Rank Maximal Matchings

Meghana Nasre

talk based on paper by
R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, K. Paluch
Outline of the Talk

• Matchings – preliminaries.
• Rank maximal matchings.
  • Problem definition.
  • Some intuition.
• Dulmage Mendelsohn decomposition.
• An efficient algo for RMM.
Matching in a graph

A matching $M$ is a set of vertex disjoint edges.
Matching in a graph

A matching $\mathcal{M}$ is a set of vertex disjoint edges.

- **Goal:** compute a maximum sized matching.
A matching $M$ is a set of vertex disjoint edges.

- **Maximal**: $M$ is not a strict subset of any other matching. There may exist a larger sized matching.
Maximal vs. Maximum matchings

A matching $\mathcal{M}$ is a set of vertex disjoint edges.

- **Maximal:** $\mathcal{M}$ is not a strict subset of any other matching. There may exist a larger sized matching.

- **Maximum:** Has size as large as possible amongst all matchings. Every maximum matching is maximal.
Alternating paths

A path having alternate matched and unmatched edges.
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 (1957): A matching $M$ is maximum iff $M$ does not admit an augmenting path with respect to it.

- Assume $P$ is an augmenting path.
Using augmenting paths

Berge’s Theorem (1957): A matching $M$ is maximum iff $M$ does not admit an augmenting path with respect to it.

- Assume $P$ is an augmenting path.
- $M' = M \oplus P$.

![Diagram showing augmenting path](image-url)
Berge’s Theorem

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

Proof (by contradiction)

- Suppose $\mathcal{M}$ does not admit any aug. path and still it is not maximum.
- Some other matching $\mathcal{M}'$ 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 $M \oplus M'$.
- Construct an aug. path w.r.t. $M$. 
Input: \( G = (V, E) \).

Goal: Compute a max. cardinality matching.
Computing max. matchings

Input: $G = (V, E)$.
Goal: Compute a max. cardinality matching.

$G$ is bipartite.
- Hopcroft and Karp (1973): $O(m\sqrt{n})$ time algorithm.
Computing max. matchings

**Input:** $G = (V, E)$.

**Goal:** Compute a max. cardinality matching.

$G$ is bipartite.

- Hopcroft and Karp (1973): $O(m\sqrt{n})$ time algorithm.

General graphs.

- Edmond’s Blossom shrinking algorithm (1963): $O(n^4)$ time.
- Micali and Vazirani (1980): $O(m\sqrt{n})$ time algorithm.
Outline of the Talk

- Matchings – preliminaries.
- Rank maximal matchings.
  - Problem definition.
  - Some intuition.
- Dulmage Mendelsohn Decomposition.
- An efficient algorithm for RMM.
Rank Maximal Matchings

Input: Bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$.

- Every $a \in \mathcal{A}$ has a preference ordering over the edges.

Goal: Compute a matching that matches as many applicants to their rank-1 posts, subject to that as many to rank-2 and so on.
Rank Maximal Matchings

Input: Bipartite graph \( G = (\mathcal{A} \cup \mathcal{P}, E) \).

- Every \( a \in \mathcal{A} \) has a preference ordering over the edges.

Goal: Compute a matching that matches as many applicants to their rank-1 posts, subject to that as many to rank-2 and so on.

Signature of matching: \( \langle x_1, x_2, \ldots, x_{r+1} \rangle \).

\( x_j \): the number of applicants matched to their rank-\( i \) posts.
Comparing two matchings

Input: Bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$.

- Every $a \in \mathcal{A}$ has a preference ordering over the edges.

| $M$: $\langle x_1, x_2, \ldots, x_{r+1} \rangle$ | $M'$: $\langle y_1, y_2, \ldots, y_{r+1} \rangle$ |

- $x_i$: # of applicants matched to their rank-i post in $M$. 
Comparing two matchings

Input: Bipartite graph $G = (A \cup P, E)$.

- Every $a \in A$ has a preference ordering over the edges.

\[
M : \langle x_1, x_2, \ldots, x_{r+1} \rangle \quad \quad M' : \langle y_1, y_2, \ldots, y_{r+1} \rangle
\]

- $x_i$: # of applicants matched to their rank-i post in $M$.
- $M$ is better than $M'$ w.r.t. rank-maximality ($M \succ M'$) iff:
  - there exists an index $1 \leq k \leq r$:
  - for $1 \leq i < k$, $x_i = y_i$ and $x_k > y_k$. 
Comparing two matchings

Input: Bipartite graph \( G = (\mathcal{A} \cup \mathcal{P}, E) \).

- Every \( a \in \mathcal{A} \) has a preference ordering over the edges.

\[
\begin{align*}
M & : \langle x_1, x_2, \ldots, x_{r+1} \rangle \\
M' & : \langle y_1, y_2, \ldots, y_{r+1} \rangle
\end{align*}
\]

- \( x_i \): \# of applicants matched to their rank-\( i \) post in \( M \).
- \( M \) is better than \( M' \) w.r.t. rank-maximality (\( M \succ M' \)) iff:
  - there exists an index \( 1 \leq k \leq r \):
  - for \( 1 \leq i < k \), \( x_i = y_i \) and \( x_k > y_k \).

- **Goal:** Compute a matching that has best signature.
Comparing two matchings

Input: Bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$.
  - Every $a \in \mathcal{A}$ has a preference ordering over the edges.

$$
\begin{align*}
M: & \langle x_1, x_2, \ldots, x_{r+1} \rangle \\
M': & \langle y_1, y_2, \ldots, y_{r+1} \rangle
\end{align*}
$$

- $x_i$: # of applicants matched to their rank-i post in $M$.
- $M$ is better than $M'$ w.r.t. rank-maximality ($M \succ M'$) iff:
  - there exists an index $1 \leq k \leq r$:
    - for $1 \leq i < k$, $x_i = y_i$ and $x_k > y_k$.
- **Goal**: Compute a matching that has best signature.
- Can be achieved by converting the problem to a max-weight matching problem with edge of rank-i given a weight of $n^{r-i}$. 
Greedy approach

- Greedily match applicants to their rank-1, subject to this, greedily match applicants to their rank-2 and so on.
Greedy approach

- Greedily match applicants to their rank-1, subject to this, greedily match applicants to their rank-2 and so on.

- Choice of rank-1 edges forces a signature of $\langle 2, 0, 1 \rangle$. 
Greedy approach

- Greedily match applicants to their rank-1, subject to this, greedily match applicants to their rank-2 and so on.

- Optimal signature of \( \langle 2, 1, 0 \rangle \).
Graph on rank-1 edges

\[ G_1 \]
Graph on rank-1 edges

Some observations

- For a RMM, the signature is $\langle 2, \times, \times \rangle$
Some observations

- For a RMM, the signature is $\langle 2, \times, \times \rangle$
- $p_2$ and $p_4$ must be matched to some rank-1 applicant.
Some observations

• For a RMM, the signature is $\langle 2, \times, \times \rangle$

• $p_2$ and $p_4$ must be matched to some rank-1 applicant.

• Safe to delete higher ranked edges incident on $p_2$ and $p_4$. 

Graph on rank-1 edges

$G_1$

$A$

$P$

$a_1 - a_2 - a_3$

$p_1 - p_2 - p_3 - p_4$
Some observations

- For a RMM, the signature is $\langle 2, \times, \times \rangle$
- $p_2$ and $p_4$ must be matched to some rank-1 applicant.
- Safe to delete higher ranked edges incident on $p_2$ and $p_4$.
- Unclear which of $a_1, a_2$ gets matched to rank-1 post in an RMM.
• Matchings – preliminaries.
• Rank maximal matchings.
  • Problem definition.
  • Some intuition.
• Dulmage Mendelsohn decomposition.
• An efficient algorithm for RMM.
Need to study DM decomposition

• Which vertices remain matched in every max. matching?
• Which edges can NEVER belong to a max. matching?
• How to use these answers to solve RMM efficiently.
Maximum matchings – some observations

- Size of max. matching: 3.
- Every max. matching matches \{a_3, a_4, p_1, p_4\}.
- No max. matching contains \( (a_3, p_1) \).

Vertices that some max. matching leaves unmatched:

\{a_1, a_2, p_2, p_3\}
Sets Odd, Even, Unreachable

- Bipartite graph $G$.
- Max. matching $M$.
- Define 3 sets of vertices:
  - even ($\mathcal{E}$)
  - odd ($\mathcal{O}$)
  - unreachable ($\mathcal{U}$)
Sets Odd, Even, Unreachable

- Bipartite graph $G$.
- Max. matching $M$.
- Define 3 sets of vertices:
  - even ($\mathcal{E}$)
  - odd ($\mathcal{O}$)
  - unreachable ($\mathcal{U}$)

- Even ($\mathcal{E}$): set of vertices which are reachable via even length alternating paths from a free vertex $x$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{sets_odd_even_unreachable.png}
\end{figure}
Sets Odd, Even, Unreachable

- Bipartite graph $G$.
- Max. matching $M$.
- Define 3 sets of vertices:
  - even ($E$)
  - odd ($O$)
  - unreachable ($U$)

- Even ($E$): set of vertices which are reachable via even length alternating paths from a free vertex $x$.
- Odd ($O$): set of vertices which are reachable via odd length alternating paths from a free vertex $x$. 
Sets Odd, Even, Unreachable

- Bipartite graph $G$.
- Max. matching $M$.
- Define 3 sets of vertices:
  - even ($\mathcal{E}$)
  - odd ($\mathcal{O}$)
  - unreachable ($\mathcal{U}$)

- Even ($\mathcal{E}$): set of vertices which are reachable via even length alternating paths from a free vertex $x$.
- Odd ($\mathcal{O}$): set of vertices which are reachable via odd length alternating paths from a free vertex $x$.
- Unreachable ($\mathcal{U}$): vertices which are neither odd nor even.
Sets Odd, Even, Unreachable

- Bipartite graph $G$.
- Max. matching $M$.
- Define 3 sets of vertices:
  - even ($\mathcal{E}$)
  - odd ($\mathcal{O}$)
  - unreachable ($\mathcal{U}$)

- Even ($\mathcal{E}$): set of vertices which are reachable via even length alternating paths from a free vertex $x$.
- Odd ($\mathcal{O}$): set of vertices which are reachable via odd length alternating paths from a free vertex $x$.
- Unreachable ($\mathcal{U}$): vertices which are neither odd nor even.
Property 1

- Define 3 sets of vertices:
  - even (\(E\))
  - odd (\(O\))
  - unreachable (\(U\))

- \(E, O, U\) form a partition of \(A \cup P\).
Property 2

Define 3 sets of vertices:
- even \((E)\)
- odd \((O)\)
- unreachable \((U)\)

\(\mathcal{E}, \mathcal{O}, \mathcal{U}\) sets are invariant of the max. matching.
Property 3

- Define 3 sets of vertices:
  - even \((\mathcal{E})\)
  - odd \((\mathcal{O})\)
  - unreachable \((\mathcal{U})\)

- No max. matching contains an \(\mathcal{O}\mathcal{O}\), \(\mathcal{O}\mathcal{U}\) edge.
Dulmage Mendelsohn decomposition (1958)

- Bipartite graph $G$. Max. matching $M$.
- Define 3 sets of vertices:
  - even ($\mathcal{E}$)
  - odd ($\mathcal{O}$)
  - unreachable ($\mathcal{U}$)
- The sets $\mathcal{E}, \mathcal{O}, \mathcal{U}$ are partitions of the vertex set.
- The sets are invariant of the maximum matching.
- Every max. matching has cardinality $|\mathcal{O}| + |\mathcal{U}|/2$.
- No max. matching has edges of the form $(\mathcal{O}, \mathcal{O})$, $(\mathcal{O}, \mathcal{U})$.
- $G$ does not have edges of the form $(\mathcal{E}, \mathcal{E})$, $(\mathcal{E}, \mathcal{U})$. 
Outline of the Talk

• Matchings – preliminaries.
• Rank maximal matchings.
  • Problem definition.
  • Some intuition.
• Dulmage Mendelsohn decomposition.
• An efficient algorithm for RMM.
An efficient combinatorial algorithm

Main idea:

• Reduce the problem of computing RMM into a problem of computing max. cardinality matching in suitably defined graphs.
An efficient combinatorial algorithm

Main idea:

- Reduce the problem of computing RMM into a problem of computing max. cardinality matching in suitably defined graphs.

- $G_i = (A \cup P, E_1 \cup E_2 \cup \ldots \cup E_i)$.
- **Goal:** Compute $M_i$ in which is RMM in $G_i$. 
An efficient combinatorial algorithm

Main idea:

- Reduce the problem of computing RMM into a problem of computing max. cardinality matching in suitably defined graphs.

\[ G_i = (A \cup P, E_1 \cup E_2 \cup \ldots \cup E_i). \]

- **Goal:** Compute \( M_i \) in which is RMM in \( G_i \).
- Easy for \( G_1 \), simply a max. matching on rank-1 edges.
- How about \( G_2 \)?
An efficient combinatorial algorithm

Start with $G'_1 = (A \cup P, E_1), M_1$.
for $i = 1, \ldots, r$ do

- Partition vertices w.r.t. $M_i$, call them $O_i, E_i, U_i$.
- Delete all higher than $i$ ranked edges incident on $O_i, U_i$.
- Delete all edges that are labeled $(O_i O_i), (O_i U_i)$.
- Add edges in $E_{i+1}$ and call the resulting graph $G'_{i+1}$.
- Compute a max. matching $M_{i+1}$ in $G'_{i+1}$ by augmenting $M_i$. 
An efficient combinatorial algorithm

Start with $G'_1 = (A \cup P, E_1), M_1$.

for $i = 1, \ldots, r$ do

- Partition vertices w.r.t. $M_i$, call them $O_i, E_i, U_i$.
- Delete all higher than $i$ ranked edges incident on $O_i, U_i$.
- Delete all edges that are labeled $(O_i O_i), (O_i U_i)$.
- Add edges in $E_{i+1}$ and call the resulting graph $G'_{i+1}$.
- Compute a max. matching $M_{i+1}$ in $G'_{i+1}$ by augmenting $M_i$.

Note: $G'_i \subseteq G_i$.

Invariants:

- No edge that we delete is present in any RMM.
- Every RMM is a maximum matching in $G'_i$. 
Summary

- RMM are very different from maximum matchings.
- A classical decomposition result gives the best known combinatorial algorithm.
- DM decomposition used subsequently to find:
  - Capacitated RMM.
  - Another notion of optimality – popular matchings.
  - Popular matchings in a two sided setting (stable marriage problem).
- DM is a special case of the Gallai Edmonds Decomposition (1965–1968) for general graphs.
Thank You.