Classified Rank-Maximal Matchings and Popular Matchings – Algorithms and Hardness

CS6130 - Advanced Graph Algorithms

Harish Soham

23 April 2024

Overview

Rank Maximal Matching

Let A be a set of applicants and P be a set of posts, and each member of A has a preference list (possibly involving ties)

Rank Maximal Matching

A matching is *rank-maximal* if it matches the maximum number of applicants to their rank-1 posts, subject to that, the maximum number of applicants to their rank-2 posts, and so on.

Definition

The signature σ_M of a matching M is an r-tuple $(x_1, ..., x_r)$ where r denotes the largest rank used by an applicant to rank any post. For $1 \leq k \leq r$, x_k denotes the number of rank k edges in M.

Rank Maximal Matching

- Consider the following preference list:

$$\begin{aligned} a_1 &: \quad p_1, p_2, p_3 \\ a_2 &: \quad (p_2, p_3), p_1 \\ a_3 &: \quad p_2, p_1, p_3 \end{aligned}$$

Let $M_1 : a_1 \rightarrow p_1, a_2 \rightarrow p_3, a_3 \rightarrow p_2$
$\implies \sigma_{M_1} = (3, 0, 0)$
Let $M_2 : a_1 \rightarrow p_2, a_2 \rightarrow p_3, a_3 \rightarrow p_1$
$\implies \sigma_{M_1} = (1, 2, 0)$
Hence, $M_1 \succ_{RM} M_2$

Attempts

- We note that the problem can be converted to a maximum weight bipartite matching problem by updating weights appropriately.
- This can be achieved by assigning a weight of n^{r-i} (where n is the number of vertices and r is the maximum rank given to any post) to each applicant's i^{th} choice, and 0 to a post that does not appear in the applicant's preference list.
- However, the use of such large integers as edge weights in the graph can lead to implementation complications.

Attempts

- Consider the following example

$$\begin{aligned} a_1 &: \quad p_1, p_2, p_3 \\ a_2 &: \quad p_2, p_3, p_1 \\ a_3 &: \quad p_2, p_1, p_3 \end{aligned}$$

- Does a simple greedy algorithm work here? - i.e. we first match maximum number of rank 1 edges and then match maximum number of rank 2 edges and so on.
  - Rank 1 edges - $a_1 \rightarrow p_1$ and $a_2 \rightarrow p_2$ are matched
  - Rank 2 edges - $a_3 \rightarrow p_3$ is matched
  - Rank 3 edges - None of them can be matched

  $\sigma_{sg} = (2, 0, 1)$
- We note that $M : a_1 \rightarrow p_1$, $a_2 \rightarrow p_3$, $a_3 \rightarrow p_2$
  $\sigma_M = (2, 1, 0)$ has a better signature

Attempts

- We want to modify the algorithm to somehow change previously matched edges while keeping the number of matches of previous ranks the same.
- We can do this with flows and the modifications are done using augmenting paths.
- The final algorithm is very similar to the algorithm shown in class for popular matchings with ties
  i.e. augment the matching after deleting $O - O$ and $O - U$ edges in each step from rank$=1$ to rank$=r$.
  (*Rank-Maximal Matchings* - Irving et. al)

Comparison with Popular matchings

- Are the two notions of optimality equivalent?
- Are rank maximal matchings popular?
- Are popular matchings (provided they exist) rank maximal?

Comparison with Popular matchings

- Are the two notions of optimality equivalent? **False**
  Popular matchings may not exist while rank maximal matchings always do.
- Are rank maximal matchings popular? **False**
  For the same reason as above
- Are popular matchings (provided they exist) rank maximal?

Comparison with Popular matchings

Are popular matchings (provided they exist) rank maximal? **False**

Counter-example:
Consider the following

$$\begin{aligned} a_1 &: \quad p_1, p_2 \\ a_2 &: \quad p_1 \end{aligned}$$

$M = \{a_1 \to p_1, a_2 \to\}$ is a popular matching but clearly is not rank-maximal

Problem Statement - Motivation

- Consider the case of resident doctors being assigned to hospitals.
- Hospitals can accommodate multiple resident doctors.
- Assume the residents have a preference lists on the hospitals they want to be assigned to.
- The hospitals on the other hand have restrictions on the number of doctors it can accommodate.
- Note that we are looking at a many-one matching instead of the usual one-one scenario.

Problem Statement - Motivation

Preference list is as follows

$$\begin{aligned} D_1 &: \quad H_1, H_2, H_3 \\ D_2 &: \quad (H_2, H_3), H_1 \\ D_3 &: \quad H_2, H_1, H_3 \\ D_4 &: \quad H_1, H_2 \end{aligned}$$

Quotas of hospitals are as follows

$$\begin{aligned} Q(H_1) &= 1 \\ Q(H_2) &= 2 \\ Q(H_3) &= 2 \end{aligned}$$

Problem Statement - Motivation

- As a natural extension, hospitals can also have limits on how many residents it takes in that belong to a particular medical specialization.
- It could also have restrictions based on gender, demography etc.

Problem Statement - Classifications

- Suppose hospital H_2 can take a maximum of two doctors specializing in Radiology and only one specializing in neurology.
- We can add quotas as follows:
    - $q(H_2 \leftarrow \{D_1, D_3\}) = 2$
    - $q(H_2 \leftarrow \{D_2, D_4\}) = 1$
- We formalize the notation below

Definition

A *classification* is a set of subsets of $N(u)$. Each subset $C_u^i \in \mathcal{C}_u$ is called a *class*, and each class has its own quota $0 < q(C_u^i) \leq q(u)$.

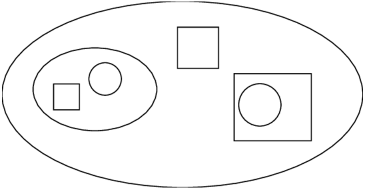
Problem Statement - Classifications

Pref. list		Classifications	Quotas
D_1 :	H_1, H_2, H_3	$C_{H_2} = \{C^1_{H_2}, C^2_{H_2}\}$	$q(C^1_{H_2}) = 2$
D_2 :	$(H_2, H_3), H_1$	$C^1_{H_2} = \{D_1, D_3\}$	$q(C^2_{H_2}) = 1$
D_3 :	H_2, H_1, H_3	$C^2_{H_2} = \{D_2, D_4\}$	$q(C^1_{H_3}) = 1$
D_4 :	H_1, H_2	$C_{H_3} = \{C^1_{H_3}\}$	$q(H_1) = 1$
		$C^1_{H_3} = \{D_1, D_2\}$	$q(H_2) = 2$
			$q(H_3) = 2$

Laminar Families

Definition

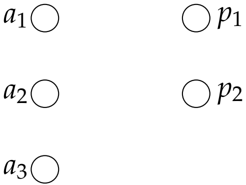
A family F of subsets of a set S is said to be laminar if, for every pair of sets $X, Y \in F$, either $X \subseteq Y$ or $Y \subseteq X$ or $X \cap Y = \phi$.



For the scope of the problem statement, we assume that the classification of each vertex forms a laminar family.

Problem Statement - Summary

- We have applicants A and posts P. Each applicant have preference lists on P (with possible ties).
- Each post P has a quota referring to the maximum number of applicants that can be matched to that post.
- Each post P also has a laminar classification with individual class quotas.



$$a_i : p_{k1}, p_{k2} \ldots \qquad C_{p_i} = \{C^1_{p_i}, C^2_{p_i}, \ldots\} \qquad q(p_i) = q_{k_i}$$

$$q(C^j_{p_i}) = q^j_{k_i}$$

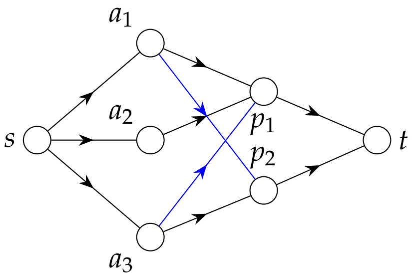
. . .

. . .

Pref. List — Classifications — Quotas

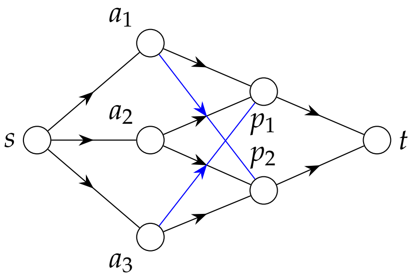
Constructing the Flow Network

Consider the simple scenario where the preference lists are $a_1 : p_1, p_2 \quad a_2 : p_1 \quad a_3 : p_2, p_1$. The flow network is then:



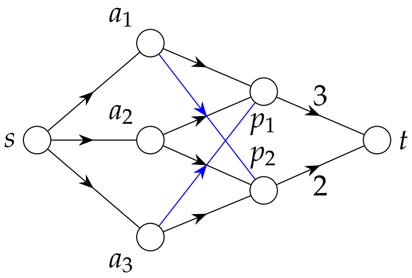
Ties

Let us allow ties, so that the preference lists are now
$a_1 : p_1, p_2 \quad a_2 : (p_1, p_2) \quad a_3 : p_2, p_1$. The flow network is then simply:



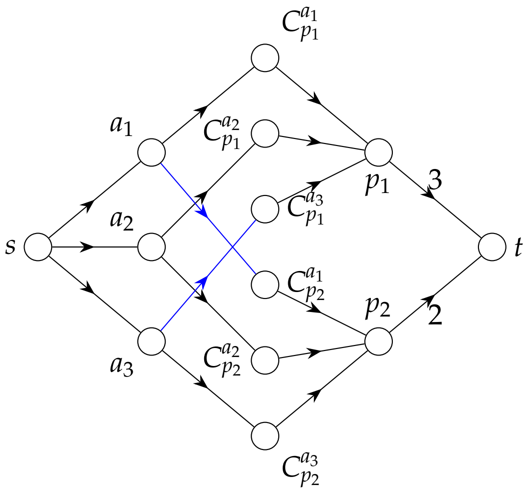
Quotas

Let us allow posts to accept multiple applicants, e.g. $q(p_1) = 3$, $q(p_2) = 2$. The flow network is then simply:



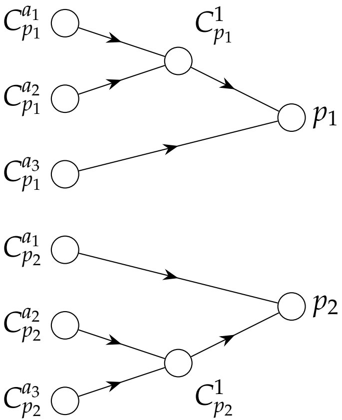
Dummies

Now add dummy $C_{p_j}^{a_i}$ nodes, such that any (a_i, p_j) edge is replaced by two edges: $(a_i, C_{p_j}^{a_i})$ and $(C_{p_j}^{a_i}, p_j)$.



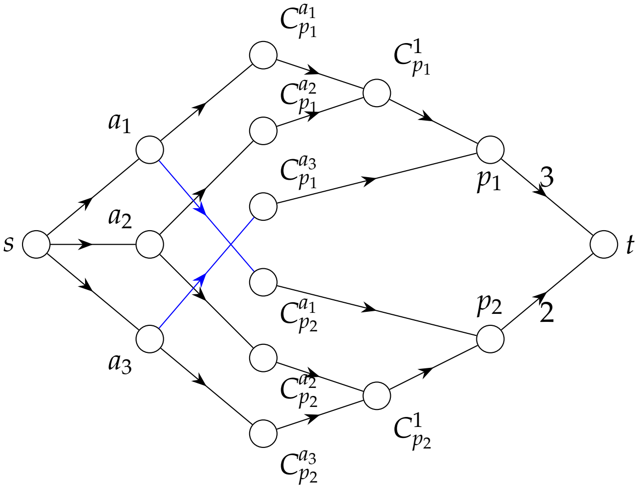
Laminar Classifications

We can represent the Laminar Classifications $\mathcal{C}_{p_1} = \left\{ C^1_{p_1} = \{a_1, a_2\} \right\}$, $\mathcal{C}_{p_2} = \left\{ C^1_{p_2} = \{a_2, a_3\} \right\}$ with $q(\mathcal{C}_{p_1}) = q(\mathcal{C}_{p_2}) = 1$ as the following subtrees:



Flow Network

Thus, we finally have the following flow network:



Algorithm

Algorithm 1 Laminar CRMM

Construct the flow network $H_0 = (V, F_0)$ as described in Section 2.
Let $F'_0 = F_0$ and for each i set $E'_i = E_i$.
for $k = 1$ to r **do**
    $H_k = (V, F_k)$ where $F_k = F'_{k-1} \cup \{(C_a^p, C_p^a) \mid (a,p) \in E'_k\}$.
    Let f_k be a max-flow in H_k. Compute the residual graph $H_k(f_k)$ w.r.t. flow f_k.
    Compute the sets S_k, T_k and U_k.
    Delete all edges of the form $(T_k \cup U_k, S_k)$ in $H_k(f_k)$.
    Delete an edge $(a,p) \in E'_j$ where $j > k$ if $C_a^p \in T_k \cup U_k$ or $C_p^a \in S_k \cup U_k$.
    Let $H'_k = (V, F'_k)$ be the modified $H_k(f_k)$ and let $G'_k = (A \cup P, \bigcup_{i=1}^{k} E'_i)$.
    Let $M_k = \{(a,p) | (C_p^a, C_a^p) \in H'_k\}$.
end for
Return M_r.

Example

Consider the following instance:

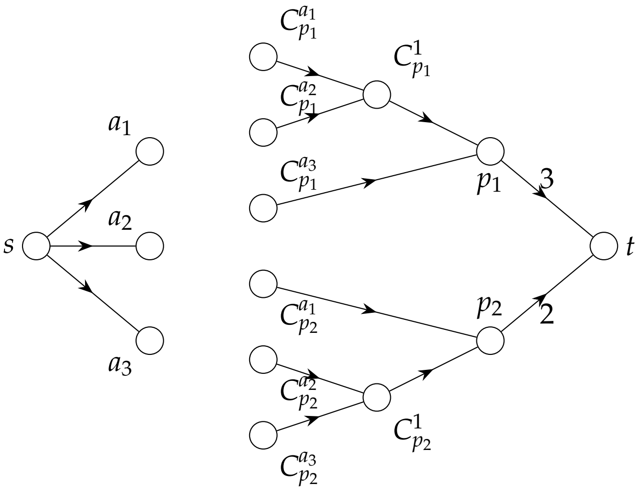
$$a_1 : p_1, p_2 \qquad \mathcal{C}_{p_1} = \left\{ C^1_{p_1} = \{a_1, a_2\} \right\} \qquad q\left(C^1_{p_1}\right) = q\left(C^1_{p_2}\right) = 1$$

$$a_2 : (p_1, p_2) \qquad \mathcal{C}_{p_2} = \left\{ C^1_{p_2} = \{a_2, a_3\} \right\} \qquad q(p_1) = 3$$

$$a_3 : p_2, p_1 \qquad q(p_2) = 2$$

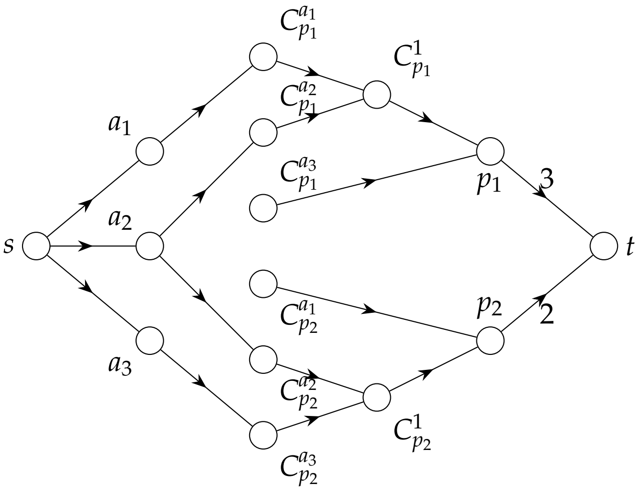
Preference Lists   Classifications   Quotas

H_0

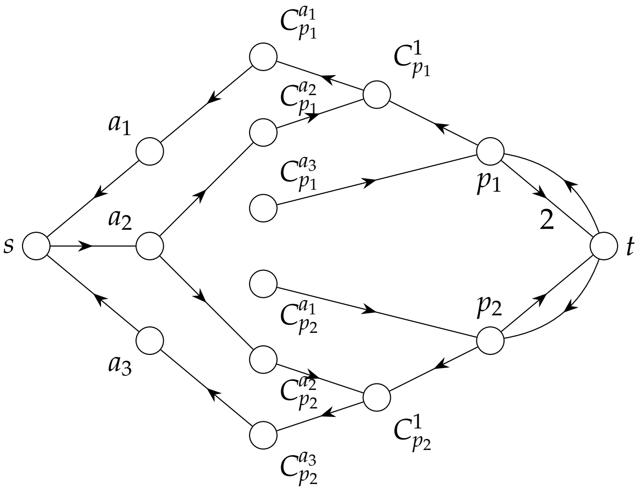


$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
3
a_2
s
t
p_2
2
$C_{p_2}^{a_1}$
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

H_1

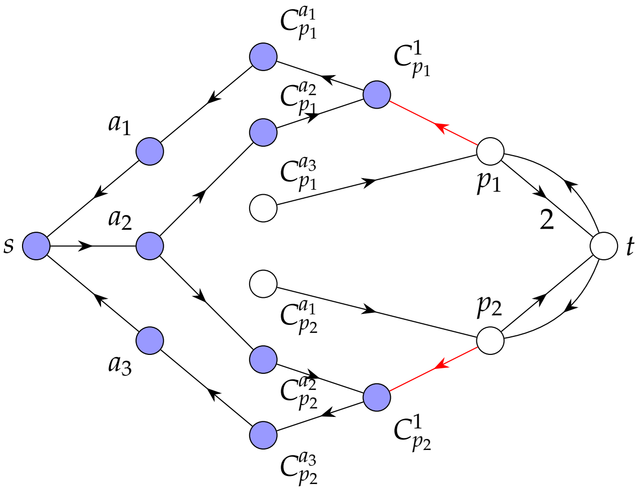

$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
3
a_2
s
t
$C_{p_2}^{a_1}$
p_2
2
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

$H_1(f_1)$



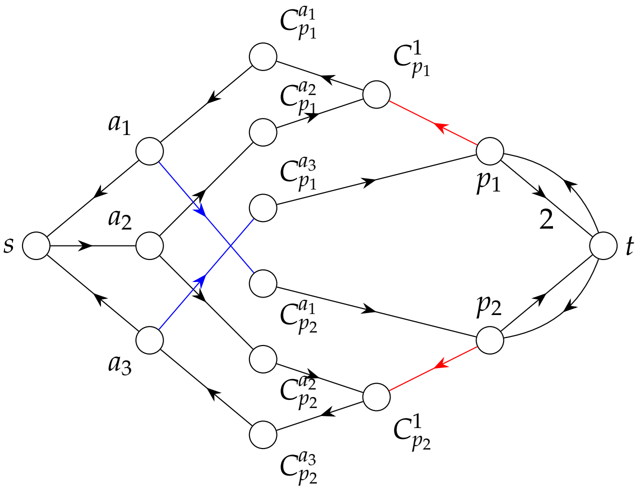
$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
2
a_2
s
t
p_2
$C_{p_2}^{a_1}$
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

$H_1(f_1)$



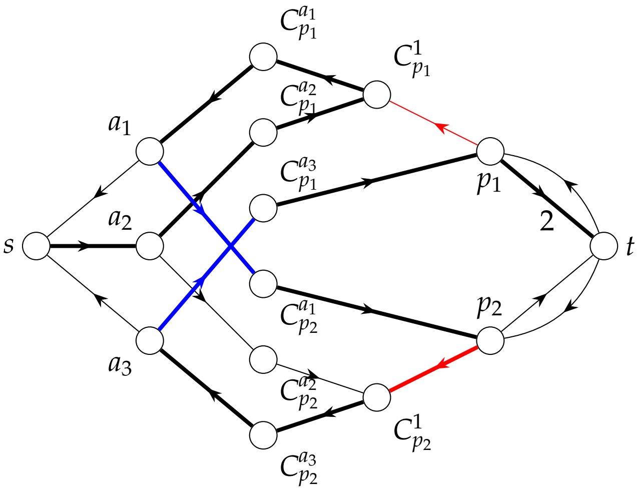
$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
2
s
a_2
t
$C_{p_2}^{a_1}$
p_2
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

Why Step 7?

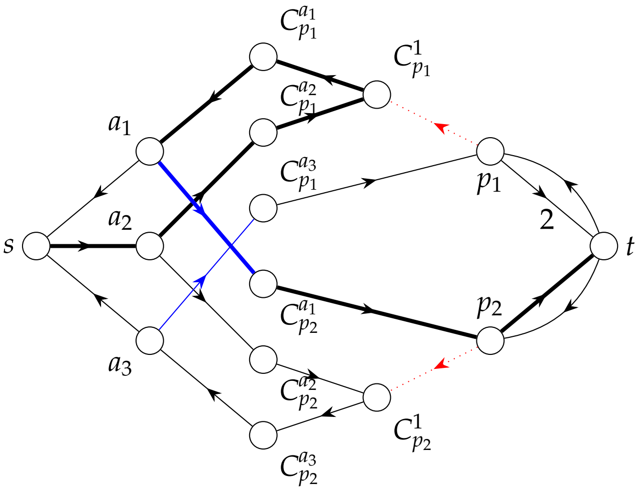


$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
a_2
2
s
t
$C_{p_2}^{a_1}$
p_2
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

Degradation of Signature



H_2


$C_{p_1}^{a_1}$
$C_{p_1}^{1}$
$C_{p_1}^{a_2}$
a_1
$C_{p_1}^{a_3}$
p_1
2
s
a_2
t
$C_{p_2}^{a_1}$
p_2
a_3
$C_{p_2}^{a_2}$
$C_{p_2}^{1}$
$C_{p_2}^{a_3}$

Running Time

The size of our flow network is equal to the total size of all preference lists, i.e., $O(|E|)$.

The maximum matching size in our instance is $O(|E|)$, so the max-flow in our network is also $O(|E|)$.

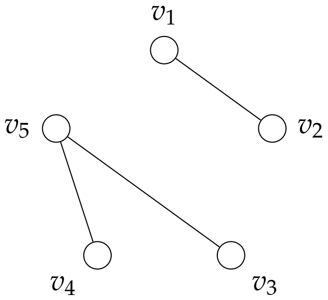
This gives an upper bound of $O(|E|^2)$ on the running time.

NP-hardness

- The Independent Set Decision Problem is NP-complete.
- It is defined as the problem of deciding whether a given graph contains an independent set of given size k.
- The decision version of Classified Rank-Maximal Matchings decides whether a suitable given graph contains a classified rank-maximal matching of given size k.
- Prove that the decision version of the Independent Set Problem reduces to that of Classified Rank-Maximal Matchings.

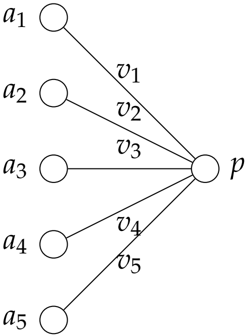
Independent Set Decision Problem

- Input: Graph $G = (V, E)$, positive integer k.
- Output: Boolean value that is true if the graph contains an independent set of size k, and false otherwise.



Reduction

Consider the following instance of the Classified Rank-Maximal Matchings Problem:



We have a single post, and for every vertex in the Independent Set Problem we have an applicant with a singleton preference list.

Reduction

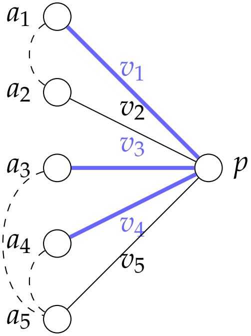
For each edge $\{u, v\}$ in the independent set decision problem instance, add a classification $C_p^i = \{u, v\}$ to $\mathcal{C}$ with $q(C_p^i) = 1$ in the Classified Rank-Maximal Matchings instance.

$$\therefore \mathcal{C} = \{C_p^1 = \{v_1, v_2\}, C_p^2 = \{v_3, v_5\}, C_p^3 = \{v_4, v_5\}\}$$

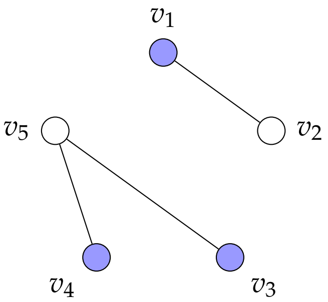
This ensures that no two adjacent vertices can be chosen.

Reduction

A Classified Rank-Maximal Matching of size k thus corresponds to an Independent Set of size k:



(a) Classified Rank-Maximal Matching



(b) Independent Set

Reduction

We have reduced the Independent Set Decision Problem to an instance of the Classified Rank-Maximal Matchings Decision Problem.

Thus, the Classified Rank-Maximal Matchings Decision Problem must be at least as hard as the Independent Set Decision Problem.

$\therefore$ The Classified Rank-Maximal Matchings Decision Problem is NP-hard.

Thank You!