Social and Information Networks

University of Toronto CSC303
Winter/Spring 2024

Week 10: Mar 18-22

This week's high-level learning goals

- Define the Stable Marriage Problem
  - Define total and partial preferences
  - Define a two-sided matching market problem
  - Define a stable matching, and determine whether a given matching is stable
  - Recall and execute the Gale-Shapely algorithm
    - Prove it terminates
    - Prove it returns stable matchings
    - Define female-optimal and male-optimal matchings, and prove the relationship between the Gale-Shapely algorithm and these matchings
    - Recall how the Gale-Shapely algorithm is susceptible to manipulation
  - Define weak, strong, and superstrong stability, and explain how they extend the stable matching problem

Key terms

- Stable Marriage Problem
  - Preferences
  - Matching market problem
  - Gale-Shapely algorithm
    - Proof of termination
    - Proof of stability of produced matchings
    - Creation of optimal matchings
    - Manipulation
  - Extensions to the stable matching problem

New topic: The stable marriage problem

Note: This material is not in the text, but fits in nicely with the focus of CSC303

Namely, we will be concerned with graph matching but now restricted to bipartite graphs; this will also be lead us to another important example of a "coalition equilibrium"

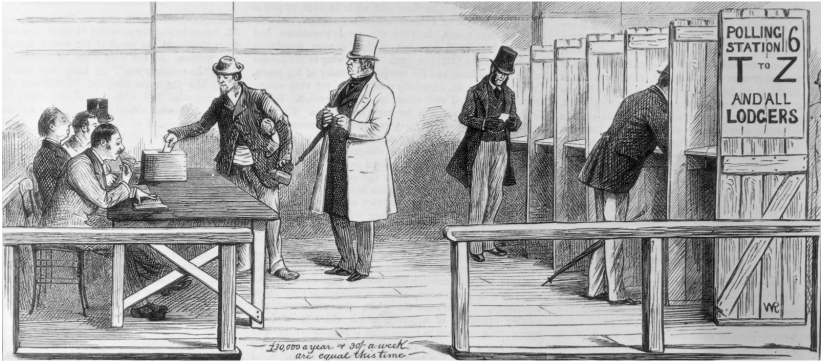
New topic: The stable marriage problem

Note: This material is not in the text, but fits in nicely with the focus of CSC303
Namely, we will be concerned with graph matching but now restricted to bipartite graphs; this will also be lead us to another important example of a "coalition equilibrium"

The stable marriage problem and the Gale Shapley algorithm, are interesting for a number of reasons:

- Mainly because it has practical application, and it is still actively considered due to variants arising from applications
- The algorithm is elegant and the analysis is interesting

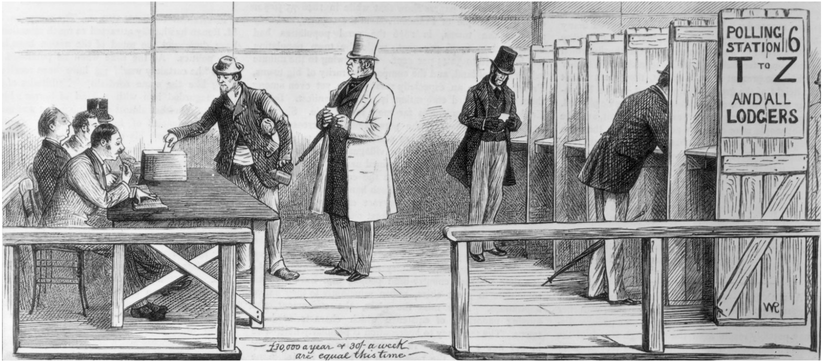
Preferences vs utilities



[Image from Encyclopedia Britannica]

- In game theory and mechanism design, individual valuations are typically numeric utilities (e.g., money)

Preferences vs utilities



[Image from Encyclopedia Britannica]

- In game theory and mechanism design, individual valuations are typically numeric utilities (e.g., money)
- In social choice theory (the study of the combination of interests, for example in voting rules) and in the stable marriage problem, individuals typically have preferences

Preferences vs utilities

Definition (Preference)

A *preference* over a set A of alternatives (e.g., candidates) is a total or partial order (also called an ordering or ranking) of the alternatives

Preferences vs utilities

Definition (Preference)

A *preference* over a set A of alternatives (e.g., candidates) is a total or partial order (also called an ordering or ranking) of the alternatives

- In many cases we find that it is difficult to place numeric values on alternatives, yet we know that we like alternative a_1 relative to alternative a_2

Preferences vs utilities

Definition (Preference)

A *preference* over a set A of alternatives (e.g., candidates) is a total or partial order (also called an ordering or ranking) of the alternatives

- In many cases we find that it is difficult to place numeric values on alternatives, yet we know that we like alternative a_1 relative to alternative a_2
  - e.g., ranking vs. assigning monetary values to the major political parties in an election

Preferences vs utilities

Definition (Preference)

A *preference* over a set A of alternatives (e.g., candidates) is a total or partial order (also called an ordering or ranking) of the alternatives

- In many cases we find that it is difficult to place numeric values on alternatives, yet we know that we like alternative a_1 relative to alternative a_2
  - e.g., ranking vs. assigning monetary values to the major political parties in an election
- For a set of alternatives $A = \{a_1, a_2, \ldots, a_n\}$, and an individual (say k), we use $\succ_k$ (or $\prec_k$) to denote k's preference between alternatives when k has such a preference

Preferences vs utilities

Definition (Preference)

A *preference* over a set A of alternatives (e.g., candidates) is a total or partial order (also called an ordering or ranking) of the alternatives

- In many cases we find that it is difficult to place numeric values on alternatives, yet we know that we like alternative a_1 relative to alternative a_2
  - e.g., ranking vs. assigning monetary values to the major political parties in an election
- For a set of alternatives $A = \{a_1, a_2, \ldots, a_n\}$, and an individual (say k), we use $\succ_k$ (or $\prec_k$) to denote k's preference between alternatives when k has such a preference
  - $a_i \succ_k a_j$ (alternatively $a_j \prec_k a_i$) if k definitely prefers a_i to a_j

Total orders vs partial orders

- If we're unsure about our preferences, we can use $a_i \succeq_k a_j$ to indicate that k likes a_i at least as much as a_j
- It could also the case that there are two alternatives for which we have no relative opinion

Total orders vs partial orders

- If we're unsure about our preferences, we can use $a_i \succeq_k a_j$ to indicate that k likes a_i at least as much as a_j
- It could also the case that there are two alternatives for which we have no relative opinion

- A *total order* $\succ$ on a set of alternatives $A = \{a_1, a_2, \ldots, a_n\}$ satisfies the following:
  - $\succ$ is transitive; that is, $a_i \succ a_j$ and $a_j \succ a_\ell$ implies $a_i \succ a_\ell$.
  - There is a permutation π such that $a_{\pi(1)} \succ_k a_{\pi(2)} \ldots \succ_k a_{\pi(n)}$.

Total orders vs partial orders

- If we're unsure about our preferences, we can use $a_i \succeq_k a_j$ to indicate that k likes a_i at least as much as a_j
- It could also the case that there are two alternatives for which we have no relative opinion

- A *total order* $\succ$ on a set of alternatives $A = \{a_1, a_2, \ldots, a_n\}$ satisfies the following:
  - $\succ$ is transitive; that is, $a_i \succ a_j$ and $a_j \succ a_\ell$ implies $a_i \succ a_\ell$.
  - There is a permutation π such that $a_{\pi(1)} \succ_k a_{\pi(2)} \ldots \succ_k a_{\pi(n)}$.
- A *partial order* $\succ$ satisfies the following:
  - $\succ$ is transitive
  - There is a way to extend the order (i.e., impose a preference on all a_i, a_j such that neither $a_i \succ a_j$ nor $a_j \succ a_i$ is given) so as to make $\succ$ into a total order
    - Missing preferences could mean various things (e.g., a lack of opinion, equivalence, etc...)

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y
- The goal is to match agents in X to agents in Y to satisfy some objective

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y
- The goal is to match agents in X to agents in Y to satisfy some objective
- Agents have the ability to leave unfavourable matches so as to obtain a more favourable match

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y
- The goal is to match agents in X to agents in Y to satisfy some objective
- Agents have the ability to leave unfavourable matches so as to obtain a more favourable match

Note: X and Y can be the same set in some applications. Possible Examples?

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y
- The goal is to match agents in X to agents in Y to satisfy some objective
- Agents have the ability to leave unfavourable matches so as to obtain a more favourable match

Note: X and Y can be the same set in some applications. Possible Examples? This was the situation in the study of network exchanges under the 1-exchange rule assumption. It is also the situation in a kidney exchange market.

Two-sided matching markets

In a two-sided matching market, we are interested in a matching in a graph/network where :

- There are two sets of agents X and Y
- The goal is to match agents in X to agents in Y to satisfy some objective
- Agents have the ability to leave unfavourable matches so as to obtain a more favourable match

Note: X and Y can be the same set in some applications. Possible Examples? This was the situation in the study of network exchanges under the 1-exchange rule assumption. It is also the situation in a kidney exchange market.

Note: As we remarked in our discussion of network exchanges, we are generally interested in b matchings in many applications where say agents (and in the bipartite case, maybe only agents on one side of the graph) can be involved in up to b edges. But for now, let us restrict our attention to the standard definition of a matching.

The bipartite case and the stable marriage problem

In the stable marriage problem, we are interested in matchings in a bipartite graph $G = (V, E)$ where $V = X \cup Y$. Furthermore, we assume that every agent X has a total preference order over Y and every Y has a total preference order over X

The bipartite case and the stable marriage problem

In the stable marriage problem, we are interested in matchings in a bipartite graph $G = (V, E)$ where $V = X \cup Y$. Furthermore, we assume that every agent X has a total preference order over Y and every Y has a total preference order over X

This total order assumption, and the restriction to matchings and not b-matchings, can be eliminated (say for the basic Gale-Shapley stable marriage algorithm) but they can present issues in some applications

Applications:

The bipartite case and the stable marriage problem

In the stable marriage problem, we are interested in matchings in a bipartite graph $G = (V, E)$ where $V = X \cup Y$. Furthermore, we assume that every agent X has a total preference order over Y and every Y has a total preference order over X

This total order assumption, and the restriction to matchings and not b-matchings, can be eliminated (say for the basic Gale-Shapley stable marriage algorithm) but they can present issues in some applications

Applications:

- Matching employees to specific positions (or tasks)
- Matching medical school graduates to specific residence positions
- The "classical" motivating example (i.e. from the early-60s U.S.A.) is matching men and women in marriages. We will stay with that terminology for consistency

The bipartite case and the stable marriage problem

In the stable marriage problem, we are interested in matchings in a bipartite graph $G = (V, E)$ where $V = X \cup Y$. Furthermore, we assume that every agent X has a total preference order over Y and every Y has a total preference order over X

This total order assumption, and the restriction to matchings and not b-matchings, can be eliminated (say for the basic Gale-Shapley stable marriage algorithm) but they can present issues in some applications

Applications:

- Matching employees to specific positions (or tasks)
- Matching medical school graduates to specific residence positions
- The "classical" motivating example (i.e. from the early-60s U.S.A.) is matching men and women in marriages. We will stay with that terminology for consistency
    - In 2020, The dating app Hinge claimed to use the Gale-Shapely algorithm
    - `https://www.vice.com/en/article/z3e3bw/how-does-the-hinge-algorithm-work`

Wed. Mar 22: Announcements and Corrections

- Assignment 2 is due tomorrow
- Your group's draft of the critical review project is due this Friday
- Does the amount of money being split affect solutions when bargaining?
  - At the very least, for the Ultimatum game, Anderson et al. (2011) "present evidence that as stakes increase, rejection rates approach zero"
  - `http://www.aeaweb.org/articles.php?doi=10.1257/aer.101.7.3427`
    - "while the offer proportions are significantly lower in the higher stakes treatments compared to the lowest stakes treatment, the actual amount offered increases as stakes increase"
    - "at low stakes [wages for 1.6 h of work] we observe rejections in the range of the extant literature, in the highest stakes condition [wages for 1600 h of work] we observe only a single rejection out of 24 responders"
  - Based on this, I would expect (at the very least), the human solutions to shift with reward in the case of graphs with extreme balanced outcomes

Stable marriages

- Let M be the set of men (with $m \in M$)
- Let W be the set of women (with $w \in W$)
- We assume $|W| = |M|$

Stable marriages

- Let M be the set of men (with $m \in M$)
- Let W be the set of women (with $w \in W$)
- We assume $|W| = |M|$
- We let the bijection $\mu : W \to M$ denote a matching
  - $\mu(w)$ is the man matched to w
  - $\mu^{-1}(m)$ is the woman matched to m
- Abusing notation, we will pretend $\mu : M \cup W \to M \cup W$

Stable marriages

- Let M be the set of men (with $m \in M$)
- Let W be the set of women (with $w \in W$)
- We assume $|W| = |M|$
- We let the bijection $\mu : W \to M$ denote a matching
  - $\mu(w)$ is the man matched to w
  - $\mu^{-1}(m)$ is the woman matched to m
- Abusing notation, we will pretend $\mu : M \cup W \to M \cup W$

Similar to the issue of stability in the network exchange process, the most basic objective is to find a maximum (in this case perfect since we assume $|M| = |W|$) matching between M and W that is *stable*:

Stable marriages

- Let M be the set of men (with $m \in M$)
- Let W be the set of women (with $w \in W$)
- We assume $|W| = |M|$
- We let the bijection $\mu : W \to M$ denote a matching
  - $\mu(w)$ is the man matched to w
  - $\mu^{-1}(m)$ is the woman matched to m
- Abusing notation, we will pretend $\mu : M \cup W \to M \cup W$

Similar to the issue of stability in the network exchange process, the most basic objective is to find a maximum (in this case perfect since we assume $|M| = |W|$) matching between M and W that is *stable*:

A stable matching in the stable matching problem

A matching μ is *unstable* if there exists an unstable (also called *blocking*) pair (m, w) such that m prefers w to his current match $\mu(m)$ and w prefers m to her current match $\mu(w)$. In this case, m and w will leave their current matches to be with each other. A match is *stable* if it contains no unstable (blocking) pairs.

Some examples of stable and unstable matches

We have to check for the presence or absence of a blocking pair; that is, a pair (m, w) such that $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.
Here are a set of preferences for the men and women :

Man	1st	2nd	3rd
x	a	b	c
y	b	a	c
z	a	b	c

Woman	1st	2nd	3rd
a	y	x	z
b	x	y	z
c	x	y	z

Which of the following matchings are stable/unstable?

- Matching 1: $a - x, b - y, c - z$ Stable?
- Matching 2: $a - y, b - x, c - z$ Stable?
- Matching 3: $a - z, b - y, c - x$ Stable?

Some examples of stable and unstable matches

We have to check for the presence or absence of a blocking pair; that is, a pair (m, w) such that $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.
Here are a set of preferences for the men and women :

Man	1st	2nd	3rd
x	a	b	c
y	b	a	c
z	a	b	c

Woman	1st	2nd	3rd
a	y	x	z
b	x	y	z
c	x	y	z

Which of the following matchings are stable/unstable?

- Matching 1: $a - x, b - y, c - z$ Stable? Yes!
- Matching 2: $a - y, b - x, c - z$ Stable?
- Matching 3: $a - z, b - y, c - x$ Stable?

Some examples of stable and unstable matches

We have to check for the presence or absence of a blocking pair; that is, a pair (m, w) such that $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.
Here are a set of preferences for the men and women :

Man	1st	2nd	3rd
x	a	b	c
y	b	a	c
z	a	b	c

Woman	1st	2nd	3rd
a	y	x	z
b	x	y	z
c	x	y	z

Which of the following matchings are stable/unstable?

- Matching 1: $a - x, b - y, c - z$ Stable? Yes!
- Matching 2: $a - y, b - x, c - z$ Stable? Yes!
- Matching 3: $a - z, b - y, c - x$ Stable?

Some examples of stable and unstable matches

We have to check for the presence or absence of a blocking pair; that is, a pair (m, w) such that $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$.
Here are a set of preferences for the men and women :

Man	1st	2nd	3rd
x	a	b	c
y	b	a	c
z	a	b	c

Woman	1st	2nd	3rd
a	y	x	z
b	x	y	z
c	x	y	z

Which of the following matchings are stable/unstable?

- Matching 1: $a - x, b - y, c - z$ Stable? Yes!
- Matching 2: $a - y, b - x, c - z$ Stable? Yes!
- Matching 3: $a - z, b - y, c - x$ Stable? No :(

In Matching 3, we can see that (b, x) is a blocking pair. What other blocking pairs exist?

Stability as an equilibrium

Stability is an equilibrium concept. But like stability in the network exchange setting, and unlike Nash equilibrium, it takes two people to conspire to deviate. In the network exchange setting that was built into the experiments.

This is a form of *coalitional stability*

Stability as an equilibrium

Stability is an equilibrium concept. But like stability in the network exchange setting, and unlike Nash equilibrium, it takes two people to conspire to deviate. In the network exchange setting that was built into the experiments.

This is a form of *coalitional stability*

In some versions of the stable matching problem, we allow individuals to remain "unmarried". This can be incorporated into the problem formulation by letting each man m (respectively, each woman) to put themself into their preference ordering $\succ_m$ (resp. $\succ_w$).

For example, if we have $m_1 \succ_w m_2 \succ_w w \succ_w m_3 \ldots \succ_w m_n$ then w would rather be by herself than with anyone other than m_1 and m_2.

Do stable matchings always exist and, if so, how do we find them?

Aside: When there are n men and women, there are $n!$ possible matchings so we certainly cannot exhaustively check all matchings. And even if we could for a given instance of the problem (ie., a set of preferences for each man and woman) that would not determine if there is always a stable matching.

Fortunately, we have the Gale Shapley algorithm which constructively and efficiently shows how to compute a stable matching for any instance.

Do stable matchings always exist and, if so, how do we find them?

Aside: When there are n men and women, there are $n!$ possible matchings so we certainly cannot exhaustively check all matchings. And even if we could for a given instance of the problem (ie., a set of preferences for each man and woman) that would not determine if there is always a stable matching.

Fortunately, we have the Gale Shapley algorithm which constructively and efficiently shows how to compute a stable matching for any instance.

There are two standard analogous varieties of the Gale Shapley algorithm:

1. Male proposes, woman disposes. Also called Male Proposing Deferred Acceptance (MPDA)
2. Female proposes, man disposes. Also called Female Proposing Deferred Acceptance (FPDA)

FPDA and MPDA are completely analogous, but in general they will produce different matchings.

The FPDA algorithm

- The algorithm will proceed in rounds, at the end of each round, all women will have a set P_w of people to whom they have previously proposed. There will also be a set C of current engagements. Both sets are initially empty

The FPDA algorithm

- The algorithm will proceed in rounds, at the end of each round, all women will have a set P_w of people to whom they have previously proposed. There will also be a set C of current engagements. Both sets are initially empty
- In each round t, every unengaged woman w proposes to the man $m \notin P_w$ that is highest in her preference ranking $\succ_w$. If every woman is engaged at the start of a round, the algorithm terminates

The FPDA algorithm

- The algorithm will proceed in rounds, at the end of each round, all women will have a set P_w of people to whom they have previously proposed. There will also be a set C of current engagements. Both sets are initially empty
- In each round t, every unengaged woman w proposes to the man $m \notin P_w$ that is highest in her preference ranking $\succ_w$. If every woman is engaged at the start of a round, the algorithm terminates
- After a round of female proposals, every man m will consider his set $P_{m,t}$ of current proposals (if any)

The FPDA algorithm

- The algorithm will proceed in rounds, at the end of each round, all women will have a set P_w of people to whom they have previously proposed. There will also be a set C of current engagements. Both sets are initially empty
- In each round t, every unengaged woman w proposes to the man $m \notin P_w$ that is highest in her preference ranking $\succ_w$. If every woman is engaged at the start of a round, the algorithm terminates
- After a round of female proposals, every man m will consider his set $P_{m,t}$ of current proposals (if any)

  We consider what each man m does in this round:

  1. $P_{m,t} = \varnothing$, then m does not do anything in this round

     So now consider the case that $P_{m,t} \neq \varnothing$,and let w^* be the most preferred woman in $P_{m,t}$. That is, $w^* \succ_m w'$ for every $w' \neq w^* \in P_{m,t}$
  2. If m is not currently engaged, he will become engaged to w^* and C is updated accordingly
  3. If m is currently engaged to w (i.e., $(m, w) \in C$), then he will break this engagement if and only if $w^* \succ_m w$ and will then become engaged to w^* In this case, $C := C \setminus \{(m, w)\} \cup \{(m, w^*)\}$

A running example for the FPDA algorithm

$Women$	Men
$\mathbf{a} : x \succ y \succ z \succ w$	$\mathbf{w} : d \succ b \succ a \succ c$
$\mathbf{b} : y \succ x \succ w \succ z$	$\mathbf{x} : b \succ a \succ d \succ c$
$\mathbf{c} : x \succ y \succ z \succ w$	$\mathbf{y} : c \succ b \succ a \succ d$
$\mathbf{d} : y \succ w \succ x \succ z$	$\mathbf{z} : d \succ b \succ c \succ a$

Round 1

Proposals:	New Engagements:
a: x	w: –
b: y	x: a
c: x	y: b
d: y	z: –

Example: Round 2

Women	*Men*
$\mathbf{a} : x^* \succ y \succ z \succ w$	$\mathbf{w} : d \succ b \succ a \succ c$
$\mathbf{b} : y^* \succ x \succ w \succ z$	$\mathbf{x} : b \succ a \succ d \succ c$
$\mathbf{c} : x^* \succ y \succ z \succ w$	$\mathbf{y} : c \succ b \succ a \succ d$
$\mathbf{d} : y^* \succ w \succ x \succ z$	$\mathbf{z} : d \succ b \succ c \succ a$

A * indicates that the man has already been proposed to by this woman.

Round 2

Current:	Proposals:	New Engagements:
w: –	a: –	w: d
x: a	b: –	x: a
y: b	c: y	y: ~~b~~ c
z: –	d: w	z: –

b is "jilted"

Example: Round 3

Women	*Men*
$\mathbf{a} : x{*} \succ y \succ z \succ w$	$\mathbf{w} : d \succ b \succ a \succ c$
$\mathbf{b} : y{*} \succ x \succ w \succ z$	$\mathbf{x} : b \succ a \succ d \succ c$
$\mathbf{c} : x{*} \succ y{*} \succ z \succ w$	$\mathbf{y} : c \succ b \succ a \succ d$
$\mathbf{d} : y^{*} \succ w^{*} \succ x \succ z$	$\mathbf{z} : d \succ b \succ c \succ a$

A * indicates that the man has already been proposed to by this woman.

Round 3

Current:	Proposals:	New Engagements:	
w: d	a: –	w: d	
x: a	b: x	x: ~~a~~ b	**a is "jilted"**
y: ~~b~~ c	c: –	y: ~~b~~ c	
z: –	d: –	z: –	

Example: Round 4

Women	*Men*
$\mathbf{a} : x* \succ y \succ z \succ w$	$\mathbf{w} : d \succ b \succ a \succ c$
$\mathbf{b} : y* \succ x^* \succ w \succ z$	$\mathbf{x} : b \succ a \succ d \succ c$
$\mathbf{c} : x* \succ y^* \succ z \succ w$	$\mathbf{y} : c \succ b \succ a \succ d$
$\mathbf{d} : y^* \succ w^* \succ x \succ z$	$\mathbf{z} : d \succ b \succ c \succ a$

A * indicates that the man has already been proposed to by this woman.

Round 4

Current:	Proposals:	New Engagements:
w: d	a: y	w: d
x: ~~a~~ b	b: –	x: ~~a~~ b
y: ~~b~~ c	c: –	y: ~~b~~ c
z: –	d: –	z: –

a's proposal
not accepted by y
(no change)

Example: Round 5

$Women$	Men
$\mathbf{a}: x* \succ y^* \succ z \succ w$	$\mathbf{w}: d \succ b \succ a \succ c$
$\mathbf{b}: y* \succ x^* \succ w \succ z$	$\mathbf{x}: b \succ a \succ d \succ c$
$\mathbf{c}: x* \succ y^* \succ z \succ w$	$\mathbf{y}: c \succ b \succ a \succ d$
$\mathbf{d}: y^* \succ w^* \succ x \succ z$	$\mathbf{z}: d \succ b \succ c \succ a$

A * indicates that the man has already been proposed to by this woman.

Round 5

Current:	Proposals:	New Engagements:
w: d	a: z	w: d
x: ~~a~~ b	b: –	x: ~~a~~ b
y: ~~b~~ c	c: –	y: ~~b~~ c
z: –	d: –	z: a

Stable:
a:z
b:x
c:y
d:w

Recap

- Stable Marriage Problem
  - Preferences
  - Matching market problem
  - Gale-Shapely algorithm

Proof FPDA terminates

- Assume $|W| = |M| = n$
- Once engaged, no man is ever "free" (i.e. not engaged to a woman)

Proof FPDA terminates

- Assume $|W| = |M| = n$
- Once engaged, no man is ever “free” (i.e. not engaged to a woman)
- If a woman has had $n-1$ unsuccessful proposals, then all $n-1$ previous men are engaged
- Therefore, the nth man must be “free” (as $|W| = n$) and will accept the proposal

Proof FPDA terminates

- Assume $|W| = |M| = n$
- Once engaged, no man is ever "free" (i.e. not engaged to a woman)
- If a woman has had $n - 1$ unsuccessful proposals, then all $n - 1$ previous men are engaged
- Therefore, the nth man must be "free" (as $|W| = n$) and will accept the proposal
- Therefore the algorithm must terminate once all women have made all possible proposals

Proof FPDA terminates

- Assume $|W| = |M| = n$
- Once engaged, no man is ever "free" (i.e. not engaged to a woman)
- If a woman has had $n-1$ unsuccessful proposals, then all $n-1$ previous men are engaged
- Therefore, the nth man must be "free" (as $|W| = n$) and will accept the proposal
- Therefore the algorithm must terminate once all women have made all possible proposals
- As each round results in at least one new proposal, and no woman can propose to the same man twice, it follows that since there are n women and n men there can be at most n^2 rounds

Why is this matching stable?

Proof FPDA produces a stable matching

Let μ be the matching produced by the FPDA. Assume for contradiction that (m, w) is a blocking pair for some man m and woman w.

Proof FPDA produces a stable matching

Let μ be the matching produced by the FPDA. Assume for contradiction that (m, w) is a blocking pair for some man m and woman w.

Therefore w prefers m to $\mu(w)$, and hence must have proposed to m before proposing to $\mu(w)$.

Proof FPDA produces a stable matching

Let μ be the matching produced by the FPDA. Assume for contradiction that (m, w) is a blocking pair for some man m and woman w.

Therefore w prefers m to $\mu(w)$, and hence must have proposed to m before proposing to $\mu(w)$.

By the assumption that (m, w) is a blocking pair, m prefers w to $\mu(m)$.

Proof FPDA produces a stable matching

Let μ be the matching produced by the FPDA. Assume for contradiction that (m, w) is a blocking pair for some man m and woman w.

Therefore w prefers m to $\mu(w)$, and hence must have proposed to m before proposing to $\mu(w)$.

By the assumption that (m, w) is a blocking pair, m prefers w to $\mu(m)$. Therefore:

1. Case 1: if w proposed to m *after* $\mu(m)$ then m would have jilted $\mu(m)$
2. Case 2: if w proposed *before* $\mu(m)$ then m would not have accepted the proposal from $\mu(m)$ as m would already be engaged to either w or someone even more preferred than w

It follows that μ is stable since there cannot be a blocking pair.

Recap

- Stable Marriage Problem
  - Preferences
  - Matching market problem
  - Gale-Shapely algorithm
    - Proof of termination
    - Proof of stability of produced matchings

Mon. Mar 25: Announcements and Corrections

- Your *individual* peer reviews are due this Fri Mar 29 will be submitted via the same assignment
  - Access via the Quercus Assignment tab (through there, you can get to PeerScholar); it's the same assignment where your group submitted it's draft
  - Please do let me know if you aren't in a group, and accordingly can't do the peer review

Properties of the FPDA (MPDA) algorithm

From the analysis of the FPDA stability, we know that FPDA always terminates within n^2 rounds.

And we know that there exists (n, n) instances on which FPDA will use $\Omega(n^2)$ rounds. Can you construct such an instance?

Properties of the FPDA (MPDA) algorithm

From the analysis of the FPDA stability, we know that FPDA always terminates within n^2 rounds.

And we know that there exists (n, n) instances on which FPDA will use $\Omega(n^2)$ rounds. Can you construct such an instance?

Additionally, the order in which women propose in a given round does not change the result. Since the same woman cannot propose to more than one man in a round, it also doesn't matter in what order the men accept or refuse new proposals. That is, the same woman w^* cannot be the reason for canceling more than one engagement. Thus the matching of FPDA is completely determined no matter what order the woman propose or the order that the men make or break engagements.

Different stable matchings

- Note that depending on the instance, the number of stable matchings can vary from exponentially many to a unique stable matching

Different stable matchings

- Note that depending on the instance, the number of stable matchings can vary from exponentially many to a unique stable matching
- In algorithm design (without any self interest by agents), we would be interested in finding an "optimal" solution
  - e.g. a maximum matching or (in the edge or vertex weighted cases) a maximum weight matching
- Do we have a sense of how "good" a given stable matching is?

Different stable matchings

- Note that depending on the instance, the number of stable matchings can vary from exponentially many to a unique stable matching
- In algorithm design (without any self interest by agents), we would be interested in finding an "optimal" solution
  - e.g. a maximum matching or (in the edge or vertex weighted cases) a maximum weight matching
- Do we have a sense of how "good" a given stable matching is?
  - With only preferences, it may not be clear at first why we would prefer one stable matching to another
  - There are many ways that we can define a numeric *social welfare* of a stable matching, but we will study an alternative approach
    - ★ It is always possible (e.g., use the Borda scoring rule) to transform a preference ranking to a utility for the agents based on the match they receive

Female-optimal and male-optimal stable matchings

Ignoring social welfare, we can ask how satisfied will either the men or women be in a stable matching produced by the FPDA and MPDA algorithms.

Female-optimal and male-optimal stable matchings

Ignoring social welfare, we can ask how satisfied will either the men or women be in a stable matching produced by the FPDA and MPDA algorithms.

Define $OPT(w)$ (resp. $Pess(w)$) to be the most (resp. least) preferred man she could be matched with *in a stable matching*. This is a well defined concept since there can only be a finite number of stable matchings.

Female-optimal and male-optimal stable matchings

A matching is *female-optimal* if every woman w is match to $OPT(w)$. Is such a matching possible?

Female-optimal and male-optimal stable matchings

A matching is *female-optimal* if every woman w is match to $OPT(w)$. Is such a matching possible?

Good news for women

Theorem: FPDA results in a female-optimal stable matching for all input instances.

Female-optimal and male-optimal stable matchings

A matching is *female-optimal* if every woman w is match to $OPT(w)$. Is such a matching possible?

Good news for women

Theorem: FPDA results in a female-optimal stable matching for all input instances.

We can also define a male-optimal stable matching in the same way.

Good news for men

Theorem: MPDA results in a male-optimal stable matching for all input instances.

Female-optimal and male-optimal stable matchings

A matching is *female-optimal* if every woman w is match to $OPT(w)$. Is such a matching possible?

Good news for women

Theorem: FPDA results in a female-optimal stable matching for all input instances.

We can also define a male-optimal stable matching in the same way.

Good news for men

Theorem: MPDA results in a male-optimal stable matching for all input instances.

Bad news for society?

FPDA (resp. MPDA) results in a *male-pessimal* (resp. female-pessimal) stable matching for all instances.

Female-optimal and male-optimal stable matchings

A matching is *female-optimal* if every woman w is match to $OPT(w)$. Is such a matching possible?

Good news for women

Theorem: FPDA results in a female-optimal stable matching for all input instances.

We can also define a male-optimal stable matching in the same way.

Good news for men

Theorem: MPDA results in a male-optimal stable matching for all input instances.

Bad news for society?

FPDA (resp. MPDA) results in a *male-pessimal* (resp. female-pessimal) stable matching for all instances.

Remember that Hinge claims they use Gale-Shapley for dating?

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Let w be the first woman rejected (either due to a failed proposal or jilting) by their optimal partner during FPDA's execution.

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Let w be the first woman rejected (either due to a failed proposal or jilting) by their optimal partner during FPDA's execution.

Let w^* be the woman that caused $OPT(w)$ to reject w. Therefore $w^* \succ_{OPT(w)} w$.

As w^* has not yet been rejected by their optimal partner (by assumption), this means that $OPT(w) \succeq_{w^*} OPT(w^*)$.

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Let w be the first woman rejected (either due to a failed proposal or jilting) by their optimal partner during FPDA's execution.

Let w^* be the woman that caused $OPT(w)$ to reject w. Therefore $w^* \succ_{OPT(w)} w$.

As w^* has not yet been rejected by their optimal partner (by assumption), this means that $OPT(w) \succeq_{w^*} OPT(w^*)$.

Now, consider any stable matching μ where w and $OPT(w)$ are matched.

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Let w be the first woman rejected (either due to a failed proposal or jilting) by their optimal partner during FPDA's execution.

Let w^* be the woman that caused $OPT(w)$ to reject w. Therefore $w^* \succ_{OPT(w)} w$.

As w^* has not yet been rejected by their optimal partner (by assumption), this means that $OPT(w) \succeq_{w^*} OPT(w^*)$.

Now, consider any stable matching μ where w and $OPT(w)$ are matched. We know that μ must exist by the definition of OPT.

Sketch of proof that FPDA is female-optimal

Suppose that there is some instance in which the FPDA result is not female-optimal.

Let w be the first woman rejected (either due to a failed proposal or jilting) by their optimal partner during FPDA's execution.

Let w^* be the woman that caused $OPT(w)$ to reject w. Therefore $w^* \succ_{OPT(w)} w$.

As w^* has not yet been rejected by their optimal partner (by assumption), this means that $OPT(w) \succeq_{w^*} OPT(w^*)$.

Now, consider any stable matching μ where w and $OPT(w)$ are matched. We know that μ must exist by the definition of OPT. Also, we know that:

- $w^* \succ_{OPT(w)} w$
- $OPT(w) \succeq_{w^*} OPT(w^*) \succeq_{w^*} \mu(w^*)$, by cases we can show $OPT(w) \succ_{w^*} \mu(w^*)$

Therefore $(OPT(w), w^*)$ are a blocking pair of μ. Contradiction.

Should you be truthful about your preferences?

It does seem reasonable for women to propose in order of their preferences and men to accept their best offer. So why should anyone manipulate and not be truthful about their preferences?

Should you be truthful about your preferences?

It does seem reasonable for women to propose in order of their preferences and men to accept their best offer. So why should anyone manipulate and not be truthful about their preferences?

However, the Gale-Shapley algorithm can be manipulated. That is, there are instances where someone can wind up better off by not stating their true preferences. Here is an example:

First, consider the truthful set of preferences:

$m_1 \succ_{w_1} m_2 \succ_{w_1} m_3$ $\qquad$ $w_2 \succ_{m_1} w_1 \succ_{m_1} w_3$

$m_2 \succ_{w_2} m_1 \succ_{w_2} m_3$ $\qquad$ $w_1 \succ_{m_2} w_2 \succ_{m_2} w_3$

$m_1 \succ_{w_3} m_2 \succ_{w_3} m_3$ $\qquad$ $w_1 \succ_{m_3} w_2 \succ_{m_3} w_3$

FPDA will compute the following stable matching:
$(w_1, m_1), (w_2, m_2), (w_3, m_3)$
You should check this by running FPDA.

But what if m_1 is not always truthful?

Suppose that m_1 lies in round 1 and rejects the proposal from w_1 (instead being engaged to w_3) even though $w_1 \succ_{m_1} w_3$.
This will result in the following matching: $(w_1, m_2), (w_2, m_1), (w_3, m_3)$ where now m_1 is matched to w_2, an improvement for him.
You should check this by running FPDA with m_1 deviating as indicated.

But what if m_1 is not always truthful?

Suppose that m_1 lies in round 1 and rejects the proposal from w_1 (instead being engaged to w_3) even though $w_1 \succ_{m_1} w_3$.
This will result in the following matching: $(w_1, m_2), (w_2, m_1), (w_3, m_3)$ where now m_1 is matched to w_2, an improvement for him.
You should check this by running FPDA with m_1 deviating as indicated.

NOTE: It is not easy to prove but in FPDA, women can never benefit by being untruthful. That is, women should always propose in the order of their preferences when using the FPDA.

But what if m_1 is not always truthful?

Suppose that m_1 lies in round 1 and rejects the proposal from w_1 (instead being engaged to w_3) even though $w_1 \succ_{m_1} w_3$.
This will result in the following matching: $(w_1, m_2), (w_2, m_1), (w_3, m_3)$ where now m_1 is matched to w_2, an improvement for him.
You should check this by running FPDA with m_1 deviating as indicated.

NOTE: It is not easy to prove but in FPDA, women can never benefit by being untruthful. That is, women should always propose in the order of their preferences when using the FPDA.

Of course, it is just the opposite when using MPDA: Men cannot benefit from lying but women can sometimes gain by an untruthful rejection.

Lots of extensions of deferred acceptance (DA) and other considerations

Many applications are *many-to-one* and not 1-1 as in the basic formulation. For example, a University accepts many students. This extension is not difficult to handle.

One way would be to replicate a University K times if it had a quota of K students. Is this a good solution?

Lots of extensions of deferred acceptance (DA) and other considerations

Many applications are *many-to-one* and not 1-1 as in the basic formulation. For example, a University accepts many students. This extension is not difficult to handle.

One way would be to replicate a University K times if it had a quota of K students. Is this a good solution?

This is inefficient (especially if K is big and it imposes an artificial ranking amongst the copies.

Instead, we can extend Gale-Shapley by having each University have a quota and while that quota is not filled, they keep admitting students. When the quota is filled and the get another request, they can reject it or take it and remove the least desirable student. (Of course, they don't announce any decisions until the end of the admission process and hopefully have a reliable way to rank students.) Now Universities (the men in FPDA) can also manipulate by misreporting their quota.

Other important considerations in stable matching

- Partial preferences. In general, our preference relation is usually partial. More specifically, our preferences are often a weak ordering (that is, we may be indifferent between various choices). Now there can be different ways to define a blocking pair and stability:
  1. Weak stability: (m, w) is a blocking pair iff both m and w are strictly better.
  2. Strong stability: (m, w) is a blocking pair iff at least one of m and w is strictly better, and the other is at least indifferent
  3. Super strong stability: (m, w) is a blocking pair if neither m nor w is worse off.

Other important considerations in stable matching

- Partial preferences. In general, our preference relation is usually partial. More specifically, our preferences are often a weak ordering (that is, we may be indifferent between various choices). Now there can be different ways to define a blocking pair and stability:
  1. Weak stability: (m, w) is a blocking pair iff both m and w are strictly better.
  2. Strong stability: (m, w) is a blocking pair iff at least one of m and w is strictly better, and the other is at least indifferent
  3. Super strong stability: (m, w) is a blocking pair if neither m nor w is worse off.

  Gale-Shapley is easily extended to handle weak stability (i.e., break ties arbitrarily), but strong and super strong stability require modifications.

Partial preferences and couples

- Partial preferences raises the issue as to how to possibly resolve some preferences, for instance by interviews.
  - However, this can be costly w.r.t. time (e.g., for both employers and candidates)
  - Given some (say probabilistic) belief about preferences, who should you choose for your interviews or where to apply?
  - Do you only go for the positions that you can most likely get, or should you try for some of your most desired choices?
  - These are called "reach and safety strategies" in contrast to just interviewing "within your tier".
- Did you have a a strategy in applying to University or if you are applying to graduate school, do you have a strategy where to apply?

Partial preferences and couples

- As mentioned before, variants of Gale-Shapely have been used to great success in matching medical school students to residence positions

Partial preferences and couples

- As mentioned before, variants of Gale-Shapely have been used to great success in matching medical school students to residence positions
- However, the number of couples graduating medical school has been increasing. (In 2015, 6% of resident applications were coupled.)
  - Couples rank residency positions, *but want to remain together*
  - This additional wrinkle makes it an NP-complete problem to determine if there is a stable matching
  - Various ways of approaching problem in practice (e.g. using SAT solvers as advocated by Drummond, Perrault and Bacchus).

Concluding stable matching

Very important and still an active topic as stable matching is used in a number of applications. In the kidney exchange application (stable matching in a non-bipartite graph whose nodes are donor-recipient pairs), it can literally be a matter of life and death. Here edges represent a compatible match. Here we can also have weights on the edges (to represent how good a match is) and weights on the nodes (to perhaps represent how urgent is the match).

As another indication of the importance of stable matching, the 2012 Nobel Prize in Economics was awarded to Lloyd Shapley and Alvin Roth for their work in the theory (Shapley) and application (Roth) of stable matching algorithms.

Recap

With practice & review, you'll be able to:

- Define the Stable Marriage Problem
  - Define total and partial preferences
  - Define a two-sided matching market problem
  - Define a stable matching, and determine whether a given matching is stable
  - Recall and execute the Gale-Shapely algorithm
    - Prove it terminates
    - Prove it returns stable matchings
    - Define female-optimal and male-optimal matchings, and prove the relationship between the Gale-Shapely algorithm and these matchings
    - Recall how the Gale-Shapely algorithm is susceptible to manipulation
  - Define weak, strong, and superstrong stability, and explain how they extend the stable matching problem