😀 Good morning !!!

STABLE MATCHING PROBLEM

Why are we interested

- **Multi agent system**: collection of multiple intelligent agents which interact
- **Ideally** we would like intelligent agents to be able to communicate and interact with other agents
- Building **societies of artificial agents** by taking **inspiration from groups of humans**

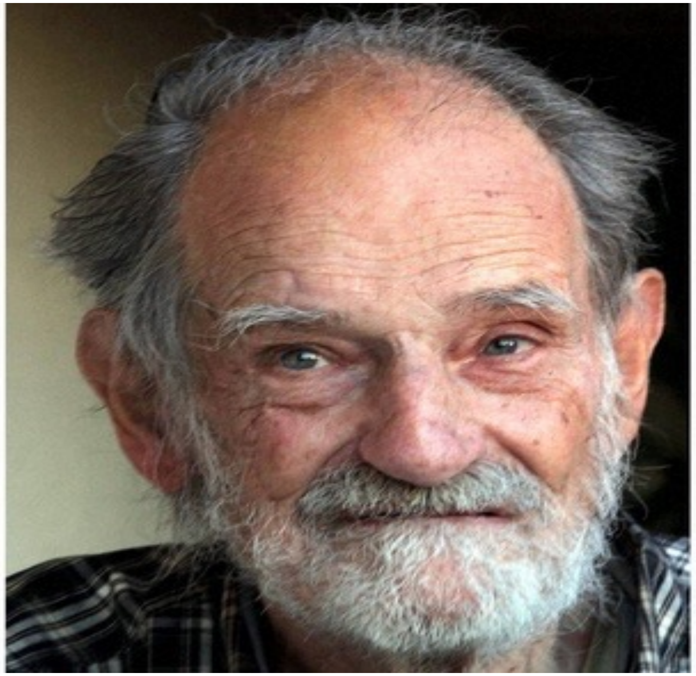
  → economics, game theory, and social choice

Chevaleyre, Endriss, Lang, Maudet, *A short introduction to Computational Social Choice*, 2007

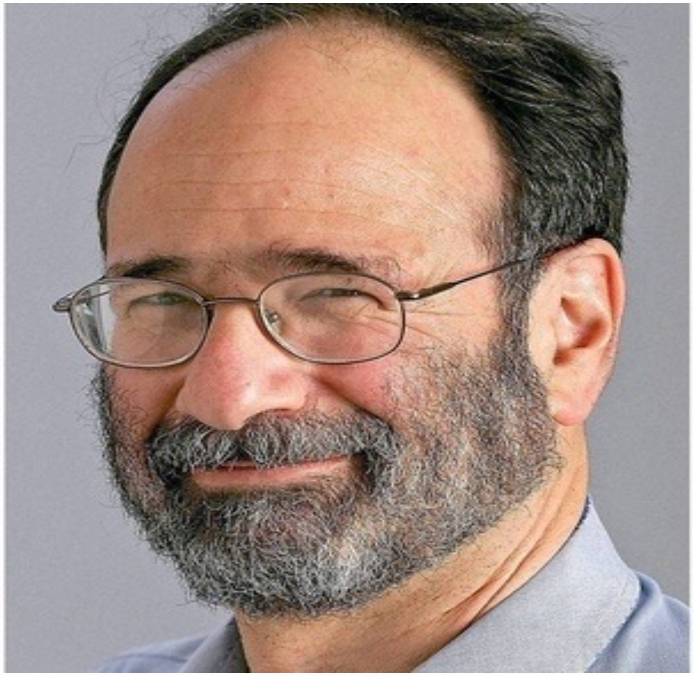
Computational Social choice

- Computational social choice: an **interdisciplinary field** at the interface of
  - Artificial intelligence
  - Economics
  - Voting theory
  - Game Theory
  - Social Choice
- Motivated mostly by the Internet
- We will see one problem in this area: **matching problems**, that are a mathematical abstraction of two-sided markets

Stable Matching theory won **Nobel Prize** in 2012



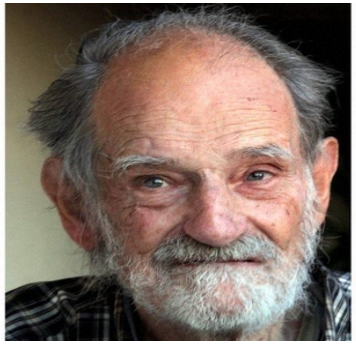
Lloyd Shapley

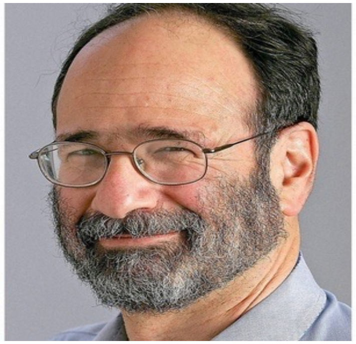


Alvin Roth

"for the **theory** of **stable allocations** and the **practice** **of market design**"

Stable Matching theory won Nobel Prize in 2012





Lloyd Shapley

He was Professor - University of California, Los Angeles

- ☐ Game theory

Alvin Roth

He is Professor of Economics - Stanford University

- ☐ Market design
- ☐ Mathematical models for strategic behaviour

Roth: **"I'm sure that when I go to class this morning my students will pay more attention"**

Matching under preferences

- Matching problems under preferences have been studied widely in
  - Mathematics
  - Computer science
  - Economics

  starting with the seminal paper by Gale and Shapley (1962)

Matching under preferences

- Matching problems with preferences occur in widespread applications such as assignment of
  - school-leavers to universities
  - junior doctors to hospitals
  - students to campus housing
  - children to schools
  - kidney transplant patients to donors
  - …

Matching under preferences

- The common thread is that
  - individuals have preferences over the possible outcomes and
  - the task is to find a matching of the participants that is in some sense optimal with respect to these preferences

Matching under preferences

- List of Topics
  - Two-sided matchings involving agents on both sides (e.g., college admissions, medical resident allocation, job markets, and school choice)
  - Two-sided matchings involving agents and objects (e.g., house allocation, course allocation, project allocation, assigning papers to reviewers, and school choice)
  - One-sided matchings (e.g., roommate problems, coalition formation games, and kidney exchange)
  - Other recent applications (e.g., refugee resettlement, food banks, social housing, and daycare)

Practical scenarios

- Matching students with schools
- Matching doctors with hospitals
- Matching kidney donors and patients
- Matching sailors to ships
- Job hunting
- …

Applications of Matching Models under Preferences.pdf https://www.matching-in-practice.eu

Matching under preferences

- **Matching in Practice**

  European network for research on matching practices in education and related markets

  Matching Practices in Europe for

  - Elementary Schools
  - Secondary Schools
  - Higher Education
  - Related Markets

Stable Matching Problems

- Two sets of agents
- Agents of one set express preferences over agents of the other set
- Goal: to <u>choose a matching</u> among the agents of the two sets based on their preferences
  - Matching: set of pairs (A1, A2), where
    - A1 comes from the first set
    - A2 comes from the second set

Stable Marriage formulation

- Two sets of agents: men and women
- Idealized model
  - Same number of men and women
  - All men **totally order** all women, and viceversa

Stable marriage problem

Two sets of agents:
{Greg, Harry, Ian}
{Amy, Bertha, Clare}

- Given preferences of n men
  - Greg: Amy>Bertha>Clare
  - Harry: Bertha>Amy>Clare
  - Ian: Amy>Bertha>Clare
- Given preferences of n women
  - Amy: Harry>Greg>Ian
  - Bertha: Greg>Harry>Ian
  - Clare: Greg>Harry>Ian
- Find a *stable* *marriage*

Stable marriage

- **Marriage:** is a one-to-one correspondence between men and women
  - ***Idealization:*** *everyone marries at the same time*

- **Stable Marriage:** a marriage with no pair (man, woman) not married to each other that would prefer to be together
  - ***Blocking pair*:**

    pair (m,w), where m is a man and w is a woman such that
    - *the marriage contains (m,w') and (m',w), but*
    - *m prefers w to w', and*
    - *w prefers m to m'*
  - ***Stable marriage*:** marriage with **no blocking pairs**
  - ***Idealization***: *assumes no cost in breaking a match*

An example of an **unstable** marriage

M = { (Greg,Clare), (Harry, Bertha), (Ian,Amy) }

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

Blocking pair:
makes the marriage not stable

- Amy: Harry > Greg > Ian
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Bertha & Greg would prefer to be together

An example of a **stable** marriage

M = { (Greg, Amy), (Harry, Bertha), (Ian, Clare) }

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

- Amy: Harry > Greg > Ian
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Men do ok, women less well

Another **stable** marriage

M = { (Greg, Bertha), (Harry, Amy), (Ian, Clare) }

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

- Amy: Harry > Greg > Ian
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Women do ok, men less well

Many stable marriages

- Given any stable marriage problem
  - There is **at least one** stable marriage
  - There may be **many** stable marriages especially in large AI domains

Gale Shapley algorithm

- **Initialize** every person to be **free**
- **While** exists **a free man**
  - **Find best woman** he has not proposed to yet
  - **If** this **woman is free**, declare them **engaged**
  - **Else**
    - **If** this **woman prefers <u>this proposal</u>** to her current partner then declare them **engaged** (and "free" her current partner)
    - **Else** this **woman prefers <u>her current partner</u>** and she **rejects the proposal**

Gale Shapley algorithm

- **Initialize** every person to be **free**
- **While** exists **a free man**
  - **Find best woman** he hasn't yet proposed to yet
  - **If this woman is free**, declare them **engaged**
  - **Else**
    - **If** this **woman** **prefers this proposal** to her current partner then declare them **engaged** (and "free" her current partner)
    - **Else** this **woman** **prefers her current partner** and she **rejects the proposal**

Two sets of agents:
{Greg, Harry, Ian}
{Amy, Bertha, Clare}

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

- Amy: Harry > Greg > Ian
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Gale Shapley algorithm

- Greg proposes to Amy, who accepts → (G,A)
- Harry proposes to Bertha, who accepts → (H,B)
- Ian proposes to Amy
- Amy is with Greg, and she prefers Greg to Ian, so she refuses
- Ian proposes to Bertha
- Bertha is with Harry, and she prefers Harry to Ian, so she refuses
- Ian proposes to Claire, who accepts → (I,C)

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

- Amy: Harry > Greg > Ian
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

M = { (Greg, Amy), (Harry, Bertha), (Ian, Clare) }

Gale Shapley algorithm

- Terminates with **everyone married**
- Terminates with a **stable marriage**

Gale Shapley algorithm

- Terminates with a **stable marriage**
  - **Suppose** there is a **blocking pair (m,w)** not married
    - Marriage contains (m,w') and (m',w)
    - m prefers w to w', and w prefers m to m'
  - **Case 1.** <u>m never proposed to w</u>
    - Not possible because men move down with the proposals, and w' is less preferred than w
  - **Case 2.** <u>m had proposed to w</u>
    - But w rejected him (immediately or later)
    - However, women only ever trade up
    - Hence w prefers m' to m
    - So the current pairing is stable

Other features of Gale Shapley algorithm

- Each of **n** men can make at most n proposals

  Hence GS runs in $O(n^2)$ time

- There may be **more than one** stable marriage

  - GS finds **man optimal solution**

    There is **no stable matching** in which any man **does better**

  - GS finds **woman pessimal solution**

    In all stable marriages, every woman does **at least as well or better**

Other stable marriages

- GS finds male-optimal (or female-optimal) marriage
- A set of agents is <u>favored</u> over the other one
- Other algorithms find "fairer" marriages
- Ex.: stable marriage which minimizes the maximum regret [Gusfield 1989]
  - **regret of a man/woman** = <u>distance</u> between his partner in the marriage and his most preferred woman/man

Example: SM formulation
3 rovers on a planet

- Are sent to a designated location and they have to perform an analysis
  - One drills
  - One takes pictures
  - One downlinks data

- **Two sets:**
  - {Rover1, Rover2, Rover3}
  - {Drill, Picture, Downlink}

Preferences of the 2 groups

- Rovers (e.g. preference of the rovers' managers)

  Rover1: Downlink>Picture>Drill

  Rover2: Picture>Downlink>Drill

  Rover3: Downlink>Picture>Drill

- Tasks (e.g. mission coordinator)

  Downlink: Rover2>Rover1>Rover3

  Picture: Rover1>Rover2>Rover3

  Drill: Rover1>Rover2>Rover3

Stable matching

- Find a *stable matching*
  - **Each rover** is assigned **a task**
    - *Idealization: everyone is matched at the same time*
  - **No blocking pairs:** (rover, task) not matched to each other would prefer to break their current matching and form a new one
    - *Idealization: assumes no cost in breaking a match*

An example of an unstable matching

M = { (Rover1, Downlink), (Rover2, Drill), (Rover3, Picture) }

Rover1: Downlink>Picture>Drill

Rover2: <u>Picture</u>>Downlink>Drill

Rover3: Downlink>Picture>Drill

Blocking pair:
makes the matching not stable

Downlink: Rover2>Rover1>Rover3

Picture: Rover1><u>Rover2</u>>Rover3

Drill: Rover1>Rover2>Rover3

Two stable matchings

M1 = { (Rover1, Downlink), (Rover2, Picture), (Rover3, Drill) }

Rover1: Downlink>Picture>Drill
Rover2: Picture>Downlink>Drill
Rover3: Downlink>Picture>Drill

Downlink: Rover2>Rover1>Rover3
Picture: Rover1>Rover2>Rover3
Drill: Rover1>Rover2>Rover3

Rover Optimal

M2 = { (Rover1, Picture), (Rover2, Downlink), (Rover3, Drill) }

Rover1: Downlink>Picture>Drill
Rover2: Picture>Downlink>Drill
Rover3: Downlink>Picture>Drill

Downlink: Rover2>Rover1>Rover3
Picture: Rover1>Rover2>Rover3
Drill: Rover1>Rover2>Rover3

Task Optimal

Review: Stable Marriage formulation

- Two sets of agents: men and women

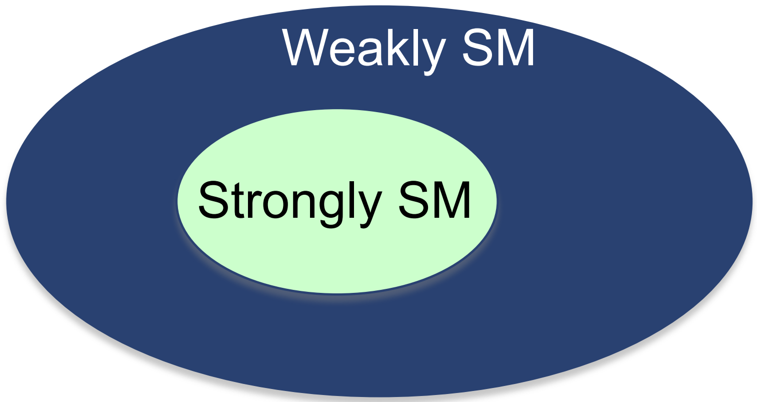
- Idealized model
  - Same number of men and women
  - All men **totally order** all women, and viceversa

Extensions: ties in preferences

- Eg.: A rover has **equally good** drill and camera
- **Preference orderings**: **total orders with ties**
- **Stability**
  - **weakly stable marriage:**

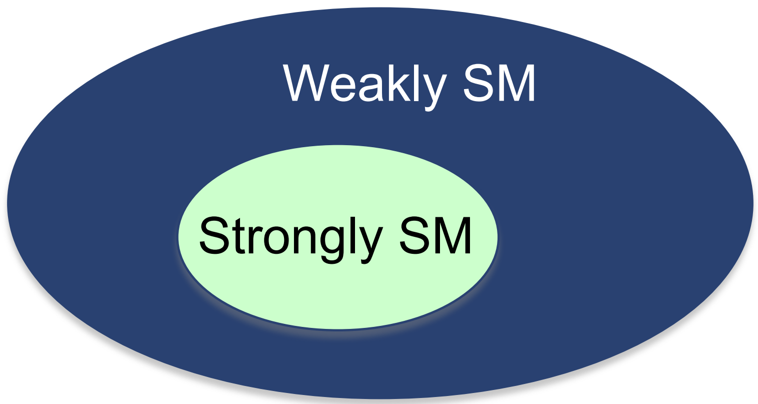
    no un-matched couple such that each one **strictly prefers** the other to the current partner
  - **strongly stable marriage:**

    no un-matched couple such that one **strictly prefers** the other, and the other **likes it as much or more** as the current partner



Extensions: ties in preferences

- **Existence**
  - **Strongly stable** marriage **may not exist**
    - $O(n^4)$ algorithm for deciding existence
  - **Weakly stable** marriage **always exists**
    - Just break ties arbitrarily
    - Run GS, resulting marriage is weakly stable!
    - → Polynomial complexity



Extensions: incomplete preferences

- Model **unacceptability of an option**
  - One of the Rovers **does not have a camera**

- **More** possible **blocking pairs**

- **(m,w)** blocking pair if
  - **m** and **w** are **unmatched** and
    do not find each other **unacceptable**, or
  - **m, w both prefer each other** to current partners, or
  - **one** of the two is **matched but acceptable** to the other and **prefers the other who is unmatched**

Extensions: incomplete preferences

- GS algorithm
  - Extends easily
  - → Polynomial complexity

- The **set of unmatched elements** is **the same** in every stable marriage

Extensions: ties & incomplete prefs

- **Weakly stable marriages** may have **different sizes**
  - Unlike with **just ties** where they are all complete
  - Or with **just incompleteness** where the cardinality is fixed

- Finding **weakly stable** marriage of **maximal cardinality** is NP-hard
  - Even if **only men** declare ties
  - Ties are of **most of length two**
  - The **whole list is a tie**

Strategy proofness

- GS is strategy proof (that is, non-manipulable) for men
  - Assuming male optimal algorithm
  - **No man can do better** than the male optimal solution

- However, women can profit from lying (that is, women can obtain a better partner by expressing different preferences from the true ones)
  - Assuming **male optimal algorithm** is run
  - Assuming **they know** complete **preference lists**

Manipulation by women

Greg: Amy > Bertha > Clare

Harry: Bertha > Amy > Clare

Ian: Amy > Bertha > Clare

Amy: Harry > Greg > Ian
Bertha: Greg > Harry > Ian
Clare: Greg > Harry > Ian

Result of running GS on **true prefs**

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

Amy lies

- Amy: Harry > Ian > Greg
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Result of running GS **on manipulated prefs**

Manipulation by women

- Greg proposes to Amy, who accepts
- Harry proposes to Bertha, who accepts
- Ian proposes to Amy, who accepts (Greg left alone)
- Greg proposes to Bertha, who accepts (Harry left alone)
- Harry proposes to Amy, who accepts (Ian left alone)
- Ian proposes to Bertha, who rejects
- Ian proposes to Clare, who accepts

- Greg: Amy > Bertha > Clare
- Harry: Bertha > Amy > Clare
- Ian: Amy > Bertha > Clare

Amy lies

- Amy: Harry > Ian > Greg
- Bertha: Greg > Harry > Ian
- Clare: Greg > Harry > Ian

Stable marriage obtained:

M={(Greg,Bertha), (Harry,Amy), (Ian,Clare)}

Impossibility of strategy-proofness

- GS can be manipulated

- Every stable marriage procedure (that is, every procedure that returns a stable marriage) can be manipulated if preference lists can be incomplete *[Roth '82]*

Impossibility of strategy proofness

Men: m1, m2
Women: w1, w2

- Consider
  - m1: w1 > w2   w1: m2 > m1
  - m2: w2 > w1   w2: m1 > m2

- **Two stable marriages:**
  - {(m1,w1), (m2,w2)}
  - {(m1,w2), (m2,w1)}

- **Suppose** we get male optimal solution
  - {(m1,w1), (m2,w2)}

Impossibility of strategy proofness

- Consider
  - m1: w1 > w2   w1: m2 > m1
  - m2: w2 > w1   w2: m1 > m2
- **Two stable marriages:**
  - {(m1,w1), (m2,w2)}
  - {(m1,w2), (m2,w1)}
- **Suppose** we get **male optimal solution**
  {(m1,w1), (m2,w2)}

- If woman **w1 lies** and says m1 is unacceptable
- Then we must get {(m2,w1), (m1,w2)} as this is **the only stable marriage**
- **Any procedure** that returns a stable matching can be manipulated if preference lists can be incomplete
- **Other cases** can be manipulated **in a similar way**

References for stable marriages

- Gale, Shapley. **College Admissions and the Stability of Marriage.** Amer. Math. Monthly, 69:9-14, 1981
- Roth. **The Economics of Matching: Stability and Incentives.** Mathematics of Operational Research, 7:617-628, 1982
- Gusfield, Irving. **The Stable Marriage Problem: Structure and Algorithms.** MIT Press, 1989
- Gale, Sotomayor. **Machiavelli and the stable matching problem.** Amer. Math. Monthly, 92:261-268, 1985
- Gusfield. **Three fast algorithms for four problems in stable marriage.** SIAM J. of Computing, 16(1), 1987
- Teo, Sethuraman, Tan. **Gale-Shapley Stable Marriage Problem Revisited: Stategic Issues and Applications.** Management Science, 47(9): 1252-1267, 2001
- Pini, Rossi, Venable, Walsh. **Manipulation complexity and gender neutrality in stable marriage procedures.** Autonomous Agents and Multi-Agent Systems, 22(1):183-199, 2011