Mechanism Design

By Marzie Nilipour

Spring 2023

Contents

- Social Choice & Stable Matching
- Auction Theory

Example

Adam

basketball > Walk > Karting

Karting

Bob

basketball > Walk > Karting

Basketball

Carl

Walk > Karting > basketball

Walk

David

Karting > Walk > basketball

Example

Adam

basketball > Walk > Karting

Bob

basketball > Walk > Karting

Carl

Walk > Karting > basketball

David

Karting > Walk > basketball

Karting

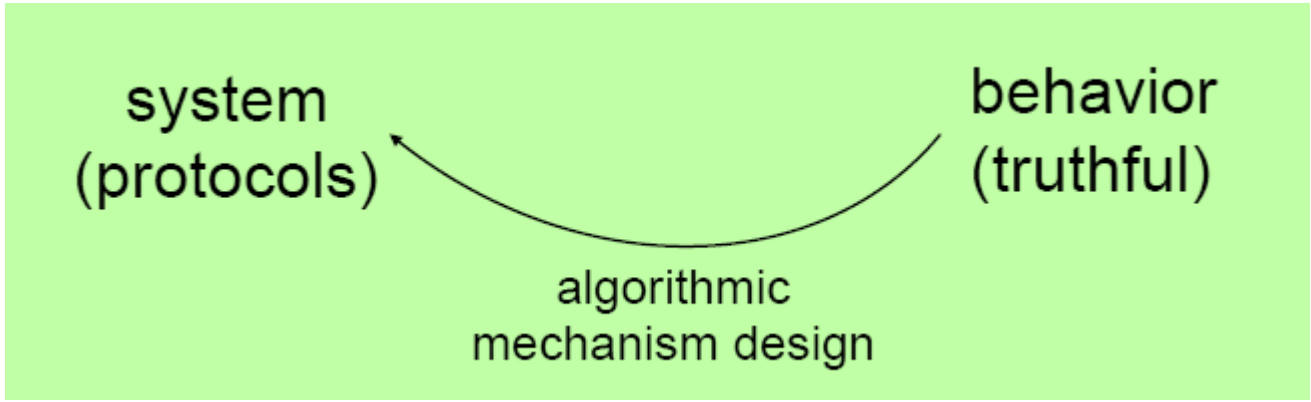
Basketball

Walk

selecting the activity with the highest number of votes, breaking ties **reverse alphabetically** **But David Can Cheat!!**

Algorithmic Mechanism Design

- Mechanism design
- Implementation theory
- It is sometimes called “reverse game theory”



Stable Marriage

Adam

Geeta, Heiki, Irina, Fran

Fran

Adam, Bob, Carl, David

Bob

Irina, Fran, Heiki, Geeta

Geeta

Carl, David, Bob, Adam

Carl

Geeta, Fran, Heiki, Irina

Heiki

Carl, Bob, David, Adam

David

Irina, Heiki, Geeta, Fran

Irina

Adam, Carl, David, Bob

Stable Marriage

Adam

Geeta, Heiki, Irina, Fran

Bob

Irina, Fran, Heiki, Geeta

Carl

Geeta, Fran, Heiki, Irina

David

Irina, Heiki, Geeta, Fran

Fran

Adam, Bob, Carl, David

Geeta

Carl, David, Bob, Adam

Heiki

Carl, Bob, David, Adam

Irina

Adam, Carl, David, Bob

It has many applications:

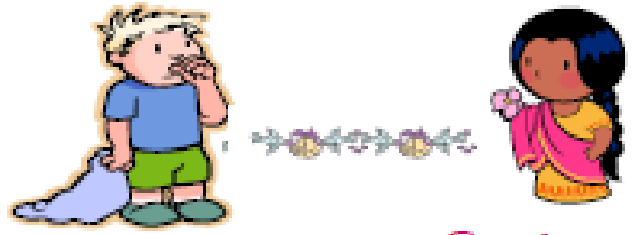
1.Dormitory rooms

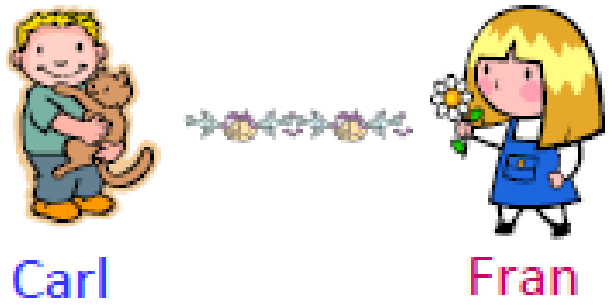
2.Apply

3.Job positions

Search For a Matching

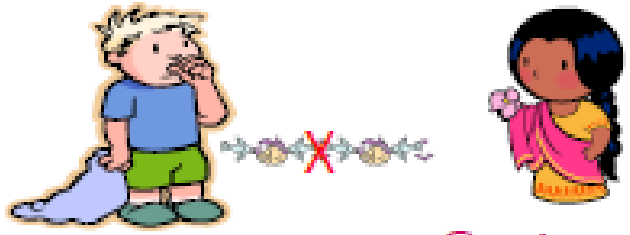
- Is this random marriage stable?



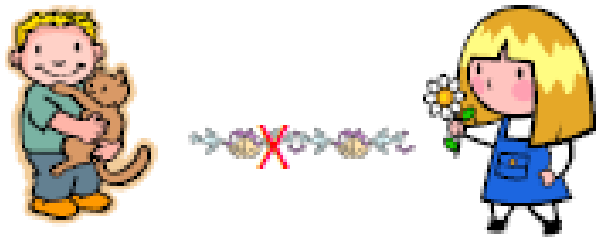


Search For a Matching

- Is this random marriage stable?
  - No



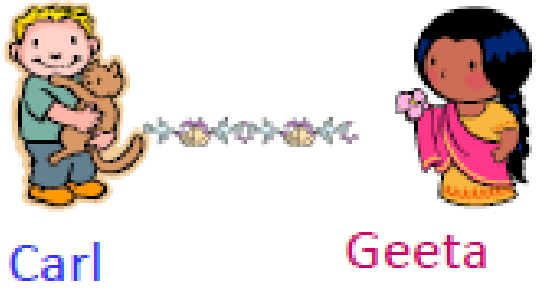
Geeta prefers Carl to Adam!

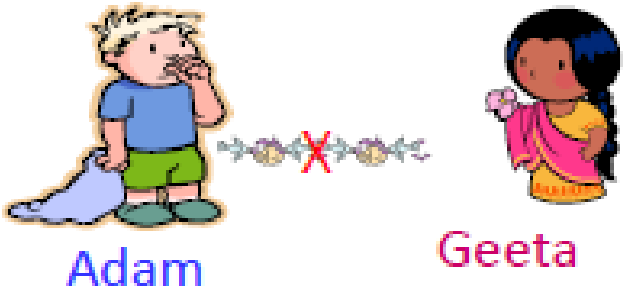


Carl likes Geeta better than Fran!

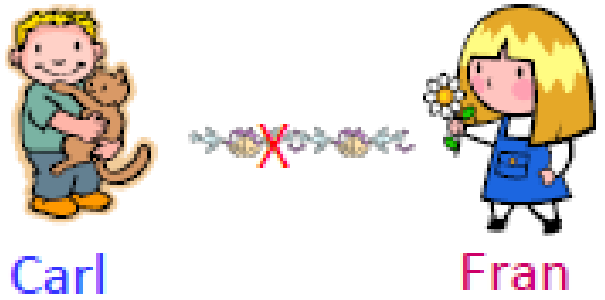
Search For a Matching

- Is this random marriage stable?
  - No
  - Because of some blocking pairs.





Geeta prefers Carl to Adam!



Carl likes Geeta better than Fran!

Stable Matching Definition

Stable matching

Is a matching without blocking pair.

A stable matching always exists, and the algorithmic problem solved by the Gale–Shapley algorithm is to find one.

Problem Definition

The stable matching problem takes as input
equal numbers of two types of participants
(*n* students and *n* internship position, for example),
and an preference list for whom to be matched to
among the participants of the other type.

Gale-Shapley Algorithm (1962)

- A stable matching always exists,
- This problem solved by the Gale–Shapley algorithm is to find one.
- Men propose, women accept/reject.

Gale-Shapley Algorithm (1962)

- A stable matching always exists,
- This problem solved by the Gale–Shapley algorithm is to find one.
- Men propose, women accept/reject.

- **Step 1:** *each student applies to his most preferred advisor.*
- **repeat**
  - **Step 2:** *each advisor keeps her most preferred acceptable application (if any) and rejects the rest (if any).*
  - **Step 3:** *each student who was rejected at the previous step applies to his next acceptable choice.*
- **until** *no student applied in the last step*

Cheating in Stable Matching

- Can men cheat?

- Can women cheat?

Cheating in Stable Matching

- Can men cheat?
  - No
  - For men, individually, being truthful is a dominant strategy
- Can women cheat?
  - Yes

Gale-Shapley Algorithm (1962)

- Runtime complexity: $O(n^2)$
  - n: number of participants of each type

Gale-Shapley Algorithm (1962)

- Runtime complexity: $O(n^2)$
  - n: number of participants of each type

- This algorithm guarantees that:
  - Everyone gets matched (At the end, there cannot be an applicant and employer both unmatched.
  - The matches are stable

Gale-Shapley Algorithm (1962)

- Runtime complexity: $O(n^2)$
  - n: number of participants of each type

- This algorithm guarantees that:
  - Everyone gets matched (At the end, there cannot be an applicant and employer both unmatched.
  - The matches are stable

- Implementation in software packages:
  - R: *matchingMarkets* and *matchingR* packages
  - Python: *matching* library, *MatchingMarkets.py* package

Stable Matching Applications

- Engineering space:
  - WiFi networks
  - Task Scheduling
  - Load-balancing for processors
  - Allocation problems

- Other spaces:
  - Students to schools or universities
  - Job-hunting

- Let’s go to Auctions!

Auction

- An auction is usually a process of buying and selling goods or services by offering them up for bids, taking bids,
- Then, depending on how the auction is defined, the winner is determined.

Auction

- An auction is usually a process of buying and selling goods or services by offering them up for bids, taking bids,
- Then, depending on how the auction is defined, the winner is determined.

- Types of auctions:
  - Single/multi buyer/seller,
  - Private/public value auction,
  - Forward auction,
  - Reverse auction,
  - Double auction,
  - …

Auction

- You compete with others, but you do not know the value of good for others.
- Formal notation:
  - True value of the good (V) that is unknown for all participants,
  - Estimation of participant i for good (v_i),
  - Bid of participant i for good (b_i).

  - Necessarily v_i is not the same with b_i

Payoff in Auction

$$\text{Payoff in the auction} = \begin{cases} V - b_i & \text{If you are the highest} \\ 0 & \text{Otherwise} \end{cases}$$

Your Estimate: $y_i = V + \varepsilon_i$

Main Lesson

- Main Lesson:
  - Bid as if you know you win, then you won't regret winning

Types of Auctions

A. First-price Auction
B. Second Price Auction
   - Winner pays the second bid (Vickery)
C. Ascending Open Auction (eBay)
D. Descending Open Auction

William Vickrey

Born	21 June 1914 Victoria, British Columbia, Canada
Died	11 October 1996 (aged 82) Harrison, New York, U.S.
Nationality	Canadian
Institution	Columbia University
Field	Public economics
Awards	Nobel Memorial Prize in Economics (1996)

First Price Auction

- In **the first-price auction** the payoff is

$$\begin{cases} v_i - b_i & \text{if win} \\ 0 & \text{o.w.} \end{cases}$$

  – Bidding your value is weakly dominated

$$b_i = v_i$$

Bonus: Proof of Dominance in Second Price Auction!

- In **second-price auction**
  - Value $\mathbf{v_i}$ and bid $\mathbf{b_i}$, the payoff is,

$$\begin{cases} \mathbf{v_i - \underline{b_j}} & \text{if } \mathbf{b_i} \text{ is the highest and } \underline{\mathbf{b_j}} \text{ is the highest other bid} \\ \mathbf{0} & \text{o.w.} \end{cases}$$

  - Setting $\mathbf{b_i = v_i}$ is weakly dominant