Introduction to Popular Matchings

Kavitha Telikepalli

(Tata Institute of Fundamental Research, Mumbai)

Laboratoire d’Informatique de Paris-Nord (LIPN)
Université Sorbonne Paris Nord
The input

A bipartite graph where every vertex has a strict ranking of its neighbors.

A well-studied model used in many two-sided markets:

▶ students to schools;
▶ medical residents to hospitals.

What we seek is a matching in this graph.
A matching is a subset of edges such that at most one edge is incident to any vertex.

Recall that vertices have preferences.

- Our problem is to find an optimal matching as per vertex preferences.
A matching $M$ is **stable** if there is no edge $ab$ such that:

$$b \succ_a M(a) \quad \text{and} \quad a \succ_b M(b)$$

(i.e., $a$ and $b$ prefer each other to their respective assignments in $M$)

- The red matching is stable but the blue one is not.
Stable matchings

Do stable matchings always exist? Can we find one efficiently?

▶ Yes [Gale and Shapley, 1962].
Stable matchings

In assigning new doctors to hospitals around the US.
In helping kidney transplant patients find a match.

https://medium.com/@UofCalifornia/how-a-matchmaking-algorithm-saved-lives-2a65ac448698
Do stable matchings always exist? Can we find one efficiently?

► Yes [Gale and Shapley, 1962].

The Gale-Shapley algorithm: agents propose and jobs dispose — this is a very simple and clean algorithm.

Let us run Gale-Shapley algorithm on this instance.
Stable matchings

Do stable matchings always exist? Can we find one efficiently?

- Yes [Gale and Shapley, 1962].

The Gale-Shapley algorithm: agents propose and jobs dispose — this is a very simple and clean algorithm.

Initially both $a$ and $s$ propose to their top neighbor $b$. 
Stable matchings

Do stable matchings always exist? Can we find one efficiently?

- Yes [Gale and Shapley, 1962].

The Gale-Shapley algorithm: agents propose and jobs dispose — this is a very simple and clean algorithm.

\[ a \rightarrow 1 \rightarrow 2 \rightarrow b \]

\[ s \rightarrow 2 \rightarrow s \]

\[ b \] (tentatively) accepts \( s \)'s proposal and rejects \( a \)'s proposal.
Stable matchings

Do stable matchings always exist? Can we find one efficiently?

- Yes [Gale and Shapley, 1962].

The Gale-Shapley algorithm: agents propose and jobs dispose — this is a very simple and clean algorithm.

```
a  1 2  b
```

```
s  2 1  t
```

$a$ has no other neighbor to propose to; we get the matching $\{sb\}$. 
Applications of stable matchings

Stable matchings are used in several problems in economics, computer science, and operations research.

To match students to schools in New York:


To match students to colleges in France:

- Stable Matching in Practice, Claire Mathieu. ESA 2018, Keynote talk.

To match students to engineering colleges in India:

Size versus Stability

All stable matchings match the same subset of vertices [Rural Hospitals Theorem].

- The size of a stable matching could be only half the size of a maximum matching.

The maximum matching \( \{ab, st\} \) is unstable.

- We seek large matchings in all applications.
- Forbidding blocking edges constrains the size of the matching.
Beyond stability

Drawbacks of stability:

- Size can be half the size of a maximum matching;
- Models a situation where every edge has a "veto power".

Can we relax stability so as to cope with these issues? We want a set that:

- contains stability as a special case;
- shifts the focus from "veto power" to "collective decision";
- allows for matchings of size larger than stable matchings.
Beyond stability

Drawbacks of stability:

▶ Size can be half the size of a maximum matching;
▶ Models a situation where every edge has a “veto power”.

Can we relax stability so as to cope with these issues? We want a set that:

▶ contains stability as a special case;
▶ shifts the focus from “veto power” to “collective decision”;
▶ allows for matchings of size larger than stable matchings.

⇒ Popular matchings
Elections between pairs of matchings

Any pair of matchings can be compared via a pairwise election.

Consider the election between the red and green matchings. 

- the red vs blue election is a tie (so red $\sim$ blue).
- the green matching loses this election, thus red $\succ$ green.

A popular matching is one that does not lose any election.
Elections between pairs of matchings

Any pair of matchings can be compared via a pairwise election.

- the red vs blue election is a tie (so red $\sim$ blue).

Consider the election between the red and green matchings.

- the green matching loses this election, thus red $\succ$ green.

A popular matching is one that does not lose any election.
Elections between pairs of matchings

Any pair of matchings can be compared via a pairwise election.

- the red vs blue election is a tie (so red $\sim$ blue).

Consider the election between the red and green matchings.
Elections between pairs of matchings

Any pair of matchings can be compared via a pairwise election.

▶ the red vs blue election is a tie (so red \( \sim \) blue).

Consider the election between the red and green matchings.

▶ the green matching loses this election, thus red \( \succ \) green.
Any pair of matchings can be compared via a pairwise election.

- the red vs blue election is a tie (so red $\sim$ blue).

Consider the election between the red and green matchings.

- the green matching loses this election, thus red $\succ$ green.

A popular matching is one that does not lose any election.
**Condorcet winner**: A candidate who defeats every other candidate in their head-to-head election.

<table>
<thead>
<tr>
<th></th>
<th>30%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
<td>c</td>
<td>b</td>
</tr>
</tbody>
</table>

- Here $a$ is the Condorcet winner.
- $a \succ b$ and $a \succ c$. (*$a$ defeats $b$ and $a$ defeats $c$*)
Condorcet winner: A candidate who defeats every other candidate in their head-to-head election.

<table>
<thead>
<tr>
<th></th>
<th>30%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
<td>c</td>
<td>b</td>
</tr>
</tbody>
</table>

- Here a is the Condorcet winner.
- $a \succ b$ and $a \succ c$. ($a$ defeats $b$ and $a$ defeats $c$)
Condorcet winner: A candidate who defeats every other candidate in their head-to-head election.

Here $a$ is the Condorcet winner.

$a \succ b$ and $a \succ c$. ($a$ defeats $b$ and $a$ defeats $c$)
A weak Condorcet winner is one that is never defeated.

- $x$ is a weak Condorcet winner $\implies x \succ y$ or $x \sim y$ for all candidates $y$. 
A weak Condorcet winner is one that is never defeated.

- $x$ is a weak Condorcet winner $\implies x \succ y$ or $x \sim y$ for all candidates $y$.

However a (weak) Condorcet winner need not always exist.

<table>
<thead>
<tr>
<th></th>
<th>33.3%</th>
<th>33.3%</th>
<th>33.3%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a$</td>
<td>$b$</td>
<td>$c$</td>
</tr>
<tr>
<td>2</td>
<td>$b$</td>
<td>$c$</td>
<td>$a$</td>
</tr>
<tr>
<td>3</td>
<td>$c$</td>
<td>$a$</td>
<td>$b$</td>
</tr>
</tbody>
</table>

- Here we have: $a \succ b \succ c \succ a$. 
A weak Condorcet winner is one that is never defeated.

- $x$ is a weak Condorcet winner $\implies x \succ y$ or $x \sim y$ for all candidates $y$.

However a (weak) Condorcet winner need not always exist.

<table>
<thead>
<tr>
<th></th>
<th>33.3%</th>
<th>33.3%</th>
<th>33.3%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a$</td>
<td>$b$</td>
<td>$c$</td>
</tr>
<tr>
<td>2</td>
<td>$b$</td>
<td>$c$</td>
<td>$a$</td>
</tr>
<tr>
<td>3</td>
<td>$c$</td>
<td>$a$</td>
<td>$b$</td>
</tr>
</tbody>
</table>

- Here we have: $a \succ b \succ c \succ a$. 
Weak Condorcet winner

A weak Condorcet winner is one that is never defeated.

▶ $x$ is a weak Condorcet winner $\implies x \succ y$ or $x \sim y$ for all candidates $y$.

However a (weak) Condorcet winner need not always exist.

<table>
<thead>
<tr>
<th></th>
<th>33.3%</th>
<th>33.3%</th>
<th>33.3%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a$</td>
<td>$b$</td>
<td>$c$</td>
</tr>
<tr>
<td>2</td>
<td>$b$</td>
<td>$c$</td>
<td>$a$</td>
</tr>
<tr>
<td>3</td>
<td>$c$</td>
<td>$a$</td>
<td>$b$</td>
</tr>
</tbody>
</table>

▶ Here we have: $a \succ b \succ c \succ a$. 
A weak Condorcet winner is one that is never defeated.

$x$ is a weak Condorcet winner $\iff x \succ y$ or $x \sim y$ for all candidates $y$.

However a (weak) Condorcet winner need not always exist.

Here we have: $a \succ b \succ c \succ a$. 
Weak Condorcet winner in our setting

Matching \( M \) is a weak Condorcet winner \( \equiv M \succ N \) or \( M \sim N \) for all matchings \( N \).

- Do weak Condorcet winners always exist in our setting?

Every stable matching is a weak Condorcet winner [Gärdenfors, 1975].

Comparing a stable matching \( S \) with any matching \( N \):

- \( u \) prefers \( N \) to \( S \) \( \implies \) \( N(u) \) has to prefer \( S \) to \( N \);

  (otherwise the edge between \( u \) and \( N(u) \) blocks \( S \))
Matching $M$ is a weak Condorcet winner $\equiv M \succ N$ or $M \sim N$ for all matchings $N$.

- Do weak Condorcet winners always exist in our setting?

Every stable matching is a weak Condorcet winner [Gärdenfors, 1975].

Comparing a stable matching $S$ with any matching $N$:

- $u$ prefers $N$ to $S \implies N(u)$ has to prefer $S$ to $N$;
  (otherwise the edge between $u$ and $N(u)$ blocks $S$)
- so the number of votes for $N \leq$ the number of votes for $S$.  

T. Kavitha
Introduction to Popular Matchings
Matching $M$ is a weak Condorcet winner if $M \succ N$ or $M \sim N$ for all matchings $N$.

- Do weak Condorcet winners always exist in our setting?

Every stable matching is a weak Condorcet winner [Gärdenfors, 1975].

Comparing a stable matching $S$ with any matching $N$:

- $u$ prefers $N$ to $S \implies N(u)$ has to prefer $S$ to $N$;
  (otherwise the edge between $u$ and $N(u)$ blocks $S$)
- so the number of votes for $N \leq$ the number of votes for $S$.

Matchings that are weak Condorcet winners = Popular matchings.
Popular matchings

Properties of popular matchings:

▶ contains stability as a special case;
▶ shifts the focus from “veto power” to “collective decision”; ✓
▶ allows for matchings of size larger than stable matchings.
Popular matchings

Properties of popular matchings:

▶ contains stability as a special case; ✓

▶ shifts the focus from “veto power” to “collective decision”; ✓

▶ allows for matchings of size larger than stable matchings.

Every stable matching is popular [Gärdenfors, 1975].
Popular matchings

Properties of popular matchings:

- contains stability as a special case; ✓
- shifts the focus from “veto power” to “collective decision”; ✓
- allows for matchings of size larger than stable matchings. ✓

Every stable matching is popular [Gärdenfors, 1975].

- Stable matchings are min-size popular matchings.
Properties of popular matchings:

- contains stability as a special case; ✓
- shifts the focus from “veto power” to “collective decision”; ✓
- allows for matchings of size larger than stable matchings. ✓

Every stable matching is popular [Gärdenfors, 1975].

- Stable matchings are min-size popular matchings.

Is there an efficient algorithm to find a max-size popular matching?
An interesting example

There is a popular matching of size 2 and there is also one of size 4.

But there is no popular matching of size 3 here.
An interesting example

There is a popular matching of size 2 and there is also one of size 4.

But there is no popular matching of size 3 here.

So the following iterative approach — have a popular matching of size $i$ and use this popular matching to build one of size $i + 1$ — will not work.
To find a max-size popular matching, can we adapt the Gale-Shapley algorithm?

▶ Stability is easy to check: no edge blocks a stable matching.

▶ Popularity requires comparing our matching with all the matchings in $G$. 

Our goal is to find the matching $\{ab, st\}$ of size 2 via the Gale-Shapley algorithm.
To find a max-size popular matching

To find a max-size popular matching, can we adapt the Gale-Shapley algorithm?

- Stability is easy to check: no edge blocks a stable matching.
- Popularity requires comparing our matching with all the matchings in $G$.

Suppose $G$ is our earlier example.

Our goal is to find the matching $\{ab, st\}$ of size 2 via the Gale-Shapley algorithm.

- This is a max-size popular matching in $G$. 
A new instance $G'$

A new graph $G'$ such that $\{ab, st\}$ is the stable matching in $G'$?

Suppose we replace every edge $uv$ in $G$ by the pair of edges $uv$ and $uv$ in $G'$:

- that is, by two parallel edges: one red and the other blue.

The corresponding graph $G'$ is:

- Every vertex on the left prefers any red edge to any blue edge.
- Every vertex on the right prefers any blue edge to any red edge.
A new instance $G'$

So the graph $G'$ with preferences is:

![Graph diagram]

- The preference order of $s$ in $G$ is $b \succ t$. Its preference order in $G'$ is:
  \[ b \succ t \succ b \succ t. \]

- The preference order of $b$ in $G$ is $s \succ a$. Its preference order in $G'$ is:
  \[ s \succ a \succ s \succ a. \]
A new instance $G'$

The graph $G'$ with preferences is:

![Graph](image)

Recall the stable matching $\{sb\}$ in $G$.

- In the graph $G'$, neither $\{sb\}$ nor $\{sb\}$ is stable.
  - The edge $ab$ blocks the matching $\{sb\}$.
  - The edge $st$ blocks the matching $\{sb\}$.
Computing a stable matching in $G'$

Let us run Gale-Shapley algorithm in $G'$.

![Graph diagram]

- Both $a$ and $s$ propose to $b$ along their red edges.
- $b$ prefers $s$'s proposal to $a$'s proposal.
Let us run Gale-Shapley algorithm in $G'$.

- So $b$ (tentatively) accepts $s$'s proposal and rejects $a$'s proposal.
- Then $a$ proposes along its next favorite edge: this is $ab$. 

![Diagram showing the Gale-Shapley algorithm process](diagram.png)
Computing a stable matching in $G'$

Let us run Gale-Shapley algorithm in $G'$.

Let $ab$ be a matching in $G'$. Suppose $s$ proposes to $b$.

Observe that now $b$ prefers $a$'s proposal to $s$'s proposal.

So $b$ (tentatively) accepts $a$'s proposal and rejects $s$'s proposal.
Computing a stable matching in $G'$

Let us run Gale-Shapley algorithm in $G'$.

Then $s$ proposes along its next most favorite edge $st$.

$t$ (tentatively) accepts $s$’s proposal. This is the end of the algorithm.
Computing a stable matching in $G'$

So we get the stable matching $\{ab, st\}$ in $G'$.

Ignoring colors, this is the desired matching $M = \{ab, st\}$ in $G$.

---

Our algorithm in $G = (A \cup B, E)$

- Construct the red/blue graph $G' = (A \cup B, E')$.
- Run Gale-Shapley algorithm in $G'$ to compute $M'$.
- Return the corresponding matching $M$ in $G$. 

---
Computing a stable matching in $G'$

So we get the stable matching $\{ab, st\}$ in $G'$.

Ignoring colors, this is the desired matching $M = \{ab, st\}$ in $G$.

---

**Our algorithm in $G = (A \cup B, E)$**

- Construct the red/blue graph $G' = (A \cup B, E')$.
- Run Gale-Shapley algorithm in $G'$ to compute $M'$.
- Return the corresponding matching $M$ in $G$.

---

**Claim.** $M$ is a max-size popular matching in $G$.

- We use linear programming to prove the popularity of $M$. 
Analyzing our algorithm

Every popular matching admits a simple certificate of its popularity.

- The certificate for $M$ is given by red/blue edge colours in the matching $M'$. 

Analyzing our algorithm

Every popular matching admits a simple certificate of its popularity.

- The certificate for $M$ is given by red/blue edge colours in the matching $M'$.

Let us define an edge weight function in $G$. For any edge $ab$:

$$\text{wt}_M(ab) = \text{vote}_a(b, M(a)) + \text{vote}_b(a, M(b)).$$

Here $\text{vote}_v(u, u') = \begin{cases} 
1 & \text{if } v \text{ prefers } u \text{ to } u' \\
-1 & \text{if } v \text{ prefers } u' \text{ to } u \\
0 & \text{otherwise.}
\end{cases}$

So $\text{wt}_M(e) \in \{0, \pm2\}$ for any edge $e$.

- Observation. For any edge $e$, $\text{wt}_M(e) = 2 \iff e$ is a blocking edge to $M$. 

T. Kavitha  
Introduction to Popular Matchings
An appropriate edge weight function

Let us augment $G$ with self-loops:

- any matching $\Rightarrow$ a perfect matching via self-loops.

For any self-loop $uu$:

$$\text{let } wt_{M}(uu) = \text{vote}_{u}(u, M(u)) = \begin{cases} 0 & \text{if } M(u) = u \\ -1 & \text{otherwise.} \end{cases}$$
An appropriate edge weight function

Let us augment \( G \) with self-loops:

- any matching \( \rightsquigarrow \) a perfect matching via self-loops.

For any self-loop \( uu \):

\[
\text{let } wt_M(uu) = \text{vote}_u(u, M(u)) = \begin{cases} 
0 & \text{if } M(u) = u \\
-1 & \text{otherwise.}
\end{cases}
\]

Observation. For any perfect matching \( N \):

\[
wt_M(N) = \# \text{ of votes for } N - \# \text{ of votes for } M.
\]

- \( M \) is popular \( \iff \) \( wt_M(N) \leq 0 \) for any perfect matching \( N \).

\( \iff \) any perfect matching in \( G \) with edge weights given by \( wt_M \) has weight at most 0.
LP for max-weight perfect matching

\[
\max \sum_{e} \text{wt}_M(e) \cdot x_e
\]

\[
\sum_{e \in \delta(u) \cup \{uu\}} x_e = 1 \quad \forall u \in A \cup B
\]

\[
x_e \geq 0 \quad \forall e \in E \cup \{\text{self-loops}\}.
\]

\(M\) is popular \iff the optimal value of this LP is at most 0.
LP for max-weight perfect matching

\[
\max \sum_e \text{wt}_M(e) \cdot x_e
\]

\[
\sum_{e \in \delta(u) \cup \{uu\}} x_e = 1 \quad \forall u \in A \cup B
\]

\[
x_e \geq 0 \quad \forall e \in E \cup \{\text{self-loops}\}.
\]

\(M\) is popular \iff the optimal value of this LP is at most 0.

Dual LP

\[
\min \sum_u \alpha_u
\]

\[
\alpha_a + \alpha_b \geq \text{wt}_M(ab) \quad \forall \, ab \in E
\]

\[
\alpha_u \geq \text{wt}_M(uu) \quad \forall \, u \in A \cup B.
\]

\(M\) is popular \iff the optimal value of the dual LP is at most 0.
Dual certificate

Every stable matching $S$ has a simple dual certificate: $\vec{\alpha} = \vec{0}$.

- This is because $\text{wt}_S(e) \leq 0$ for all edges $e$.

Does $M$ computed by our algorithm have an easy-to-describe dual certificate?
Dual certificate

Every stable matching $S$ has a simple dual certificate: $\vec{a} = \vec{0}$.

- This is because $\text{wt}_S(e) \leq 0$ for all edges $e$.

Does $M$ computed by our algorithm have an easy-to-describe dual certificate?

For each vertex $a \in A$:
- $a$ is matched along a red edge in $M'$: set $\alpha_a = 1$.
- $a$ is matched along a blue edge in $M'$: set $\alpha_a = -1$.
- $a$ is unmatched in $M'$: set $\alpha_a = 0$.

For each vertex $b \in B$:
- $b$ is matched along a red edge in $M'$: set $\alpha_b = -1$.
- $b$ is matched along a blue edge in $M'$: set $\alpha_b = 1$.
- $b$ is unmatched in $M'$: set $\alpha_b = 0$. 

T. Kavitha
Introduction to Popular Matchings
Dual certificate

A useful picture:

So vertices matched along red edges are in $A_0 \cup B_0$.
And vertices matched along blue edges are in $A_1 \cup B_1$.

- Unmatched vertices of $A$ (resp., $B$) are in $A_1$ (resp., $B_0$).

$\alpha$-values were assigned as follows:

- $\alpha_u = 1$ for all $u \in A_0 \cup B_1$;
- $\alpha_u = -1$ for all matched $u \in A_1 \cup B_0$;
- $\alpha_u = 0$ for all unmatched $u$. 
Dual feasibility of $\vec{\alpha}$

We need to show this vector $\vec{\alpha}$ is a feasible solution to the dual LP.

**Dual LP**

$$\min \sum_{u} \alpha_u$$

$$\alpha_a + \alpha_b \geq \text{wt}_M(ab) \quad \forall \ ab \in E$$

$$\alpha_u \geq \text{wt}_M(uu) \quad \forall \ u \in A \cup B.$$  

We will also show that $\sum_{u \in A \cup B} \alpha_u = 0$.

- This will mean the dual optimal solution is at most 0.
- This will prove $M$ is a popular matching.
Dual feasibility of $\vec{\alpha}$

Recall that $\alpha_u \in \{0, \pm 1\}$:

\[ A_1 \quad \vdots \quad B_1 \]
\[ A_0 \quad \vdots \quad B_0 \]

**Observation.** *The constraint $\alpha_u \geq \text{wt}_M(uu)$ holds for all vertices $u$.***

- For a matched vertex $u$, we have $\alpha_u \geq -1 = \text{wt}_M(uu)$.
- For an unmatched vertex $u$, we have $\alpha_u = 0 = \text{wt}_M(uu)$. 
Dual feasibility of $\bar{\alpha}$

Recall that $\alpha_u \in \{0, \pm 1\}$:

Observation. The constraint $\alpha_u \geq \text{wt}_M(uu)$ holds for all vertices $u$.

- For a matched vertex $u$, we have $\alpha_u \geq -1 = \text{wt}_M(uu)$.
- For an unmatched vertex $u$, we have $\alpha_u = 0 = \text{wt}_M(uu)$.

Lemma. The constraint $\alpha_a + \alpha_b \geq \text{wt}_M(ab)$ holds for all $ab \in E$.

- We will use the stability of $M'$ in the instance $G'$ to prove the lemma.

Conclusion. So $\bar{\alpha}$ is dual-feasible.
Optimal value of the dual LP

Every edge in $M'$ is a red edge or a blue edge.

- So $\alpha_a + \alpha_b = 0$ for all $ab \in M$.
- Since $\alpha_u = 0$ for all unmatched vertices, $\sum_{u \in A \cup B} \alpha_u = 0$.

Thus the optimal value of the dual LP is at most 0.

- Hence $M$ is a popular matching.
Proof of the lemma

To show $\alpha_a + \alpha_b \geq wt_M(ab)$ holds for all $ab \in E$.

Case 1. Suppose $\alpha_a = \alpha_b = -1$.

So $ac \in M'$ and $bd \in M'$ for some neighbors $c$ and $d$ of $a$ and $b$, respectively.

Observe that (i) $a$ prefers $c$ to $b$ and (ii) $b$ prefers $d$ to $a$.

This is because $a$ never proposed along $ab$.

Furthermore, $b$ rejected $a$'s proposal along $ab$.

Thus $wt_M(ab) = -2$, hence $\alpha_a + \alpha_b = -2 = wt_M(ab)$. 
Proof of the lemma

Case 2. Suppose \( \alpha_a = \alpha_b = 1 \).

\[ \begin{align*}
\text{Since } wt_M(ab) &\in \{0, \pm 2\}, \\
\text{we have } \alpha_a + \alpha_b &\ge 2 \ge wt_M(ab).
\end{align*} \]

Case 3. Suppose \( \alpha_a = 1 \) and \( \alpha_b = -1 \).

\[ \begin{align*}
\text{This means } ac \text{ and } bd \text{ are in } M' \text{ for some neighbors } c \text{ and } d. \\
\text{This means } ac \text{ and } bd \text{ are in } M' \text{ for some neighbors } c \text{ and } d. \\
\text{This means } ac \text{ and } bd \text{ are in } M' \text{ for some neighbors } c \text{ and } d. \\
\text{Thus } wt_M(ab) &\le 0, \text{ hence } \alpha_a + \alpha_b = 0 \ge wt_M(ab).
\end{align*} \]

Case 4. Suppose \( \alpha_a = -1 \) and \( \alpha_b = 1 \).

\[ \begin{align*}
\text{This means } ac \text{ and } bd \text{ are in } M' \text{ for some neighbors } c \text{ and } d. \\
\text{Thus } wt_M(ab) &\le 0, \text{ hence } \alpha_a + \alpha_b = 0 \ge wt_M(ab).
\end{align*} \]
Proof of the lemma

Case 5. Suppose $\alpha_a = 0$.

Since $M'$ is stable in $G'$, $ab$ does not block $M'$.

- This means $bd \in M'$ for some neighbor $d$ that $b$ prefers to $a$.

Thus $\alpha_b = 1$, hence $\alpha_a + \alpha_b = 1 \geq 0 = \text{wt}_M(ab)$.

An analogous analysis holds when $\alpha_b = 0$.

- Then $\alpha_a = 0$ and $\alpha_b = 1$, so $\alpha_a + \alpha_b = 1 \geq 0 = \text{wt}_M(ab)$.

This finishes the proof of the lemma.
Proof of the lemma

Case 5. Suppose $\alpha_a = 0$.

Since $M'$ is stable in $G'$, $ab$ does not block $M'$.

$\blacktriangleright$ This means $bd \in M'$ for some neighbor $d$ that $b$ prefers to $a$.

Thus $\alpha_b = 1$, hence $\alpha_a + \alpha_b = 1 \geq 0 = wt_M(ab)$.

An analogous analysis holds when $\alpha_b = 0$.

$\blacktriangleright$ Then $\alpha_a = 0$ and $\alpha_b = 1$, so $\alpha_a + \alpha_b = 1 \geq 0 = wt_M(ab)$.

This finishes the proof of the lemma.

A useful observation

For any edge $ab$ incident to an unmatched vertex (either $a$ or $b$ is unmatched):

$\blacktriangleright$ we have $\alpha_a + \alpha_b = 1 > 0 = wt_M(ab)$, thus the edge $ab$ is slack.
The dual LP and slack edges

\[
\min \sum_{u} \alpha_u
\]

\[
\alpha_a + \alpha_b \geq wt_M(ab) \quad \forall \, ab \in E
\]

\[
\alpha_u \geq wt_M(uu) \quad \forall \, u \in A \cup B.
\]

Recall that \(\overline{\alpha}\) is an optimal solution to the dual LP.

**Complementary Slackness**

Any matching \(N\) with a slack edge is not an optimal solution to the primal LP;

- in other words, \(wt_M(N) < 0\) (equivalently, \(M\) defeats \(N\)).

Thus any matching larger than \(M\) is unpopular.

- So \(M\) is a max-size popular matching.

Thus there is a linear time algorithm to find a max-size popular matching.
Claim. There is no length 3 augmenting path wrt $M$ in $G$.

$a - b - s - t$ is an augmenting path wrt $M \implies$ either $ab$ or $st$ blocks $M'$
(a contradiction to $M''$'s stability in $G'$)

Hence any augmenting path in $M \oplus M_{\max}$ has length $\geq 5$.

Thus $|M| \geq \frac{2}{3} \cdot |M_{\max}|$.

There are simple examples where $|M| = 2$ and $|M_{\max}| = 3$. 
Maximum matchings

Applications where the size of the matching is more important than vertex preferences:

- matching medical students to hospitals for residency;
- matching doctors to hospitals in a pandemic;
- assigning accommodation to sailors.

Here \( \text{\{admissible solutions\}} = \text{\{maximum matchings\}} \).

The goal is to find a best maximum matching as per vertex preferences.

- How about a maximum matching with the minimum number of blocking edges?
  - Finding such a matching is NP-hard [Biro, Manlove, and Mittal, 2010].

- How about a maximum matching that is popular?
It can be the case that no maximum matching is popular.

How about a maximum matching $M$ that is popular among maximum matchings?

- $M$ is a maximum matching.
- Furthermore, $M \succ N$ or $M \sim N$ for all maximum matchings $N$.

Does such a "popular maximum matching" always exist in $G$?

Furthermore, is it easy to find one?
Maximum matchings and popularity

It can be the case that no maximum matching is popular.

How about a maximum matching $M$ that is popular among maximum matchings?

▶ So $M$ is a maximum matching.

▶ Furthermore, $M \succ N$ or $M \sim N$ for all maximum matchings $N$. 
Maximum matchings and popularity

It can be the case that no maximum matching is popular.

How about a maximum matching $M$ that is popular among maximum matchings?

- So $M$ is a maximum matching.
- Furthermore, $M \succ N$ or $M \sim N$ for all maximum matchings $N$.

Does such a “popular maximum matching” always exist in $G$?

- Furthermore, is it easy to find one?
More colorful graphs

Suppose we use $n$ colors, where $|A| = n$. Call the resulting graph $G^*$. Every edge $ab$ in $G$ has $n$ parallel copies in $G^*$: $ab, ab, \ldots, ab, \ldots, ab, ab$.

For any vertex on the left:

$$
\text{red} \succ \text{blue} \succ \cdots \succ \text{green} \succ \cdots \succ \text{magenta} \succ \text{cyan}.
$$

For any vertex on the right:

$$
\text{cyan} \succ \text{magenta} \succ \cdots \succ \text{green} \succ \cdots \succ \text{blue} \succ \text{red}.
$$

Within any color class, every vertex maintains its original preference order $\succ$. 
An extension of our algorithm

- Construct the colorful graph $G^* = (A \cup B, E^*)$.
- Run Gale-Shapley algorithm in $G^*$ to compute $M^*$.
- Return the corresponding matching $M$ in $G$.

Claim 1. $M$ is a maximum matching in $G$.

Claim 2. $M \succ N$ or $M \sim N$ for every maximum matching $N$ in $G$.

Claims 1 and 2 $\Rightarrow$ $M$ is a popular maximum matching.

Moreover, such a matching can be computed easily.
Recall the following edge weight function $w_t_M$ in $G$. For any edge $ab$:

$$w_t_M(ab) = \text{vote}_a(b, M(a)) + \text{vote}_b(a, M(b)).$$

Here $\text{vote}_v(u, u') = \begin{cases} 
1 & \text{if } v \text{ prefers } u \text{ to } u' \\
-1 & \text{if } v \text{ prefers } u' \text{ to } u \\
0 & \text{otherwise.}
\end{cases}$

So $w_t_M(e) \in \{0, \pm 2\}$ for any edge $e$.

Let $M$ be a maximum matching in $G$.

**Observation.** $w_t_M(N) \leq 0$ for all maximum matchings $N$

$\Rightarrow M$ is a popular maximum matching in $G$. 
The LP method

LP for max-weight maximum matching in $G$:

$$\max \sum_{e} \text{wt}_M(e) \cdot x_e$$

$$\sum_{e \in \delta(u)} x_e \leq 1 \quad \forall u \in A \cup B$$

$$\sum_{a \in A} \sum_{e \in \delta(a)} x_e = k \quad \text{and} \quad x_e \geq 0 \quad \forall e \in E.$$ 

Here $k$ is the size of a maximum matching in $G$.

Optimal value of this LP is at most 0 $\Rightarrow$ $\text{wt}_M(N) \leq 0$ for all maximum matchings $N$ $\Rightarrow$ $M$ is a popular maximum matching in $G$. 

The dual LP

Dual LP

\[
\begin{align*}
\min & \quad k \cdot z + \sum_u \alpha_u \\
\alpha_a + \alpha_b + z & \geq \text{wt}_M(ab) \quad \forall \, ab \in E \\
\alpha_u & \geq 0 \quad \forall \, u \in A \cup B.
\end{align*}
\]

Our goal is to show that the optimal value of the dual LP is at most 0.

- Thus our goal is to show a dual feasible solution \((\vec{\alpha}, z)\) such that
  \[
  k \cdot z + \sum_u \alpha_u = 0.
  \]

- Recall the colorful graph \(G^*\):
  - let color 0, color 1, \ldots, color \(n - 1\) denote the \(n\) colors (here \(n = |A|\)).
A partition of the vertex set $A \cup B$

For $0 \leq i \leq n - 1$, let $A_i = \{a \in A : a$ is matched along a color $i$ edge in $M^*\}$.

For $0 \leq i \leq n - 1$, let $B_i = \{b \in B : b$ is matched along a color $i$ edge in $M^*\}$.

Unmatched vertices of $A$ are in $A_{n-1}$ and unmatched vertices of $B$ are in $B_0$. 

\begin{center}
\begin{tikzpicture}
    \draw[very thick, color=blue] (0,0) -- (0,4);
    \draw[very thick, color=purple] (1,0) -- (1,4);
    \draw[very thick, color=green] (2,0) -- (2,4);
    \draw[very thick, color=red] (3,0) -- (3,4);
    \draw[very thick, color=teal] (4,0) -- (4,4);

    \node at (0,0) [below] {$A_0$};
    \node at (1,0) [below] {$A_1$};
    \node at (2,0) [below] {$A_{n-2}$};
    \node at (3,0) [below] {$A_{n-1}$};
    \node at (4,0) [below] {$A_n$};

    \node at (0,4) [above] {$B_0$};
    \node at (1,4) [above] {$B_1$};
    \node at (2,4) [above] {$B_{n-2}$};
    \node at (3,4) [above] {$B_{n-1}$};

    \draw[very thick, dashed, color=blue] (0,0) -- (4,4);
    \draw[very thick, dashed, color=purple] (1,0) -- (4,4);
    \draw[very thick, dashed, color=green] (2,0) -- (4,4);
    \draw[very thick, dashed, color=red] (3,0) -- (4,4);

    \node at (0,0) [below] {$A_0$};
    \node at (0,4) [above] {$B_0$};
    \node at (1,0) [below] {$A_1$};
    \node at (1,4) [above] {$B_1$};
    \node at (2,0) [below] {$A_{n-2}$};
    \node at (2,4) [above] {$B_{n-2}$};
    \node at (3,0) [below] {$A_{n-1}$};
    \node at (3,4) [above] {$B_{n-1}$};
    \node at (4,0) [below] {$A_n$};
    \node at (4,4) [above] {$B_n$};
\end{tikzpicture}
\end{center}
A partition of the vertex set $A \cup B$

The following properties hold due to the stability of $M^*$ in $G^*$:

1. For any $i$, the matching $M$ restricted to $A_i \cup B_i$ is stable.
2. For any edge $ab$ where $a \in A_{i+1}$ and $b \in B_i$: $\text{wt}_M(ab) = -2$.
3. $G$ has no edge in $A_i \times B_j$ where $i \geq j + 2$.
4. There is no augmenting path with respect to $M$. 

T. Kavitha
Introduction to Popular Matchings
A dual certificate

Property (4) implies that $M$ is a maximum matching in $G$.

For $0 \leq i \leq n - 1$:

- $a \in A_i \Rightarrow \text{set } \alpha_a = 2(n - 1) - 2i$;
- $b \in B_i \Rightarrow \text{set } \alpha_b = 2i$.

  - so $\alpha_u = 0$ for any $u \in A_{n-1} \cup B_0$.

Set $z = -2(n - 1)$.

Properties (1)-(3) allow us to prove the dual-feasibility of $\bar{\alpha}$.

$\alpha_a + \alpha_b + z = 2(n - 1) - 2i + 2i - 2(n - 1) = 0$ for each $ab \in M$.

(because $a \in A_i$ and $b \in B_i$ for some $i \in \{0, \ldots, n - 1\}$)

- Hence $k \cdot z + \sum_u \alpha_u = \sum_{ab \in M}(\alpha_a + \alpha_b + z) = 0$.

  (since $\alpha_u = 0$ for unmatched $u$)
Popular maximum matchings

Interestingly, *every* popular maximum matching occurs as a stable matching in the *colorful* graph $G^*$.

- So popular maximum matchings are very well-structured.

Max-size popular matchings

![Graph showing max-size popular matchings with nodes labeled $a_0, a_1, a_2$ and $b_0, b_1, b_2$ with edges and weights labeled.

There are two max-size popular matchings here: purple and green.

- Only the green matching occurs as a stable matching in the red/blue graph $G'$. 
Similar to popular maximum matchings, we can define popular optimal matchings.

Popular optimal matchings

- Suppose there is a utility function $f : E \to \mathbb{Q}$.
- It is only max-utility matchings that are relevant.

Does there exist a max-utility matching that is popular among max-utility matchings?

- If so, is it easy to find one?
Optimal solutions and popularity

Similar to popular maximum matchings, we can define popular optimal matchings.

Popular optimal matchings

- Suppose there is a utility function $f : E \rightarrow \mathbb{Q}$.
- It is only max-utility matchings that are relevant.

Does there exist a max-utility matching that is popular among max-utility matchings?
- If so, is it easy to find one?
  - The answer to both questions is “yes”.
Characterizing max-utility matchings

LP for max-utility matching in $G = (A \cup B, E)$

$$\max \sum_{e} f(e) \cdot x_e$$

$$\sum_{e \in \delta(u)} x_e \leq 1 \ \forall u \in A \cup B$$

$$x_e \geq 0 \ \forall e \in E.$$  

The polytope of max-utility matchings is a face of the matching polytope.

Thus $M$ is a max-utility matching $\iff M \subseteq E_0$ for some $E_0 \subseteq E$ and

- $M$ matches all vertices in $C$ for some $C \subseteq A \cup B$.

We want a $C$-perfect matching $M$ in $G_0 = (A \cup B, E_0)$ such that:

- $M \succ N$ or $M \sim N$ for all $C$-perfect matchings $N$ in $G_0$. 

T. Kavitha
Introduction to Popular Matchings
Popular C-perfect matchings

This problem can be reduced to the stable matching problem in a colorful graph $G_0^\dagger$.

- The colors of any edge $ab$ in $G_0^\dagger$ depend on whether $a \in C$ and $b \in C$.
  - For any $ab$ in $E_0$, there is always one green copy $ab$.
  - Every $ab$ in $E_0$ where $b \in C$ has $|C \cap B|$ more copies: $ab, ab, \ldots$.
  - Every $ab$ in $E_0$ where $a \in C$ has $|C \cap A|$ more copies: $ab, \ldots, ab$.

For any vertex in $A$:

- \( \text{red} \succ \text{blue} \succ \cdots \succ \text{green} \succ \text{magenta} \succ \cdots \succ \text{cyan} \).

For any vertex in $B$:

- \( \text{cyan} \succ \cdots \succ \text{magenta} \succ \text{green} \succ \cdots \succ \text{blue} \succ \text{red} \).

Within any color class, every vertex maintains its original preference order $\succ$.

The Gale-Shapley algorithm in $G_0^\dagger$ solves the popular C-perfect matching problem.
References

1. T. Kavitha.  

2. T. Kavitha.  
   *Maximum matchings and popularity.* In ICALP 2021 (to appear in SIDMA).

3. T. Kavitha.  
   *Popular Solutions for Optimal Matchings.* To appear in WG 2024.

Thank you! Any questions?