CS6130: Paper Presentation Report

Harish Manoharan[1] and Soham Roy[2]

IIT Madras
bs19b012@smail.iitm.ac.in[1], ee20b130@smail.iitm.ac.in[2]

1 Introduction

The major problem studied in the paper is that of rank maximal matchings. Previous work includes studies on the rank maximal matching problem in the many to many and one to one setting [Rank Maximal Matchings - Irving et al.]. The paper studies a generalization of the many to many setting with the addition of classifications. The addition of classifications broadens the scope of rank maximal matchings to a variety of real world applications such as student-course matching, resident-hospital matching etc. The paper introduces a $O(|E|^2)$ algorithm for the classified rank maximal matching problem when the classification for each vertex forms laminar family and shows that the problem is NP-hard when the classes are non laminar.

2 Problem Statement

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.
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.
A *classification* is a set of subsets of $N(u)$. Each subset $C_u^i \in \mathrm{C}_u$ is called a *class*, and each class has its own quota $0 < q(C_u^i) \leq q(u)$.
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). Posts can accept multiple applicants (many-one setting). Given classifications and their corresponding quotas for each post, Compute a rank maximal matching satisfying the quotas.
Example: Suppose doctor residents are being matched to hospitals. The residents give their preference for their hospitals (preference list). The hospitals could have quotas on residents of a particular specialization or on their age or gender etc. (classifications and quotas). Note that multiple residents could be matched to the same hospital (many-one setting)

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

3 Intuition / Examples / Your Obsevations

3.1 Intuition

We start by looking at the simple one-one case and extend further.
Consider the following example

$a_1 : p_1, p_2, p_3$
$a_2 : p_2, p_3, p_1$
$a_3 : p_2, p_1, p_3$

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 - $a_1 \to p_1$ and $a_2 \to p_2$ are matched
- Rank 2 edges - $a_3 \to p_3$ is matched
- Rank 3 edges - None of them can be matched

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

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)

3.2 Alternate solutions

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. Why n^{r-i}? We want to assign a weight in such a way that even one matching of rank r outweighs any number of matchings of rank r-1 or lower. However, the use of such large integers as edge weights in the graph can lead to implementation complications.

3.3 Comparison with Popular Matchings

Popular matching - A matching M is popular if there is no matching M' such that more applicants prefer M' over M. From the definition of rank maximal matchings, it is simple to see that a rank maximal matching always exists. However, this is not the case for popular matchings.
In the same sense, we can say that rank maximal matchings need not be popular. So when popular matchings do exist, are they rank maximal? Consider the following:

$$a_1 : p_1, p_2$$
$$a_2 : p_1$$

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

We further note that popular matchings are maximal in the number of rank-1 edges similar to rank maximal matchings but the same does not follow for rank > 1

4 Results and Main Techniques

The paper provides an algorithm for the classified rank-maximal matching problem for laminar classifications. The algorithm has an upper bound of $O(|E|^2)$ on the running time. The main idea is to construct a specific flow network, and repeatedly compute the maxflow in the network and modify the network. Finally we have a rank-maximal matching, after edges of each rank are processed.

4.1 Algorithm

We first construct our initial graph with laminar classifications represented as a tree each on either side of the graph. Then, for each rank in the preference list, we add a set of edges to this graph. We compute the maxflow, the residual network, and the sets S, T, U. We delete certain edges to prevent degradation of the signature in subsequent steps. We repeat until all ranks are depleted. Thus, we obtain our matching by considering the L-R edges which are saturated.

5 Key takeaway from the paper

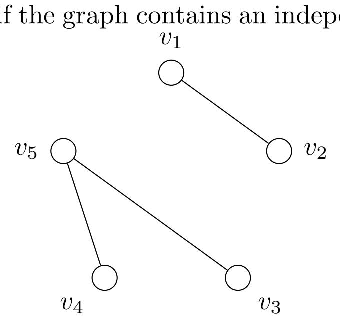
The paper has provided us an algorithm for the classified rank-maximal matching problem for laminar classifications. It also shows that the problem is NP-hard in the general setting. It does so by reducing the monotone 1-in-3 SAT to a decision version of the classified rank-maximal matching problem. However, a simpler proof is to reduce the independent set decision problem instead.

6 Frame a question

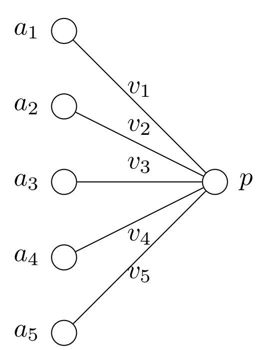
6.1 Question

Consider the Independent Set Decision Problem

- Independent Set: An independent set is a set of vertices such that no two vertices are adjacent.
- 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.



Consider the following instance of the Classified Rank-Maximal Matchings Problem (many-one setting):



We have a single post, and for every vertex in the Independent Set Problem we have an applicant with a singleton preference list i.e $|A| = |V|$ and $|P| = 1$.

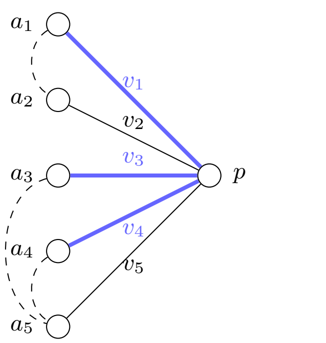
How do we represent the edges of the Independent Set Problem in our instance of the Classified Rank-Maximal Matchings Problem? i.e., how do we ensure that for each edge at most one endpoint is present in our solution set?

6.2 Answer

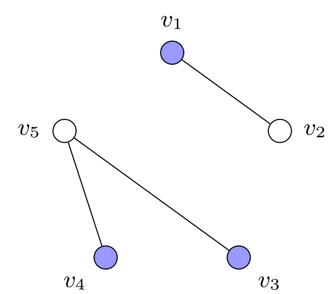
For each edge $\{u, v\}$ in the independent set decision problem instance, add a classification $C_p^i = \{u, v\}$ to $\mathcal{C}$ with $q(C_p^i) = 1$ in the Classified Rank-Maximal Matchings instance.
In the example shown above, we have the following classifications:

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

This ensures that no two adjacent vertices can be chosen. 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