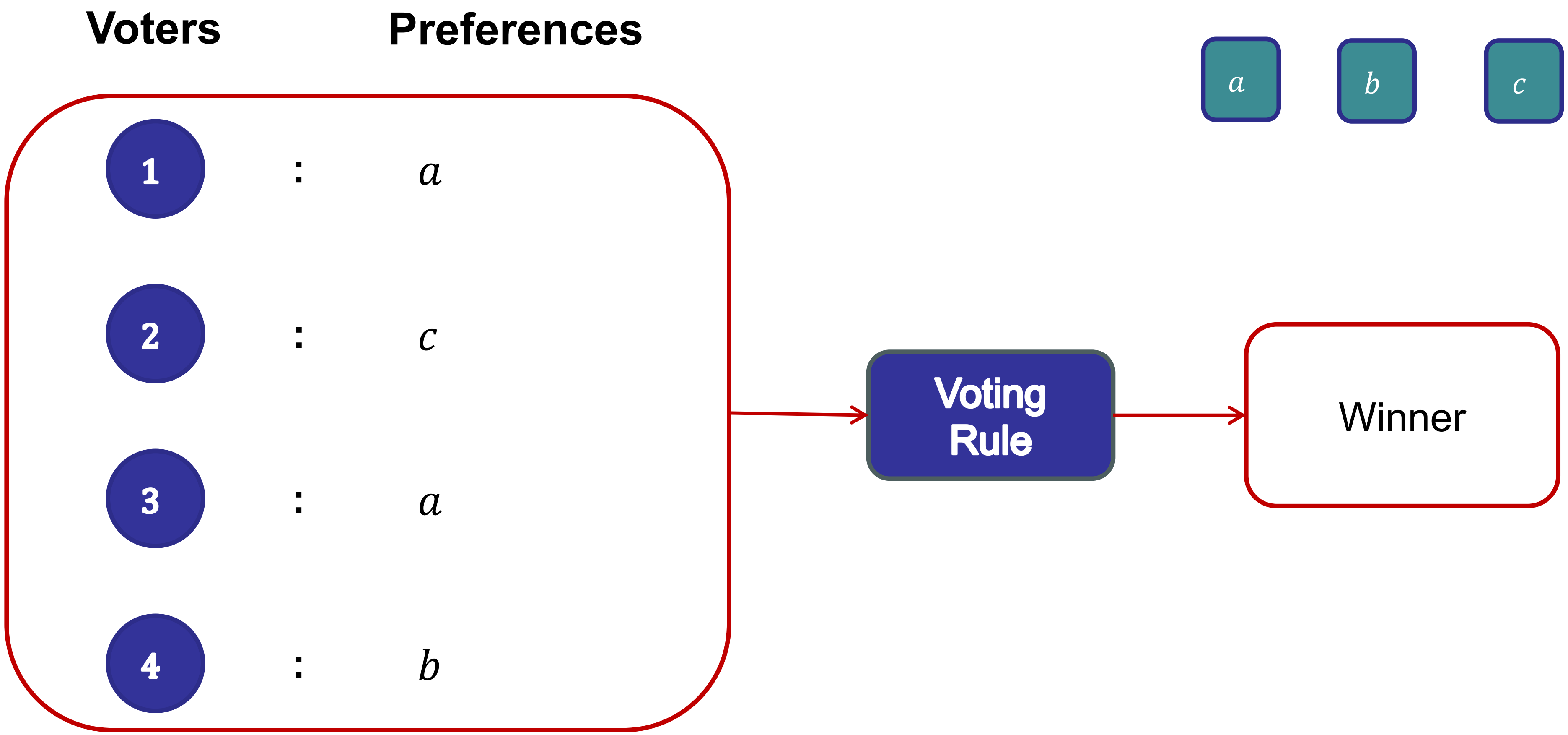
Related Paper

Learning-Augmented Mechanism Design: Leveraging Predictions for Facility Location

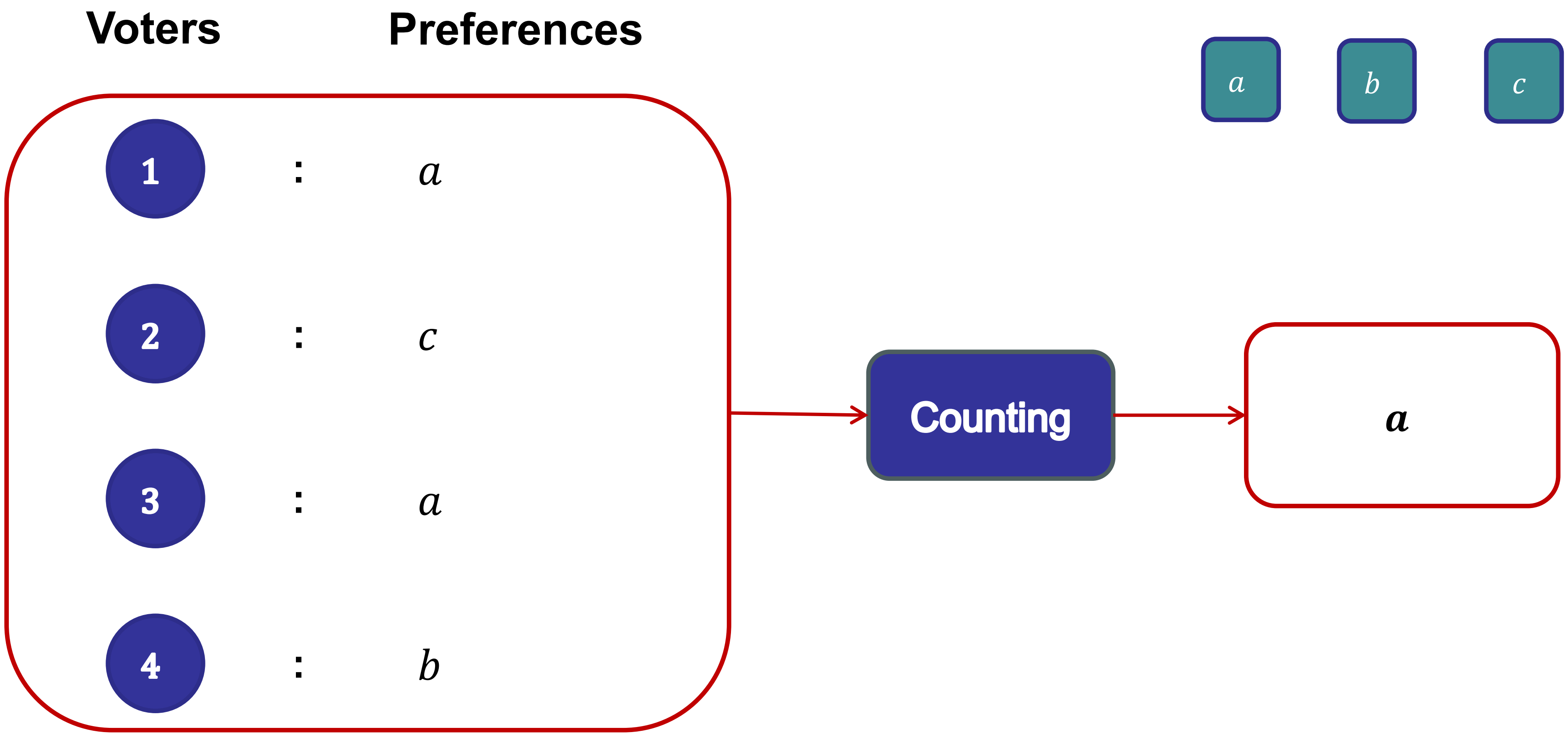
Voting



Top Choice Voting



Top Choice Voting



Ranked Voting

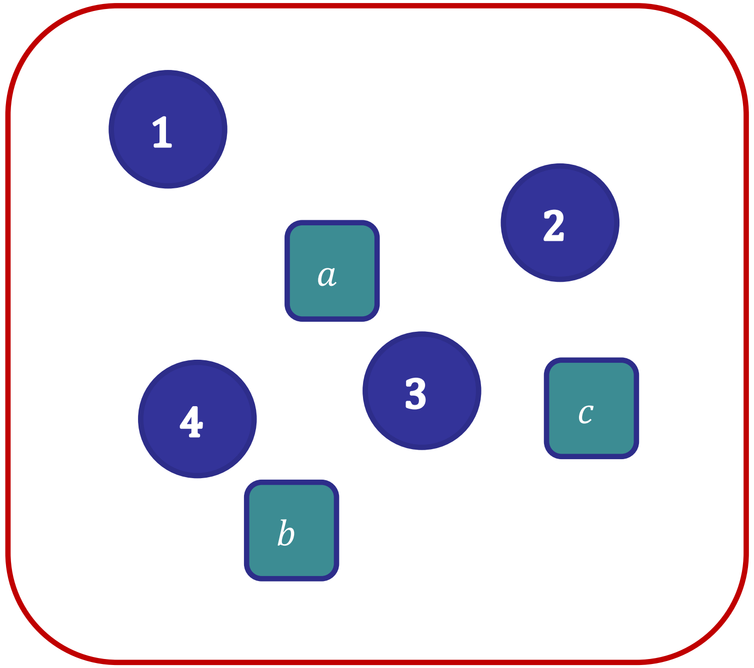
Voters		Preferences
1	:	$a \succ b \succ c$
2	:	$c \succ a \succ b$
3	:	$a \succ c \succ b$
4	:	$b \succ a \succ c$

a b c

Voting Rule → Winner?

Metric Distortion

Voters		Preferences
1	:	$a \succ b \succ c$
2	:	$c \succ a \succ b$
3	:	$a \succ c \succ b$
4	:	$b \succ a \succ c$



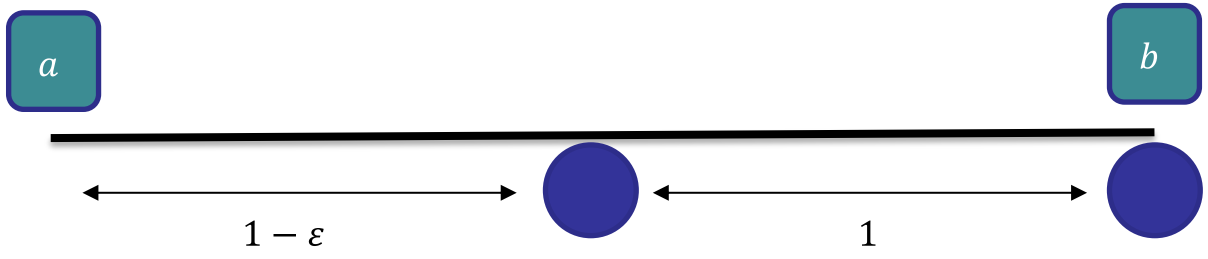
- Voters and Candidates lie in a metric space
- Voter’s cost for a candidate is their distance
- Distances satisfy triangle inequality
- The ranked list of each voter is consistent with the distances

Metric Distortion - Example

$\frac{n}{2}$ voters : $a \succ b$

$\frac{n}{2}$ voters : $b \succ a$

$\rightarrow$ $\boldsymbol{a}$



Cost(a) = $\frac{n}{2} + n$,

Cost(b) = $\frac{n}{2}$

Distortion = $sup_{all\ metrics} \frac{Cost\ (voting\ rule)}{optimal\ cost} = \frac{Cost\ (a)}{Cost\ (b)} = \frac{\frac{3n}{2}}{\frac{n}{2}} = 3$

Related Paper

Resolving the Optimal Metric Distortion Conjecture (17.06)

Fair Division

Cake-Cutting

How to fairly divide a cake among agents with differing preferences?

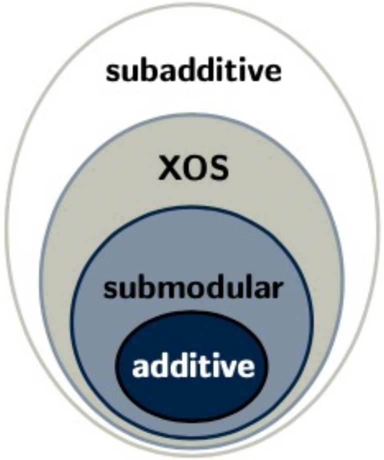
- A mother wants to divide a cake between her two kids
- She does not know what do the kids prefer
- She should come up with a mechanism!

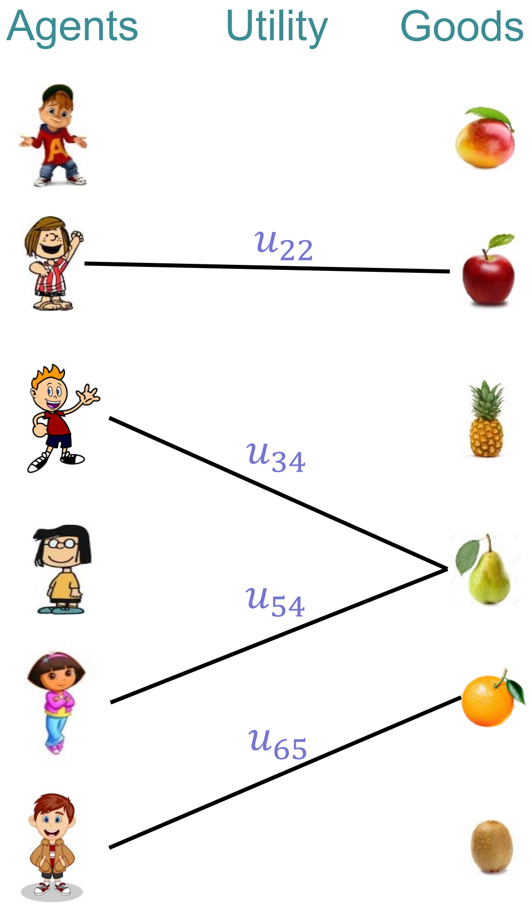
Cut-and-choose Protocol

- One kid cut and the other chose!

Fair Division

- N: set of n agents
- M: set of m divisible/indivisible goods
- u_{ij}: utility of agent i for a good g_j
- $u_i: 2^M \to R^+$: valuation functions

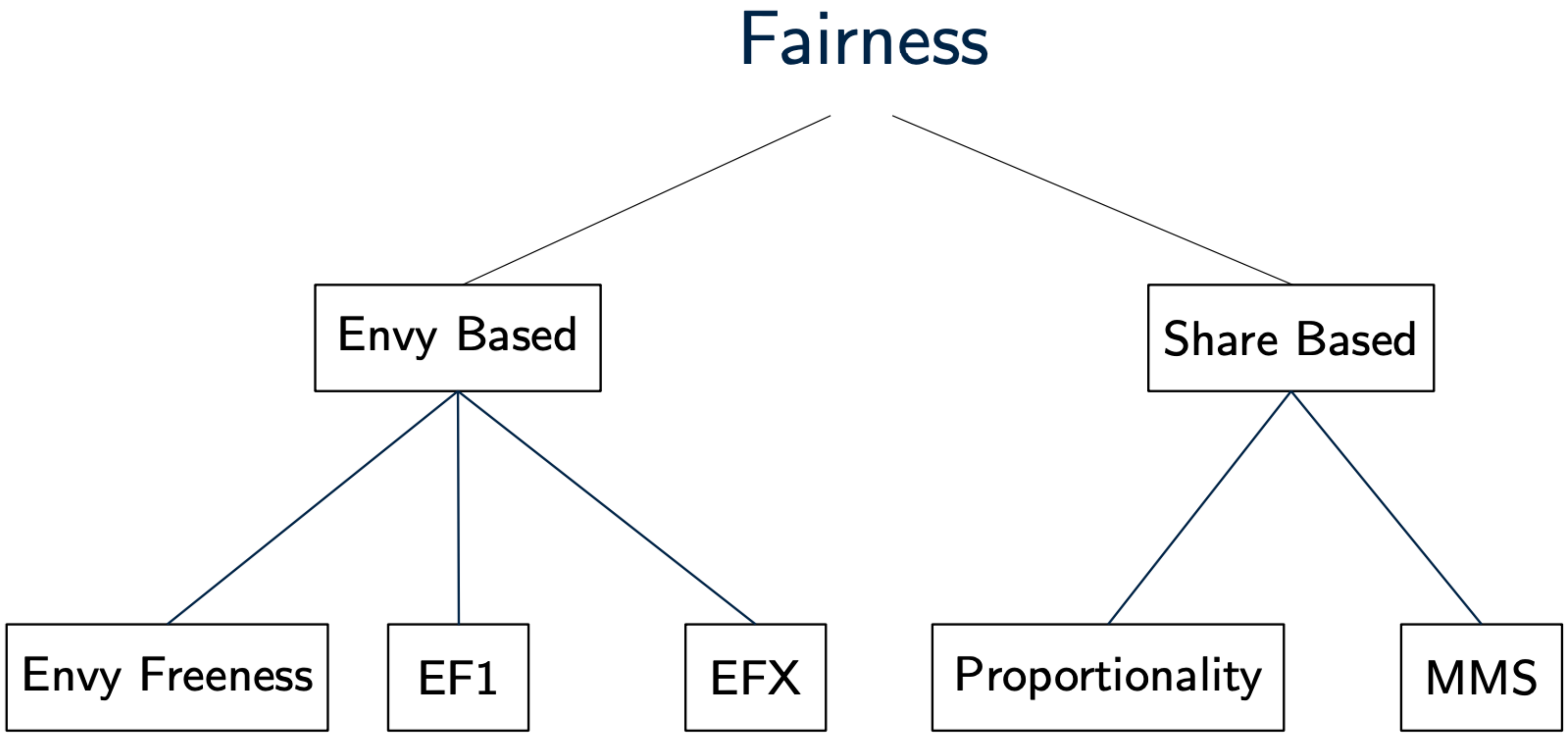




Goal

- Find a **fair** allocation of the goods to the agents.

Fair Division



Fair Division

- **EF**: An allocation X is envy free, if and only if for all agents a_i, a_j: $\boldsymbol{v_i(X_i) \geq v_i(X_j)}$
  - May not always exist.

- **EFX**: An allocation X is envy free up to any item or EFX, if and only if for all agents a_i, a_j, and for all goods $g \in X_j$: $\boldsymbol{v_i(X_i) \geq v_i(X_j \setminus \{g\})}$

Do complete EFX allocations always exist?

Related Paper

EFX: A Simpler Approach and an (Almost) Optimal Guarantee via Rainbow Cycle

Number (01.07)

Seminar Overview

1. **Stable matchings**: Effective Affirmative Action in School Choice (27.05)
2. **Kidney exchange**: Mix and Match: A Strategyproof Mechanism for Multi-hospital Kidney Exchange (03.06)
3. **Facility location**: Learning-Augmented Mechanism Design: Leveraging Predictions for Facility Location (10.06)
4. **Distortion**: Resolving the Optimal Metric Distortion Conjecture (17.06)
5. **Impartial selection**: Optimal Impartial Selection (24.06)
6. **Fair division**: EFX: A Simpler Approach and an (Almost) Optimal Guarantee via Rainbow Cycle Number (01.07)
7. **Budget aggregation**: Truthful Aggregation of Budget Proposals (08.07)
8. **Strategyproof mechanism with Withholding agents**: Truthful Assignment without Money (15.07)

Don’t forget!

Send us your preferred list of the student papers by **April 29th.**

Thank you!