Stable Marriages

Let's play matchmaker.

Stable Marriages

Let's play matchmaker.

Given n people and n pets, where

- Each person has a complete list of preferences for the pets,
- Each pet has a complete list of preferences for the people.

Stable Marriages

Let's play matchmaker.

Given n people and n pets, where

- Each person has a complete list of preferences for the pets,
- Each pet has a complete list of preferences for the people.

Example. Basil, Evan, and Felicia are looking for pets, while Alina the aardvark, Casper the cat, and Dakota the dog are looking for owners,

with the following preferences:

Stable Marriages

Let's play matchmaker.

Given n people and n pets, where

- Each person has a complete list of preferences for the pets,
- Each pet has a complete list of preferences for the people.

Example. Basil, Evan, and Felicia are looking for pets, while Alina the aardvark, Casper the cat, and Dakota the dog are looking for owners,

with the following preferences:

People's Preferences

	Basil	Evan	Felicia
1^{st}	Alina	Alina	Dakota
2^{nd}	Casper	Dakota	Casper
3^{rd}	Dakota	Casper	Alina

Pets' Preferences

	Alina	Casper	Dakota
1^{st}	Felicia	Basil	Evan
2^{nd}	Basil	Felicia	Felicia
3^{rd}	Evan	Evan	Basil

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Definition: An **instability** is when one person and one pet both prefer each other to their partner.

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Definition: An **instability** is when one person and one pet both prefer each other to their partner.

Example. Suppose:

- Basil is the owner of the Alina the aardvark
- Evan is the owner of Casper the cat

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Definition: An **instability** is when one person and one pet both prefer each other to their partner.

Example. Suppose:

- Basil is the owner of the Alina the aardvark
- Evan is the owner of Casper the cat
- Casper the cat prefers Basil to Evan.

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Definition: An **instability** is when one person and one pet both prefer each other to their partner.

Example. Suppose:

- Basil is the owner of the Alina the aardvark
- Evan is the owner of Casper the cat
- Casper the cat prefers Basil to Evan.

If Basil prefers Casper to Alina: ______________________________

Stable Marriages

Goal: Create a perfect matching of **stable marriages**.

That is, find a set of n pairings where there are no instabilities:

Definition: An **instability** is when one person and one pet both prefer each other to their partner.

Example. Suppose:

- Basil is the owner of the Alina the aardvark
- Evan is the owner of Casper the cat
- Casper the cat prefers Basil to Evan.

If Basil prefers Casper to Alina: ____________________

If Basil prefers Alina to Casper: ____________________

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

1. Start with no pairings.

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

1. Start with no pairings.
2. As long as at least one person is not paired, repeat the following: Each unpartnered person proposes to his/her **next** most preferred pet (based on preference list).

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

1. Start with no pairings.
2. As long as at least one person is not paired, repeat the following: Each unpartnered person proposes to his/her **next** most preferred pet (based on preference list).
   Each pet then decides whether to accept or reject the proposal(s), as follows:
   - If the pet has one proposal, it accepts the pairing (tentatively).
   - If the pet has ≥ 1 proposal (old or new), it uses its preference list to decides which proposal to accept, rejecting all others.

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

1. Start with no pairings.
2. As long as at least one person is not paired, repeat the following:
   Each unpartnered person proposes to his/her **next** most preferred pet (based on preference list).
   Each pet then decides whether to accept or reject the proposal(s), as follows:
   - If the pet has one proposal, it accepts the pairing (tentatively).
   - If the pet has ≥ 1 proposal (old or new), it uses its preference list to decides which proposal to accept, rejecting all others.
3. When all people are partnered, stop. These are n stable marriages.

The Gale–Shapley Algorithm

Theorem. (Gale, Shapley, 1962) In the above situation, there always exists a perfect matching of stable marriages.

Proof. Use the **Gale–Shapley Algorithm** to create the pairings.

1. Start with no pairings.
2. As long as at least one person is not paired, repeat the following: Each unpartnered person proposes to his/her **next** most preferred pet (based on preference list).
   Each pet then decides whether to accept or reject the proposal(s), as follows:
   - If the pet has one proposal, it accepts the pairing (tentatively).
   - If the pet has ≥ 1 proposal (old or new), it uses its preference list to decides which proposal to accept, rejecting all others.
3. When all people are partnered, stop. These are n stable marriages.

$<<$Time for your moment of zen$>>$

Applying the Gale–Shapley Algorithm

Here is a complete set of preferences for 4 people and 4 pets.

People:

Emma
Jae
Tracy
Robot Human

People's Preferences

	Emma	Jae	Tracy	Robot
1st	Parrot	Parrot	Parrot	Sally
2nd	Sally	Casper	Dakota	Dakota
3rd	Casper	Sally	Casper	Parrot
4th	Dakota	Dakota	Sally	Casper

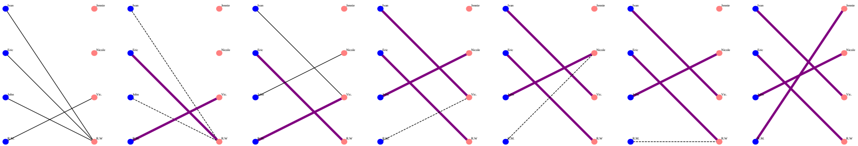
Pets:

Casper the Cat
Dakota the Dog
Sally the Snake
Robot Parrot

Women's Preferences

	Casper	Dakota	Sally	Parrot
1st	Jae	Tracy	Tracy	Jae
2nd	Tracy	Robot	Emma	Robot
3rd	Robot	Jae	Robot	Emma
4th	Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



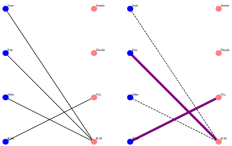
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	Parrot	Parrot	Sally
Sally	Casper	Dakota	Dakota
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	Casper

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	Tracy	Tracy	Jae
Tracy	Robot	Emma	Robot
Robot	Jae	Robot	Emma
Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



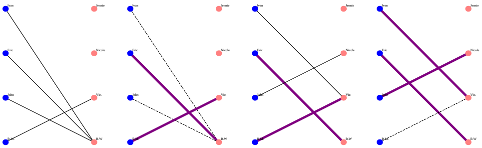
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	**Parrot**	**Sally**
Sally	Casper	Dakota	Dakota
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	Casper

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	Tracy	Tracy	**Jae**
Tracy	Robot	Emma	Robot
Robot	Jae	**Robot**	**Emma**
Emma	Emma	Jae	**Tracy**

The Algorithm, Pictorially



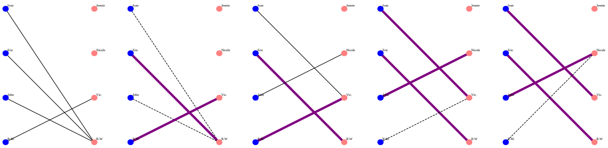
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	Parrot	**Sally**
Sally	Casper	**Dakota**	Dakota
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	Casper

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	**Tracy**	Tracy	**Jae**
Tracy	Robot	**Emma**	Robot
Robot	Jae	**Robot**	Emma
Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



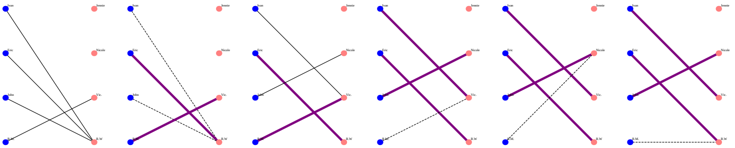
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	Parrot	Sally
Sally	Casper	**Dakota**	**Dakota**
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	Casper

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	**Tracy**	Tracy	**Jae**
Tracy	**Robot**	**Emma**	Robot
Robot	Jae	Robot	Emma
Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



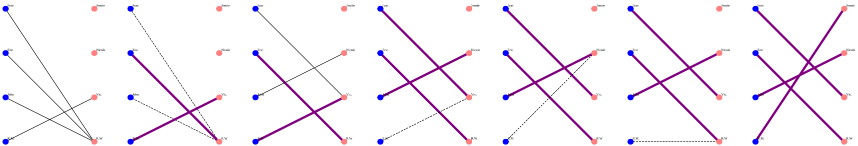
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	Parrot	Sally
Sally	Casper	**Dakota**	Dakota
Casper	Sally	Casper	**Parrot**
Dakota	Dakota	Sally	Casper

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	**Tracy**	Tracy	**Jae**
Tracy	Robot	**Emma**	**Robot**
Robot	Jae	Robot	Emma
Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



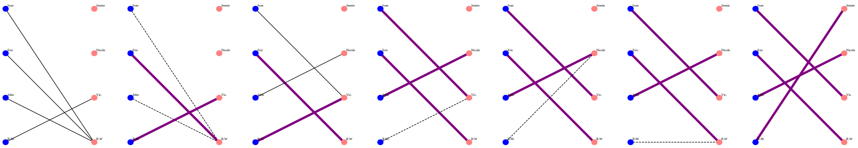
People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	Parrot	Sally
Sally	Casper	**Dakota**	Dakota
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	**Casper**

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	**Tracy**	Tracy	**Jae**
Tracy	Robot	**Emma**	Robot
Robot	Jae	Robot	Emma
Emma	Emma	Jae	Tracy

The Algorithm, Pictorially



People's Preferences

Emma	Jae	Tracy	Robot
Parrot	**Parrot**	Parrot	Sally
Sally	Casper	**Dakota**	Dakota
Casper	Sally	Casper	Parrot
Dakota	Dakota	Sally	**Casper**

Pets' Preferences

Casper	Dakota	Sally	Parrot
Jae	**Tracy**	Tracy	**Jae**
Tracy	Robot	**Emma**	Robot
Robot	Jae	Robot	Emma
Emma	Emma	Jae	Tracy

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof. We must show that the algorithm always stops, and that when it stops, the output is indeed a full set of stable marriages.

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof. We must show that the algorithm always stops, and that when it stops, the output is indeed a full set of stable marriages.

The algorithm terminates.

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof. We must show that the algorithm always stops, and that when it stops, the output is indeed a full set of stable marriages.

The algorithm terminates.

- In each step, at least one proposal occurs.
- There are only a finite number of possible proposals.
- No proposal occurs more than once.

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof. We must show that the algorithm always stops, and that when it stops, the output is indeed a full set of stable marriages.

The algorithm terminates.

- In each step, at least one proposal occurs.
- There are only a finite number of possible proposals.
- No proposal occurs more than once.

Claim: Upon termination, everyone is partnered.

Proof of Correctness

Claim. The Gale–Shapley Algorithm gives a set of n stable marriages.

Proof. We must show that the algorithm always stops, and that when it stops, the output is indeed a full set of stable marriages.

The algorithm terminates.

- In each step, at least one proposal occurs.
- There are only a finite number of possible proposals.
- No proposal occurs more than once.

Claim: Upon termination, everyone is partnered.

- Once a pet finds a partner, it stays partnered.
- If a pet is not partnered at the end, it had no proposal.
- It follows that there is also some person not engaged. However, he/she must have proposed to the lonely pet during some round!

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.
- During the algorithm, Bob would have proposed to Casper **before** his current pet.

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.
- During the algorithm, Bob would have proposed to Casper **before** his current pet.
- Casper must have turned down Bob.

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.
- During the algorithm, Bob would have proposed to Casper **before** his current pet.
- Casper must have turned down Bob.
  - (Which means Casper was proposed to by someone he prefers!)

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.
- During the algorithm, Bob would have proposed to Casper **before** his current pet.
- Casper must have turned down Bob.
  - (Which means Casper was proposed to by someone he prefers!)
- Hence, whatever person is Casper's owner in the end, Casper certainly prefers his owner to Bob.

Proof of Correctness

The output is a set of stable marriages.

We ask: Is there an instability?

- Suppose Bob (human) prefers Casper to his current pet.
- During the algorithm, Bob would have proposed to Casper **before** his current pet.
- Casper must have turned down Bob.
  - (Which means Casper was proposed to by someone he prefers!)
- Hence, whatever person is Casper's owner in the end, Casper certainly prefers his owner to Bob.
- Therefore, there is no instability.

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**.

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a “sub-optimal” pet.

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a "sub-optimal" pet.

- Let H be the first human who is rejected by his optimal pet P during the algorithm.

H-Optimal Pairings ($\mathcal{S}'$)

Relative Pref's

I	P

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a "sub-optimal" pet.

- Let H be the first human who is rejected by his optimal pet P during the algorithm.

[*That is, there is some other set* $\mathcal{S}'$ *of stable marriages in which* H *is paired with* P.]

H-Optimal Pairings ($\mathcal{S}'$)

Relative Pref's

I	P

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a "sub-optimal" pet.

- Let H be the first human who is rejected by his optimal pet P during the algorithm.
[*That is, there is some other set* $\mathcal{S}'$ *of stable marriages in which* H *is paired with* P.]
- H is rejected because some human I proposes to P whom P prefers to H.

H-Optimal Pairings ($\mathcal{S}'$)

Relative Pref's

I	P

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a "sub-optimal" pet.

- Let H be the first human who is rejected by his optimal pet P during the algorithm. [*That is, there is some other set $\mathcal{S}'$ of stable marriages in which H is paired with P.*]
- H is rejected because some human I proposes to P whom P prefers to H.
- Since H is the *first* human rejected, we know I likes P at least as much as his optimal pet.

H-Optimal Pairings ($\mathcal{S}'$)

Relative Pref's

I	P

Human-optimality

Claim. The marriages $\mathcal{S}$ generated by the Gale–Shapley Algorithm are **human optimal**. That is, given any other set of stable marriages, each person will only be paired with a pet lower on his preference list.

Proof. Suppose that during the Gale–Shapley Algorithm, there is a human who is paired with a "sub-optimal" pet.

- Let H be the first human who is rejected by his optimal pet P during the algorithm.
[*That is, there is some other set $\mathcal{S}'$ of stable marriages in which H is paired with P.*]
- H is rejected because some human I proposes to P whom P prefers to H.
- Since H is the *first* human rejected, we know I likes P at least as much as his optimal pet.
- This, in turn, creates an instability in $\mathcal{S}'$ since P prefers I to H and I prefers P to the pet he is paired with.

***H*-Optimal Pairings ($\mathcal{S}'$)**

Relative Pref's

I	P

Last remarks

- The marriages generated by Gale–Shapley are human optimal.
- The marriages generated by Gale–Shapley are pet pessimal.

Last remarks

- The marriages generated by Gale–Shapley are human optimal.
- The marriages generated by Gale–Shapley are pet pessimal.
- Run the algorithm with the pets proposing to reverse the roles.
  If you do this and get the same marriages, ________________

Last remarks

- The marriages generated by Gale–Shapley are human optimal.
- The marriages generated by Gale–Shapley are pet pessimal.
- Run the algorithm with the pets proposing to reverse the roles. If you do this and get the same marriages, ________________
- If not all rankings are made, then there may be unpaired entities. For example, what if Robot Human did not like Casper?

Last remarks

- The marriages generated by Gale–Shapley are human optimal.
- The marriages generated by Gale–Shapley are pet pessimal.
- Run the algorithm with the pets proposing to reverse the roles.
  If you do this and get the same marriages, ________________
- If not all rankings are made, then there may be unpaired entities.
  For example, what if Robot Human did not like Casper?
- This algorithm was originally devised by Gale and Shapley
  with n men proposing to n women.

Last remarks

- The marriages generated by Gale–Shapley are human optimal.
- The marriages generated by Gale–Shapley are pet pessimal.
- Run the algorithm with the pets proposing to reverse the roles.
  If you do this and get the same marriages, ______________
- If not all rankings are made, then there may be unpaired entities. For example, what if Robot Human did not like Casper?
- This algorithm was originally devised by Gale and Shapley with n men proposing to n women.
- The National Resident Matching Program implements this algorithm to match medical students to residency programs. (`http://www.nrmp.org`)