UNIVERSITÀ DELLA CALABRIA

Stable Matching

Sebastiano A. Piccolo

Slides based on Ulle Endriss' lecture on stable matching for her course on Computational Social Choice

Matching

- Matching is about agents who have preferences over each other

Matching

- Matching is about agents who have preferences over each other
- Many applications:
  - Matching doctors to hospitals
  - Matching students to schools
  - Allocating people to tasks
  - Matching team members
  - Kidney exchange

Matching

- Matching is about agents who have preferences over each other
- Many applications:
  - Matching doctors to hospitals
  - Matching students to schools
  - Allocating people to tasks
  - Matching team members
  - Kidney exchange
- Relatively recent algorithmic approach to an economic problem

Matching

- Matching is about agents who have preferences over each other
- Many applications:
  - Matching doctors to hospitals
  - Matching students to schools
  - Allocating people to tasks
  - Matching team members
  - Kidney exchange
- Relatively recent algorithmic approach to an economic problem
- Nobel prize in economics for Shapley and Roth in 2012

Two-sided matching

- Two groups of agents have preferences over possible matchings between them

Two-sided matching

- Two groups of agents have preferences over possible matchings between them
- GOAL: finding a "good" matching

The Stable Marriage Problem

We are given:

- n men and n women
- Each agent has a linear preference ordering over the opposite sex

Our goal is:

- A stable matching of men to women
  - No man or woman want to divorce their assigned partners
  - We cannot reassign a couple to make them better off without making someone else worse off.

The Gale-Shapley Algorithm

Theorem (Gale and Shapley, 1962) There exists a stable matching for any combination of preferences of men and women.

The Gale-Shapley algorithm works as follows:

- In each round, each man who is not engaged proposes to his favourite amongst the women he has not yet proposed to.
- In each round, each woman choses her favourite amongst the proposals she receives and the man she is currently engaged to.

D. Gale and L.S. Shapley. College Admissions and the Stability of Marriage. American Mathematical Monthly, 69:9–15, 1962. [link]

The Gale-Shapley Algorithm: example

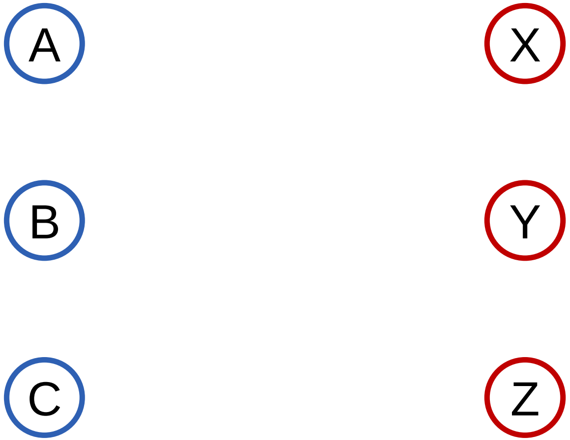
$$
\begin{aligned}
A &= [X, Y, Z] \\
B &= [Z, X, Y] \\
C &= [Z, Y, X]
\end{aligned}
$$

$$
\begin{aligned}
X &= [A, B, C] \\
Y &= [B, C, A] \\
Z &= [A, C, B]
\end{aligned}
$$

The Gale-Shapley Algorithm: example

$$
\begin{aligned}
A &= [X, Y, Z] \\
B &= [Z, X, Y] \\
C &= [Z, Y, X]
\end{aligned}
$$

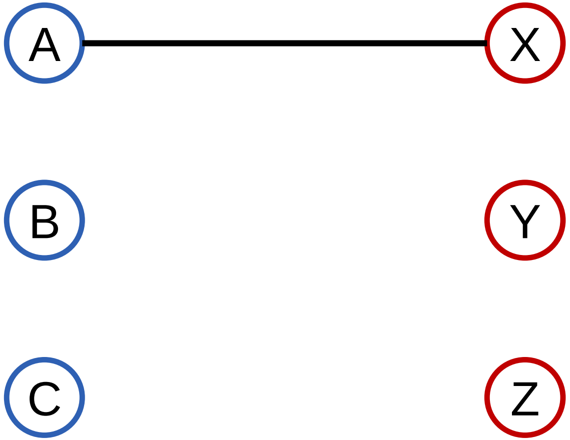
$$
\begin{aligned}
X &= [A, B, C] \\
Y &= [B, C, A] \\
Z &= [A, C, B]
\end{aligned}
$$



The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Z, X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

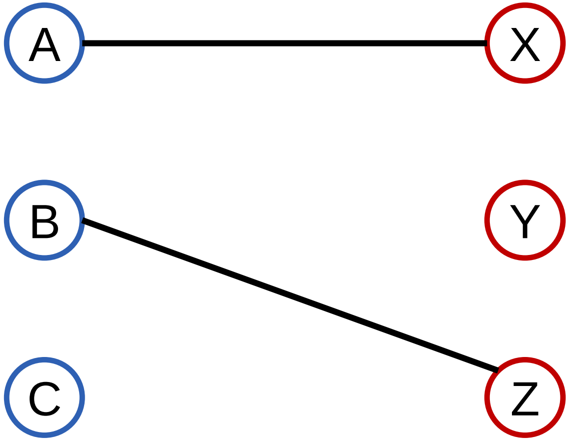


A proposes to X

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Z, X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

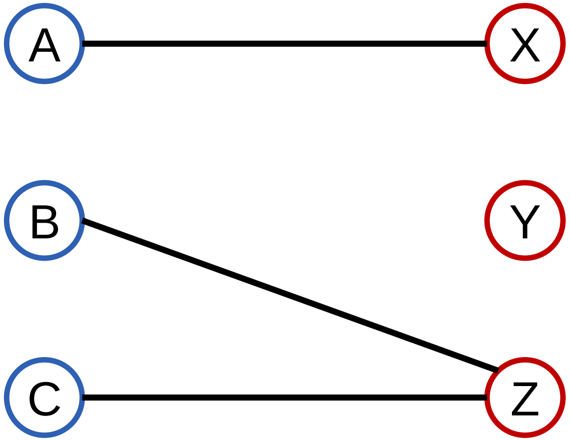


B proposes to Z

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Z, X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

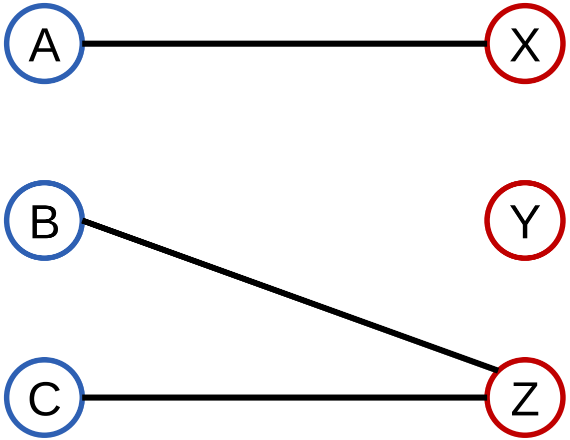


C proposes to Z

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Z, X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

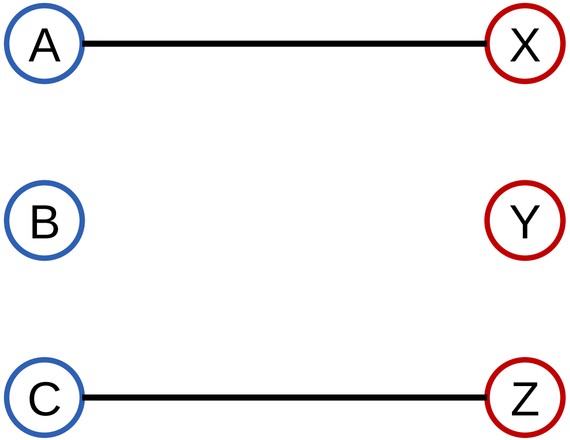


Z likes C more than B

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

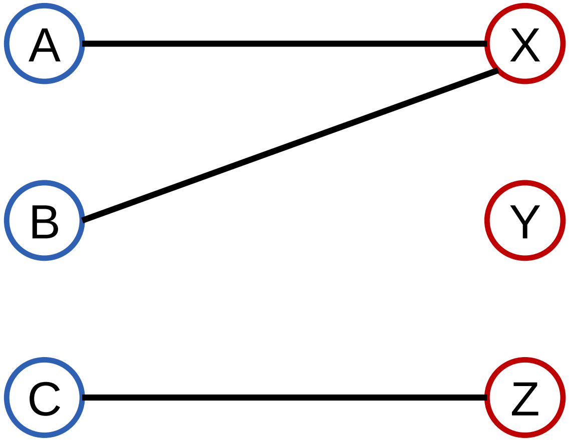


Z rejects B and matches with C

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [X, Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

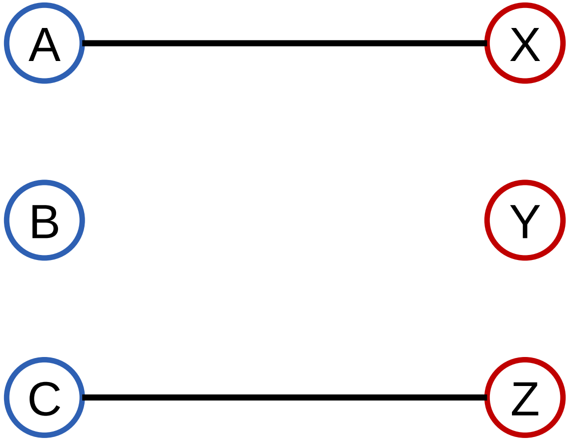


B proposes to X

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

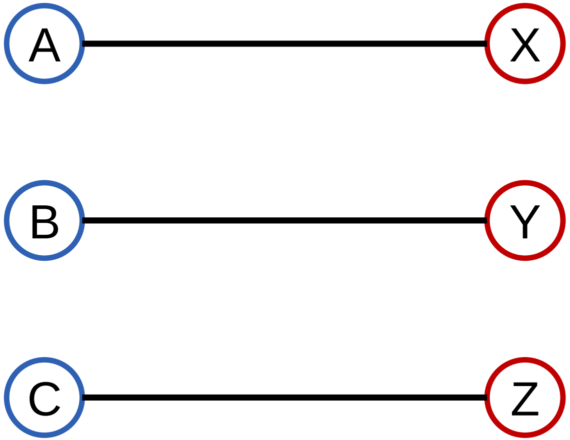


X prefers A and rejects B

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$

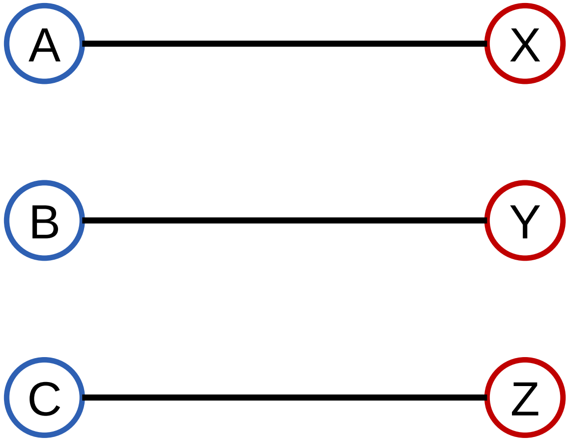


B proposes to Y

The Gale-Shapley Algorithm: example

$$A = [X, Y, Z]$$
$$B = [Y]$$
$$C = [Z, Y, X]$$

$$X = [A, B, C]$$
$$Y = [B, C, A]$$
$$Z = [A, C, B]$$



The algorithm terminates and we have a stable matching

Analysis

The Gale-Shapley algorithm:

- Always terminates
- Always returns a stable matching.
  - Otherwise the unhappy man should have proposed to the unhappy woman
- The algorithm has quadratic complexity

M-Optimal and W-Optimal matchings

- A stable matching is called M-Optimal if every man likes it at least as much as every other stable matching
- A stable matching is called W-Optimal if every woman likes it at least as much as every other stable matching

It can be shown that the matching returned by the Gale-Shapley algorithm (with men proposing) is M-Optimal, but it is not W-Optimal

M-Optimal and W-Optimal matchings

- A stable matching is called M-Optimal if every man likes it at least as much as every other stable matching
- A stable matching is called W-Optimal if every woman likes it at least as much as every other stable matching

It can be shown that the matching returned by the Gale-Shapley algorithm (with men proposing) is M-Optimal, but it is not W-Optimal

How to make it W-Optimal? Women should propose instead!