Stable Matchings

An Introduction and Basic Properties

A. Romer[1]

[1]Department of Mathematics
University of Victoria

Math 585

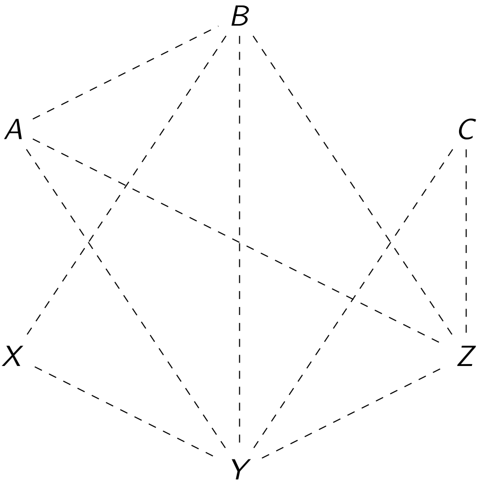
Table of Contents

What is a matching?

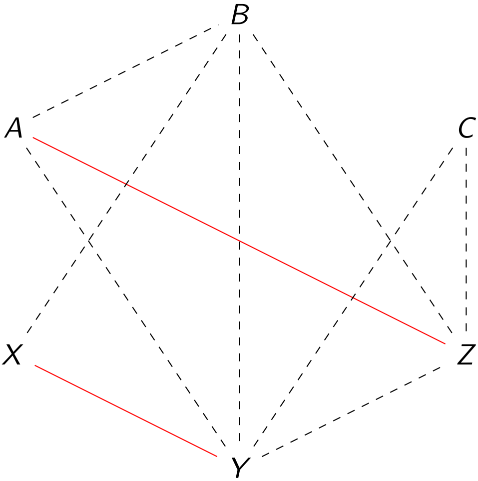
Definition

A *matching* is a set S of vertex disjoint edges in G.

What is a matching?



What is a matching?

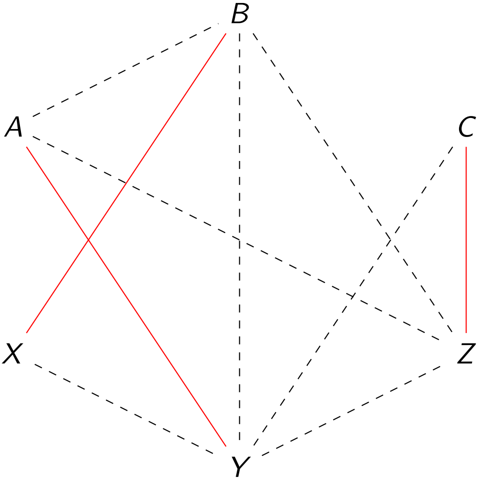


What is a matching?

Definition

A *perfect matching* is a matching that covers all the vertices of G.

What is a matching?



The Stable Matching Problem

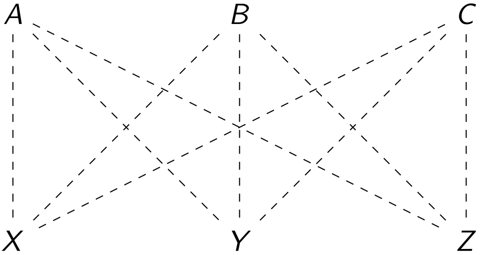
The Stable Matching Problem

Definition

A perfect matching M in a complete bipartite graph $K_{n,n}$ is *stable* if there are no two vertices $u \in X$, $v \in Y$ such that u prefers v over its current partner, v prefers u over its current partner, and $uv \notin M$.

The Stable Matching Problem

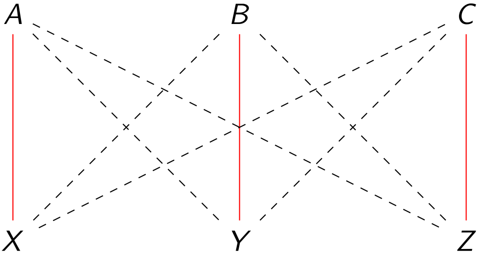
A: $\{x, y, z\}$ B: $\{y, z, x\}$ C: $\{y, z, x\}$



X: $\{c, b, a\}$ Y: $\{c, b, a\}$ Z: $\{b, a, c\}$

The Stable Matching Problem

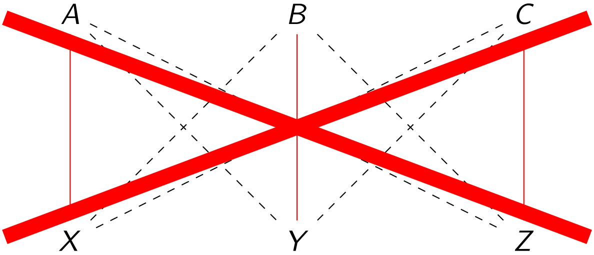
A: $\{x, y, z\}$ B: $\{y, z, x\}$ C: $\{y, z, x\}$



X: $\{c, b, a\}$ Y: $\{c, b, a\}$ Z: $\{b, a, c\}$

The Stable Matching Problem

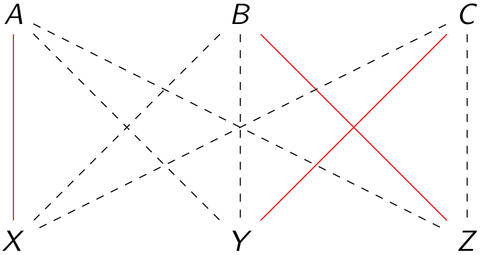
A: $\{x, y, z\}$ B: $\{y, z, x\}$ C: $\{y, z, x\}$



X: $\{c, b, a\}$ Y: $\{c, b, a\}$ Z: $\{b, a, c\}$

The Stable Matching Problem

A: $\{x, y, z\}$ B: $\{y, z, x\}$ C: $\{y, z, x\}$



X: $\{c, b, a\}$ Y: $\{c, b, a\}$ Z: $\{b, a, c\}$

Some Questions

1. When can we find a stable matching?

Some Questions

1. When can we find a stable matching?
2. What types of stable matchings can we find?

Some Questions

1. When can we find a stable matching?
2. What types of stable matchings can we find?
3. How are different stable matchings related?

The Gale-Shapley Algorithm

The Gale-Shapley Algorithm

1. All leads ask their preferred follower to dance. All followers who are asked accept the offer of their preferred person.

The Gale-Shapley Algorithm

1. All leads ask their preferred follower to dance. All followers who are asked accept the offer of their preferred person.
2. All leads who were not matched ask their second favorites. Again, followers accept the offer of their preferred person (possibly upgrading a previous matching).

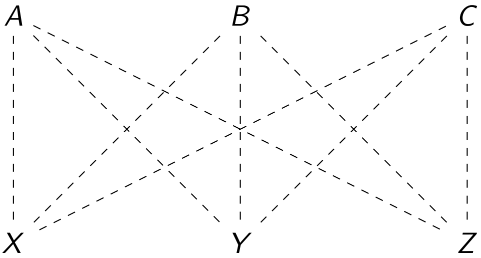
The Gale-Shapley Algorithm

1. All leads ask their preferred follower to dance. All followers who are asked accept the offer of their preferred person.
2. All leads who were not matched ask their second favorites. Again, followers accept the offer of their preferred person (possibly upgrading a previous matching).
3. Leads continue asking followers until every lead is either matched, or has exhausted his/her options.

The Gale-Shapley Algorithm

Example:

A: $\{x, y, z\}$ B: $\{x, y, z\}$ C: $\{y, z, x\}$

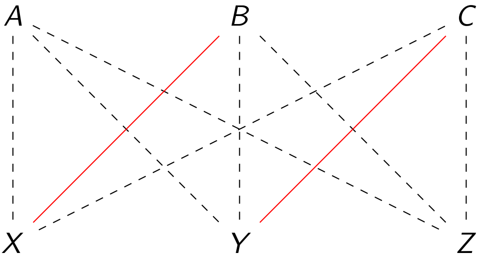


X: $\{b, c, a\}$ Y: $\{a, b, c\}$ Z: $\{b, a, c\}$

The Gale-Shapley Algorithm

Example:

A: $\{x, y, z\}$ B: $\{x, y, z\}$ C: $\{y, z, x\}$

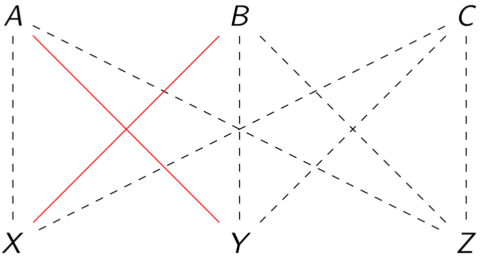


X: $\{b, c, a\}$ Y: $\{a, b, c\}$ Z: $\{b, a, c\}$

The Gale-Shapley Algorithm

Example:

A: $\{x, y, z\}$ B: $\{x, y, z\}$ C: $\{y, z, x\}$

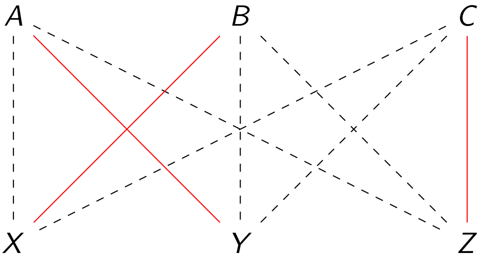


X: $\{b, c, a\}$ Y: $\{a, b, c\}$ Z: $\{b, a, c\}$

The Gale-Shapley Algorithm

Example:

A: $\{x, y, z\}$ B: $\{x, y, z\}$ C: $\{y, z, x\}$



X: $\{b, c, a\}$ Y: $\{a, b, c\}$ Z: $\{b, a, c\}$

The Gale-Shapley Algorithm

Proof of correctness:

1. The matching is perfect (everyone pairs up):

The Gale-Shapley Algorithm

Proof of correctness:

1. The matching is perfect (everyone pairs up):

2. No pairs are unstable:

The Gale-Shapley Algorithm

Proof of correctness:

1. The matching is perfect (everyone pairs up):
   a. Suppose there is an unmatched lead and follower when the algorithm ends. Then the lead must have proposed to every follower and been rejected (either immediately or through some later upgrade). This means that the unmatched follower must have received a proposition. But once a follower receives a proposition, he she or he will always be paired. Therefore, an unmatched pair is impossible.
2. No pairs are unstable:

The Gale-Shapley Algorithm

Proof of correctness:

1. The matching is perfect (everyone pairs up):
   a. Suppose there is an unmatched lead and follower when the algorithm ends. Then the lead must have proposed to every follower and been rejected (either immediately or through some later upgrade). This means that the unmatched follower must have received a proposition. But once a follower receives a proposition, he she or he will always be paired. Therefore, an unmatched pair is impossible.
2. No pairs are unstable:
   a. Suppose some matching of a lead l and follower f is unstable. Then l must prefer some other follower f', and f' must also prefer l over their current partner l'. If this is case, l must have asked f' before f. But followers can only reject a proposal if they then accept the proposal of someone else they prefer. This means that f' must in fact prefer l' over l. This is a contradiction. We conclude all matches are stable.

Properties of Stable Matchings

Observations

1. The Gale-Shapley algorithm provides a matching that gives the leads their top preferences.
2. By symmetry, allowing the followers to propose first gives them their top preferences.

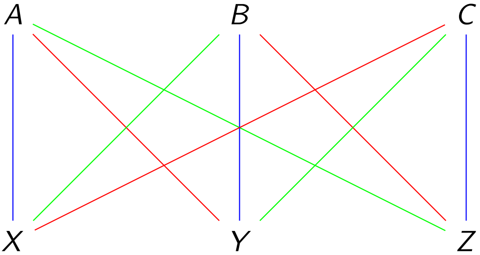
Properties of Stable Matchings

Observations

1. The Gale-Shapley algorithm provides a matching that gives the leads their top preferences.
2. By symmetry, allowing the followers to propose first gives them their top preferences.
3. Are there other intermediate matchings not produced by the GS algorithm? How are all solutions to a stable matching problem related?

Properties of Stable Matchings

A: $\{y, x, z\}$ B: $\{z, y, x\}$ C: $\{x, z, y\}$



X: $\{b, a, c\}$ Y: $\{c, b, a\}$ Z: $\{a, c, b\}$

{AY, BZ, CX} {AX, BY, CZ} {AX, BX, CY}

Properties of Stable Matchings

Partial ordering

For stable matchings $M, N \in S$, $M \leq N$ if and only if every lead prefers matching N to matching M. E.g. In the previous example, RED $\geq$ BLUE $\geq$ GREEN.

Properties of Stable Matchings

Partial ordering

For stable matchings $M, N \in S$, $M \leq N$ if and only if every lead prefers matching N to matching M. E.g. In the previous example, RED $\geq$ BLUE $\geq$ GREEN.

1. Reflexivity ($M \leq M$)

Properties of Stable Matchings

Partial ordering

For stable matchings $M, N \in S$, $M \leq N$ if and only if every lead prefers matching N to matching M. E.g. In the previous example, RED $\geq$ BLUE $\geq$ GREEN.

1. Reflexivity ($M \leq M$)
2. Anti-symmetry ($M \leq N$ and $N \leq M$ are not both true)

Properties of Stable Matchings

Partial ordering

For stable matchings $M, N \in S$, $M \leq N$ if and only if every lead prefers matching N to matching M. E.g. In the previous example, RED $\geq$ BLUE $\geq$ GREEN.

1. Reflexivity ($M \leq M$)
2. Anti-symmetry ($M \leq N$ and $N \leq M$ are not both true)
3. Transitivity (if $M \leq N$ and $N \leq L$, then $M \leq L$)

Properties of Stable Matchings

Partial ordering

RED $= \{AY, BZ, CX\} = \{1, 1, 1\}$

BLUE $= \{AX, BY, CZ\} = \{2, 2, 2\}$

GREEN $= \{AZ, BX, CY\} = \{3, 3, 3\}$

Properties of Stable Matchings

Lattice

We define two more operations on stable matchings: $\wedge$ and $\vee$:

Properties of Stable Matchings

Lattice

We define two more operations on stable matchings: $\wedge$ and $\vee$:

1. For $N, M \in S$, let $N \wedge M$ define the matching where each lead is given their *lowest* preference follower of those they are assigned in M and N.

Properties of Stable Matchings

Lattice

We define two more operations on stable matchings: $\wedge$ and $\vee$:

1. For $N, M \in S$, let $N \wedge M$ define the matching where each lead is given their *lowest* preference follower of those they are assigned in M and N.
2. For $N, M \in S$, let $N \vee M$ define the matching where each lead is given their *highest* preference follower of those they are assigned in M and N.

Properties of Stable Matchings

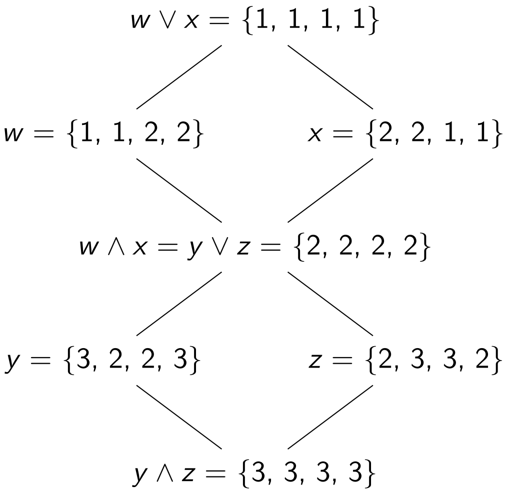
Lattice

We define two more operations on stable matchings: $\wedge$ and $\vee$:

1. For $N, M \in S$, let $N \wedge M$ define the matching where each lead is given their *lowest* preference follower of those they are assigned in M and N.
2. For $N, M \in S$, let $N \vee M$ define the matching where each lead is given their *highest* preference follower of those they are assigned in M and N.
3. In the case of matchings, $\wedge$ and $\vee$ are distributive. I.e. $N \wedge (M \vee L) = (N \wedge M) \vee (N \wedge L)$ and $N \vee (M \wedge L) = (N \vee M) \wedge (N \vee L)$.

Properties of Stable Matchings

Lattice



Properties of Stable Matchings

Distributed lattice

$$max = \{\}$$

$$min = \{\}$$

Properties of Stable Matchings

Lattice

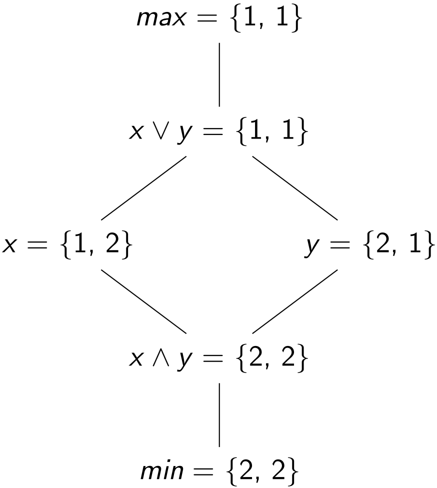
$$max = \{1\}$$

$$x = \{1\}$$

$$min = \{1\}$$

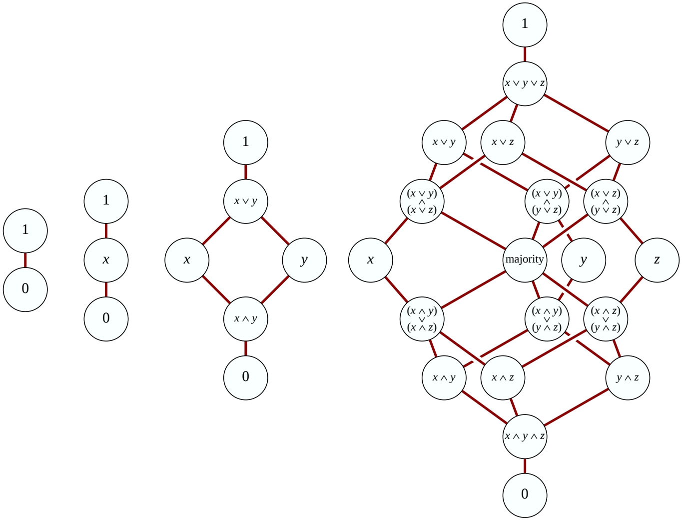
Properties of Stable Matchings

Lattice



Properties of Stable Matchings

Distributive lattice



Dedekind numbers

2
3
6
20
168
7,581
7,828,354
2,414,682,040,998
56,130,437,228,687,557,907,788
???

Thanks!