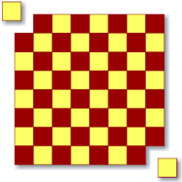
Bipartite Matchings and Stable Marriage

Meghana Nasre
CSE, IIT Madras

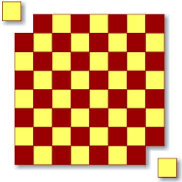
Faculty Development Program
SSN College of Engineering, Chennai
Jan. 27, 2016

Lets start with some fun..



- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.

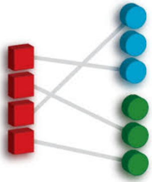
Lets start with some fun..



- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.

$\boxed{\text{Can you tile the chess board?}}$

Some routine admin tasks..



- How does your dept. do project allotments?
- How does it do course allotments to faculty?
- What do we want to achieve while doing these allotments?

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

- Who are the participants?
  1. Students who have passed 12-th class.
  2. Programs at various institutions say cse-iitb, mech-nitk, ...

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

- Who are the participants?
  1. Students who have passed 12-th class.
  2. Programs at various institutions say cse-iitb, mech-nitk, ...
- A student ranks programs according to his/ her preference.
  Rohan: cse-iitb, cse-nitk, cse-iitp

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

- Who are the participants?
  1. Students who have passed 12-th class.
  2. Programs at various institutions say cse-iitb, mech-nitk, ...
- A student ranks programs according to his/ her preference.
  Rohan: cse-iitb, cse-nitk, cse-iitp
- A program also ranks students depending on:

	JEE Main	JEE Advanced
Rohan	32	447
Avanti	10	663

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

- Who are the participants?
  1. Students who have passed 12-th class.
  2. Programs at various institutions say cse-iitb, mech-nitk, ...
- A student ranks programs according to his/ her preference.
  Rohan: cse-iitb, cse-nitk, cse-iitp
- A program also ranks students depending on:

	JEE Main	JEE Advanced
Rohan	32	447
Avanti	10	663

- Scale of the problem: 34,000 odd seats across the country competed by 1.5 lakh students.

Same problem at a larger scale..

Have you heard of Joint Seat Allocation?

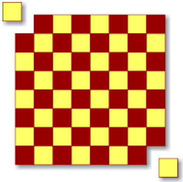
Common Counselling for IITs, NITs, IIITs, GFTIs – India 2015

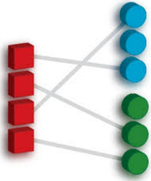
- Who are the participants?
  1. Students who have passed 12-th class.
  2. Programs at various institutions say cse-iitb, mech-nitk, ...
- A student ranks programs according to his/ her preference.
  Rohan: cse-iitb, cse-nitk, cse-iitp
- A program also ranks students depending on:

	JEE Main	JEE Advanced
Rohan	32	447
Avanti	10	663

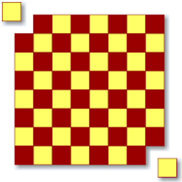
- Scale of the problem: 34,000 odd seats across the country competed by 1.5 lakh students.
- No longer possible to do manual allotments!

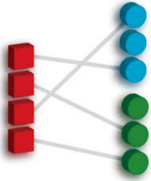
What ties them together?





What ties them together?





A beautiful theory of matchings and stable marriage

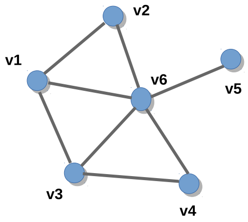
Outline of talk

- Quick introduction to graphs.
- Matchings in graphs.
- Bipartite matching algorithm.
- Stable matchings.

Introduction to graphs

Why graphs?

Represent binary relations between objects

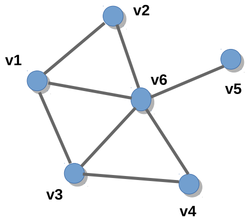


$$G = (V, E) \; ; \; n = |V| = 6, \; m = |E| = 8.$$

How large can m be?

Why graphs?

Represent binary relations between objects



$$G = (V, E) \; ; \; n = |V| = 6, \; m = |E| = 8.$$

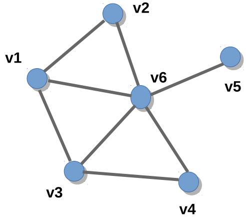
How large can m be?

Representation:

- Adj. lists – $O(m + n)$ space.
- Adj. matrix – $O(n^2)$ space.

Why graphs?

Represent binary relations between objects



$G = (V, E)$; $n = |V| = 6$, $m = |E| = 8$.
How large can m be?

Representation:

- Adj. lists – $O(m+n)$ space.
- Adj. matrix – $O(n^2)$ space.

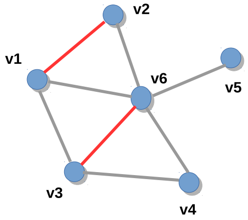
Search methods:

- Breadth First Search.
- Depth First Search.

Matching in a graph

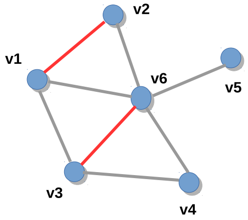
Matching in a graph

A matching M is a set of vertex disjoint edges.



Matching in a graph

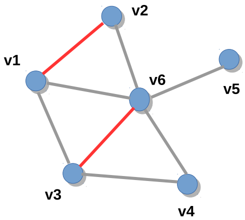
A matching M is a set of vertex disjoint edges.



- $\boxed{\text{Goal:}}$ compute a largest sized matching.
- $\boxed{\text{Question:}}$ is the above matching as large as possible?

Maximal vs. maximum matchings

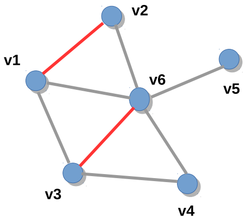
maximal



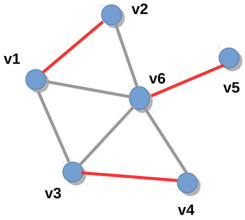
- No more edges can be added.
- Size need not be the largest possible.

Maximal vs. maximum matchings

maximal



maximum

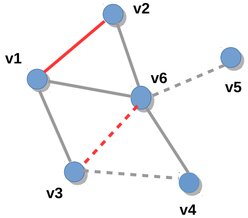


- No more edges can be added.
- Size need not be the largest possible.

- Largest possible size.
- Maximum $\geq$ maximal.

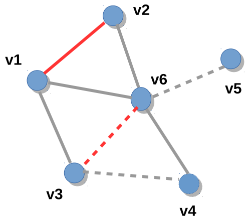
Alternating paths

A path having alternate matched and unmatched edges.



Alternating paths

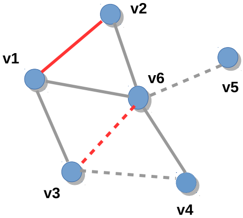
A path having alternate matched and unmatched edges.



- Is there any other alternating path?

Alternating paths

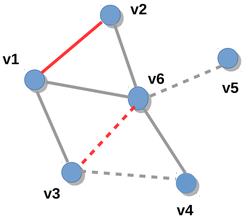
A path having alternate matched and unmatched edges.



- Is there any other alternating path?
- Which paths are not alternating?

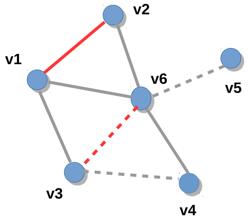
Augmenting paths

An alternating path starting and ending in free vertices.



Augmenting paths

An alternating path starting and ending in free vertices.



- How are augmenting paths useful?
- Properties of augmenting paths.

Using augmenting paths

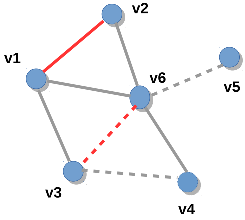
Berge's Theorem

- If aug. path p is present $\Rightarrow$ size of matching can be increased.

Using augmenting paths

Berge's Theorem

- If aug. path p is present $\Rightarrow$ size of matching can be increased.
- $M' = M \oplus P$.

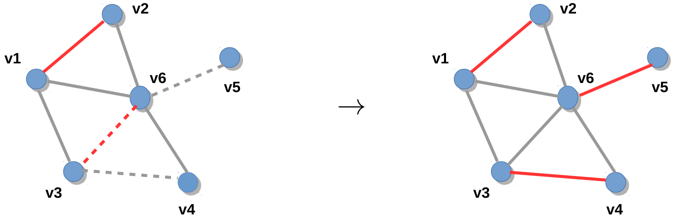


$\rightarrow$

Using augmenting paths

Berge's Theorem

- If aug. path p is present $\Rightarrow$ size of matching can be increased.
- $M' = M \oplus P$.



Using augmenting paths

Berge's Theorem

- If no aug. path w.r.t. $M \Rightarrow M$ is maximum.

Using augmenting paths

Berge's Theorem

- If no aug. path w.r.t. $M \Rightarrow M$ is maximum.

Proof (by contradiction)

- Suppose M does not admit any aug. path and still it is not maximum.
- Some other matching M' is maximum.

Using augmenting paths

Berge's Theorem

- If no aug. path w.r.t. $M \Rightarrow M$ is maximum.

Proof (by contradiction)

- Suppose M does not admit any aug. path and still it is not maximum.
- Some other matching M' is maximum.
- Consider $H = (V, M \oplus M')$.
  - Every vertex has degree at most 2.
  - H is a collection of paths and even cycles.
- Construct an aug. path w.r.t. M.

Maximum matching algorithm

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Maximum matching algorithm

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Two questions that need to be always asked:

1. correctness

Maximum matching algorithm

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Two questions that need to be always asked:

1. correctness ✓.

Maximum matching algorithm

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Two questions that need to be always asked:

1. correctness ✓.
2. complexity/ running time?

Maximum matching algorithm

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

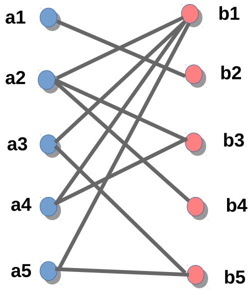
Two questions that need to be always asked:

1. correctness ✓.
2. complexity/ running time?

How to efficiently compute an augmenting path?

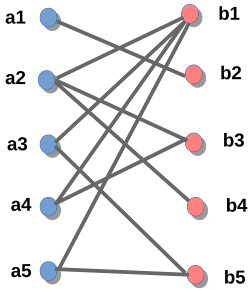
Bipartite matching algorithm

Bipartite graphs



When is G bipartite?

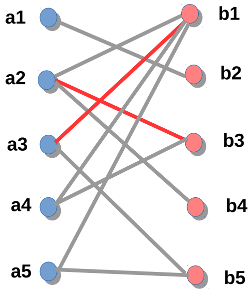
Bipartite graphs



When is G bipartite?

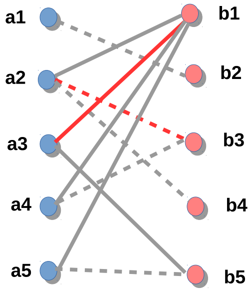
- Vertices can be partitioned into 2 disjoint sets.
- G does not have any odd cycle.
- G is 2-colorable.

Finding aug. path in bipartite graphs



- Is this matching maximal?
- Are there augmenting paths with respect to M?

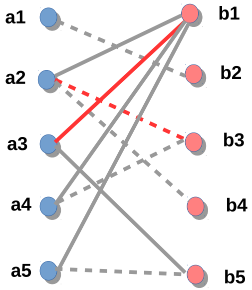
Finding aug. path in bipartite graphs



Aug. paths w.r.t. M:

- $\langle a_1, b_2 \rangle$.
- $\langle a_5, b_5 \rangle$.
- $\langle a_4, b_3, a_2, b_4 \rangle$.

Finding aug. path in bipartite graphs



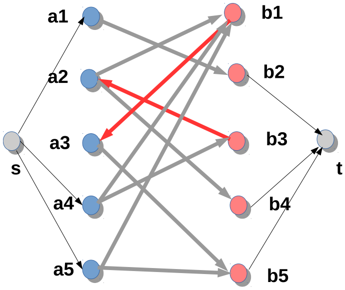
Aug. paths w.r.t. M:

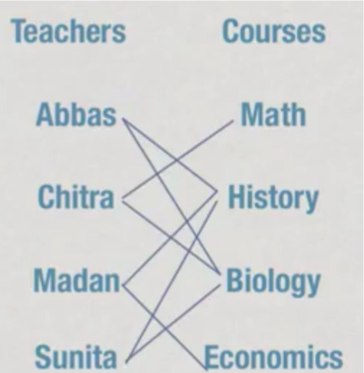
- $\langle a_1, b_2 \rangle$.
- $\langle a_5, b_5 \rangle$.
- $\langle a_4, b_3, a_2, b_4 \rangle$.

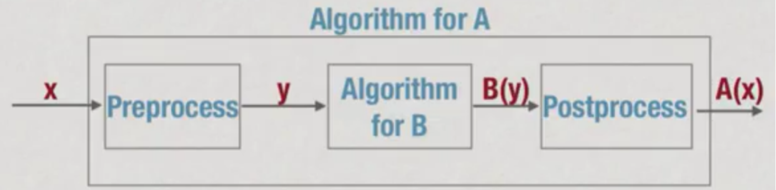
Finding aug. paths – reachability in a modified graph.

This makes the bipartite case easy!

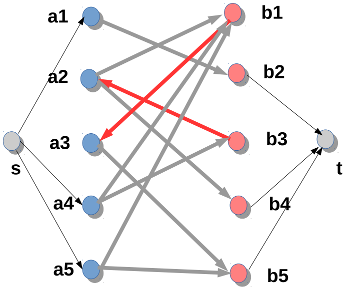
Finding aug. path in bipartite graphs





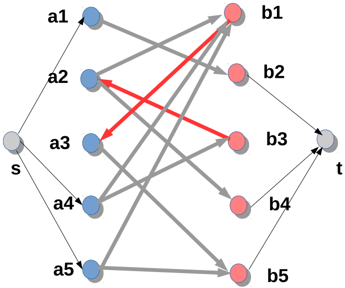


Finding aug. path in bipartite graphs



- Add dummy nodes s and t.
- Add edges from s to unmatched vertices in A.
- Add edges from unmatched vertices in B to t.
- Matched edges: $B \to A$.
- Unmatched edges: $A \to B$.

Finding aug. path in bipartite graphs



- Add dummy nodes s and t.
- Add edges from s to unmatched vertices in A.
- Add edges from unmatched vertices in B to t.
- Matched edges: $B \to A$.
- Unmatched edges: $A \to B$.

A directed path p from s to $t \leftrightarrow$ there exists an aug. path w.r.t. M.

Matching algorithm is Reduction

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

1. How many iterations?

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

1. How many iterations? $O(n)$

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

1. How many iterations? $O(n)$
2. How long does each iteration take?

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

1. How many iterations? $O(n)$
2. How long does each iteration take? $O(m + n)$.

Maximum matching in bipartite graph

Iterative improvement algorithm

Input: Graph $G = (V, E)$.
Output: A maximum sized matching M in G.

---

1. initialize M to be empty.
2. while there exists an aug. path p w.r.t. M
   - $M = M \oplus p$.
3. return M.

Running time?

1. How many iterations? $O(n)$
2. How long does each iteration take? $O(m + n)$.
3. Total running time: $O(mn)$.

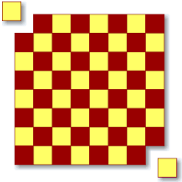
Efficient matching algorithms

- The best known for both bipartite and general graphs is $O(m\sqrt{n})$ time algorithms.
- Algorithms on general graphs are significantly involved.

Efficient matching algorithms

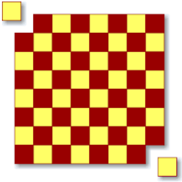
- The best known for both bipartite and general graphs is $O(m\sqrt{n})$ time algorithms.
- Algorithms on general graphs are significantly involved. Edmond's Blossom Shrinking algorithm. (Also the paper where the notion of polynomial time being efficient was formalized).
- Weighted matchings can also be computed in polynomial time.

Back to square ... 62



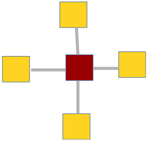
- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.
- Can you tile the chess board with dominoes?

Back to square ... 62

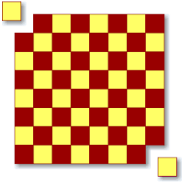


- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.
- Can you tile the chess board with dominoes?

- Cells of the board – vertices.
- 2 cells are adjacent if a domino can cover it.

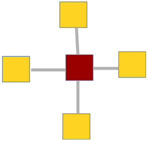


Back to square ... 62

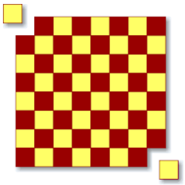


- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.
- Can you tile the chess board with dominoes?

- Cells of the board – vertices.
- 2 cells are adjacent if a domino can cover it.

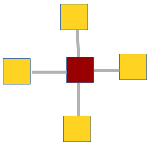


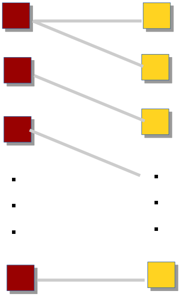
Back to square ... 62



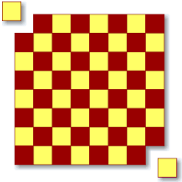
- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.
- Can you tile the chess board with dominoes?

- Cells of the board – vertices.
- 2 cells are adjacent if a domino can cover it.



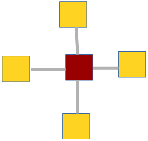


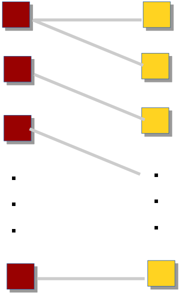
Back to square ... 62



- A chess board with 2 corners removed.
- An infinite supply of 2×1 tiles.
- Can you tile the chess board with dominoes?

- Cells of the board – vertices.
- 2 cells are adjacent if a domino can cover it.





It is impossible to match all vertices, therefore no tiling exists!

Stable matching problem

A bit of context..

- Introduced and studied by Gale and Shapley (1962).

A bit of context..

- Introduced and studied by Gale and Shapley (1962).
- Context of college admissions.

A bit of context..

- Introduced and studied by Gale and Shapley (1962).
- Context of college admissions.
- Wide range of applications in real world:
  - National Residency Matching Program (NRMP), USA.

A bit of context..

- Introduced and studied by Gale and Shapley (1962).
- Context of college admissions.
- Wide range of applications in real world:
  - National Residency Matching Program (NRMP), USA.
  - Scottish Foundation Allocation (SFA), UK.

A bit of context..

- Introduced and studied by Gale and Shapley (1962).
- Context of college admissions.
- Wide range of applications in real world:
  - National Residency Matching Program (NRMP), USA.
  - Scottish Foundation Allocation (SFA), UK.
  - Joint Seat Allocation (JoSA), India 2015.
  - ...
- Classical setting: marriage between n men and n women.

Setting

Input:

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \quad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

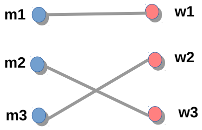
Goal: To compute a matching that is "optimal".

Setting

Input:

m_1 :	w_1, w_2, w_3	w_1 :	m_3, m_2, m_1
m_2 :	w_2, w_3, w_1	w_2 :	m_2, m_1, m_3
m_3 :	w_3, w_1, w_2	w_3 :	m_1, m_3, m_2

Goal: To compute a matching that is "optimal".



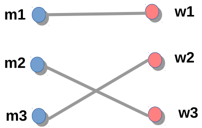
Question: Is this a good matching?

Setting

Input:

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \quad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "optimal".



Question: Is this a good matching? pair (m_2, w_2) is bad!

Setting

Input:

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \qquad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

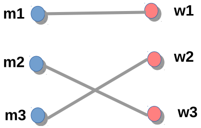
Goal: To compute a matching that is "stable".

Setting

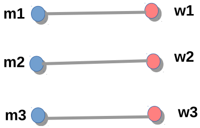
Input:

$$
\begin{array}{llcll}
m_1: & w_1, w_2, w_3 & \qquad & w_1: & m_3, m_2, m_1 \\
m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\
m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2
\end{array}
$$

Goal: To compute a matching that is "stable".



unstable.



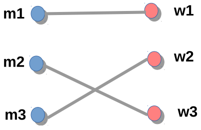
stable.

Setting

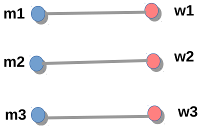
Input:

$$\begin{array}{ll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{ll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".



unstable.



stable.

unstable pair: A pair (m, w) not matched to each other, both of which prefer each other to their current partners in M.

Stable marriage problem

Input:

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \qquad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance?

Stable marriage problem

Input:

$$\begin{array}{lll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{lll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance?
- Is stable marriage unique?

Stable marriage problem

Input:

$$\begin{array}{ll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{ll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance?
- Is stable marriage unique?
- Can it be computed efficiently?

Stable marriage problem

Remarkable result of Gale and Shapley

$$
\begin{array}{llcll}
m_1: & w_1, w_2, w_3 & \qquad & w_1: & m_3, m_2, m_1 \\
m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\
m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2
\end{array}
$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance?

Stable marriage problem

Remarkable result of Gale and Shapley

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \quad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance? Yes, always!
- Is stable marriage unique?

Stable marriage problem

Remarkable result of Gale and Shapley

$$\begin{array}{llcll} m_1: & w_1, w_2, w_3 & \quad & w_1: & m_3, m_2, m_1 \\ m_2: & w_2, w_3, w_1 & & w_2: & m_2, m_1, m_3 \\ m_3: & w_3, w_1, w_2 & & w_3: & m_1, m_3, m_2 \end{array}$$

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance? Yes, always!
- Is stable marriage unique? No, there is a range of stable matchings; can be unique in some cases.
- Can it be computed efficiently?

Stable marriage problem

Remarkable result of Gale and Shapley

m_1 :	w_1, w_2, w_3	w_1 :	m_3, m_2, m_1
m_2 :	w_2, w_3, w_1	w_2 :	m_2, m_1, m_3
m_3 :	w_3, w_1, w_2	w_3 :	m_1, m_3, m_2

Goal: To compute a matching that is "stable".

- Does a stable marriage exist in any instance? Yes, always!
- Is stable marriage unique? No, there is a range of stable matchings; can be unique in some cases.
- Can it be computed efficiently? Yes, in $O(n^2)$ time.

Stable marriage algorithm

$$\begin{array}{ll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{ll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Gale and Shapley algorithm

- set all men and women as unengaged.
- while there exists an unengaged man m
  1. m proposes to the most preferred woman w to whom he has not yet proposed.
  2. w accepts if either she is unengaged or she is engaged to m' and w prefers m to m'.

Stable marriage algorithm

m_1 :	w_1, w_2, w_3	w_1 :	m_3, m_2, m_1
m_2 :	w_2, w_3, w_1	w_2 :	m_2, m_1, m_3
m_3 :	w_3, w_1, w_2	w_3 :	m_1, m_3, m_2

Gale and Shapley algorithm

- set all men and women as unengaged.
- while there exists an unengaged man m
  1. m proposes to the most preferred woman w to whom he has not yet proposed.
  2. w accepts if either she is unengaged or she is engaged to m' and w prefers m to m'.

Questions:

- Does the algorithm even terminate?

Stable marriage algorithm

$$\begin{array}{ll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{ll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Gale and Shapley algorithm

- set all men and women as unengaged.
- while there exists an unengaged man m
  1. m proposes to the most preferred woman w to whom he has not yet proposed.
  2. w accepts if either she is unengaged or she is engaged to m' and w prefers m to m'.

Questions:

- Does the algorithm even terminate?
- Why does it output a stable marriage?

Stable marriage algorithm

$$\begin{array}{ll} m_1: & w_1, w_2, w_3 \\ m_2: & w_2, w_3, w_1 \\ m_3: & w_3, w_1, w_2 \end{array} \qquad \begin{array}{ll} w_1: & m_3, m_2, m_1 \\ w_2: & m_2, m_1, m_3 \\ w_3: & m_1, m_3, m_2 \end{array}$$

Gale and Shapley algorithm

- set all men and women as unengaged.
- while there exists an unengaged man m
  1. m proposes to the most preferred woman w to whom he has not yet proposed.
  2. w accepts if either she is unengaged or she is engaged to m' and w prefers m to m'.

Questions:

- Does the algorithm even terminate?
- Why does it output a stable marriage?
- How does the ordering of men in the while loop matter?

Stable marriage algorithm

- A surprisingly simple algorithm that is guaranteed to produce a stable matching.
- A rich structure underlying the problem.

Stable marriage algorithm

- A surprisingly simple algorithm that is guaranteed to produce a stable matching.
- A rich structure underlying the problem.
  Has been dealt in two books – one by Irving and Gusfield and a recent one by Manlove.
- National Residency Matching program – one of the most important applications amongst several others.
- Pioneering work by Roth and Shapley won the 2012 Nobel prize for Economics.

Summary

- Matchings, definitions, augmenting/alternating paths.
- A template for finding matchings by iterative improvement.
- An efficient algorithm in the bipartite case.
- Stable marriage algorithm and properties.

Thank You!