Optimal Matchings with One-Sided Preferences: Fixed and Cost-Based Quotas

Santhini K. A.
Indian Institute of Technology Madras
Chennai, India
santhini@cse.iitm.ac.in

Govind S. Sankar
Duke University
Durham, USA
govind.subash.sankar@duke.edu

Meghana Nasre
Indian Institute of Technology Madras
Chennai, India
meghana@cse.iitm.ac.in

ABSTRACT

We consider the well-studied many-to-one bipartite matching problem of assigning applicants $\mathcal{A}$ to posts $\mathcal{P}$ where applicants rank posts in the order of preference. This setting models many important real-world allocation problems like assigning students to courses, applicants to jobs, amongst many others. In such scenarios, it is natural to ask for an allocation that satisfies guarantees of the form "match at least 80% of applicants to one of their top three choices" or "it is unacceptable to leave more than 10% of applicants unassigned". The well-studied notions of rank-maximality and fairness fail to capture such requirements due to their property of optimizing extreme ends of the *signature* of a matching. We, therefore, propose a novel optimality criterion, which we call as the "cumulative better signature".

We investigate the computational complexity of the new notion of optimality in the setting where posts have associated *fixed* quotas. We prove that under the fixed quota setting, the problem turns out to be NP-hard under natural restrictions. We provide randomized algorithms in the fixed quota setting when the number of ranks is constant. We also study the problem under a *cost-based quota* setting and show that min-cost cumulative better matching can be computed efficiently. Apart from circumventing the hardness, the cost-based quota setting is motivated by real-world applications like course allocation or school choice where the capacities or quotas need not be rigid.

KEYWORDS

Matchings with preferences; Network flows; Randomized algorithms; NP-hardness

ACM Reference Format:
Santhini K. A., Govind S. Sankar, and Meghana Nasre. 2022. Optimal Matchings with One-Sided Preferences: Fixed and Cost-Based Quotas. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

In this paper, we consider the many-to-one bipartite matching problem where a set of applicants $\mathcal{A}$ are assigned to a set of posts $\mathcal{P}$ and applicants rank posts in an order of preference possibly involving ties. We say that p and p' are tied in the preference list of applicant a if the ranks on the edges (a, p) and (a, p') are the same. In a standard setting of the problem, a post p has an input quota $q(p)$ denoting the maximum number of applicants that can be assigned to p. An applicant can be assigned to at most one post and prefers to be matched over remaining unmatched. A matching M is a subset of the edges such that every applicant has at most one edge incident on it, and every post has at most $q(p)$ many edges incident on it in M. The goal in such a setting is to compute a matching that is *optimal* with respect to the preferences specified by the applicants. This setting models any problem that can be formulated as an allocation problem where some items are to be allocated to agents that have preferences over them. This setting is also called as the House Allocation setting in literature [1] and several optimality criteria such as Pareto optimality [1], popularity [15], rank-maximality [9, 11, 19] and fairness [10] have been investigated for the one-sided preference list model. In this work, we are motivated by two practical considerations of the problem – (i) the measure of optimality and (ii) the assumption of fixed input quotas present as a part of the input instance.

The standard notions of rank-maximality [11] and fairness [10] (definitions given below) are not designed to capture requirements of the form "output a matching (if possible) which matches at least k out of n applicants to one of the first or second choice posts" or "the output matching must match at least ℓ out of n applicants to some post". See Figure 1 for an example instance where the input requirement could be "match at least 50% of applicants to their top choice, at least 75% of the applicants to one of their first and second choice". The matching M (see Figure 1) satisfies this requirement but, as will be seen, is neither rank-maximal nor fair. Furthermore, if the requirement were to "output a matching that matches at least 50% applicants to their rank-1 posts and all applicants to one of their rank-1 or rank-2 posts", then the instance (with unit quotas) does not admit any such matching. However, it is easy to see that if it possible to violate or increase the quota of posts, then $M' = \{(a_1, p_4), (a_2, p_1), (a_3, p_1), (a_4, p_2), (a_5, p_5), (a_6, p_3)\}$ satisfies this requirement. In our work, we address the above by firstly introducing a new notion of optimality and secondly by allowing costs to control quotas of the posts. We need some definitions before we formally define our problems.

Pareto-optimality [1] is the weakest notion that is expected of any allocation – a matching is Pareto-optimal if no applicant can improve its allocation without *demoting* some applicant. Stronger notions of optimality like rank-maximality and fairness have been studied. Both these are appealing since rank maximal as well as fair matchings, apart from being Pareto-optimal, always exist, impose a total order on the set of admissible matchings in the instance, and are efficiently computable[10, 11, 19]. Rank-maximality and fairness can easily be described as an optimization over the *signatures.* For a matching M, the signature $\sigma(M)$ is a $r + 1$ tuple $(x_1, x_2, \ldots, x_r, x_{r+1})$ where x_i denotes the number of applicants

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

$a_1 : p_1, p_4$	$M_R = \{(a_1, p_1), (a_2, p_5), (a_3, p_6),$
$a_2 : p_1, p_2, p_5$	$(a_4, p_2), (a_5.p_4), (a_6, p_3)\}$
$a_3 : p_1, p_2, p_6$	$M_F = \{(a_1, p_4), (a_2, p_1), (a_3, p_2),$
$a_4 : p_2, p_3$	$(a_4, p_3), (a_5.p_5), (a_6, p_6)\}$
$a_5 : p_4, p_5$	$M = \{(a_1, p_4), (a_2, p_1), (a_3, p_6),$
$a_6 : p_3, p_6$	$(a_4, p_2), (a_5, p_5), (a_6, p_3)\}$

Figure 1: An instance with six applicants and six posts. Every post has unit quota. Preference lists of applicants to be read as: p_1 is rank 1 post of a_1, p_4 is the rank 2 post of a_1 and so on. M_R and M_F are the rank-maximal and fair matching respectively. M is an "in between" matching that is neither rank-maximal matching nor fair matching.

matched to their rank-i post. The value x_{r+1} denotes the number of applicants left unmatched by the matching M.

Rank-maximality: For two matchings M and M' with signatures $\sigma(M) = (x_1, x_2, \ldots, x_r, x_{r+1})$ and $\sigma(M') = (y_1, y_2, \ldots, y_r, y_{r+1})$, we say $M >_{\mathcal{R}} M'$, that is, M is better than M' w.r.t. rank-maximality if there exists an ℓ such that $1 \leq \ell \leq r$ such that $x_\ell > y_\ell$ and for $1 \leq j \leq \ell - 1$, we have $x_j = y_j$. A matching M is rank-maximal if M has the maximum signature under the ordering $>_{\mathcal{R}}$.

Fairness: For two matchings M and M' with signatures $\sigma(M) = (x_1, x_2, \ldots, x_r, x_{r+1})$ and $\sigma(M') = (y_1, y_2, \ldots, y_r, y_{r+1})$, we say that $M >_{\mathcal{F}} M'$, that is, M is better than M' w.r.t. to fairness if there exists an ℓ, $1 < \ell \leq r + 1$ such that $x_\ell < y_\ell$ and for $\ell + 1 \leq j \leq r + 1$, we have $x_j = y_j$. A matching M is fair if M has the maximum signature under the ordering $>_{\mathcal{F}}$.

Both rank-maximality and fairness impose a total order on the set of matchings admissible in the instance. Thus, all rank-maximal matchings have the same signature and the same holds for all fair matchings. Rank-maximal or fair matching optimize the head or tail end, respectively of the signature and hence one may perform poorly on the other criteria. Consider the matchings M_R and M_F as shown in Figure 1 which are rank-maximal and fair matchings in the instance with $\sigma(M_R) = (4, 0, 2, 0)$ and $\sigma(M_F) = (1, 5, 0, 0)$. As mentioned earlier, if the requirement is to match at least 50% of the applicants (3 out of 6) to their top choice and at least 75% of applicants to one of their first and second choice, neither M_R nor M_F satisfy the requirement. Note that the instance admits a matching M where $\sigma(M) = (3, 2, 1, 0)$ which satisfies the above requirement. The matching M lies "in between" M_R and M_F and although M is neither optimal w.r.t. to rank-maximality nor fairness, it is appealing since it is *more rank-maximal* than M_F and fairer than M_R. This motivates our new definition of comparing two signatures called *cumulative better signature*.

Cumulative better signature: Let $\rho = (x_1, \ldots, x_{r+1})$ and $\rho' = (y_1, \ldots, y_{r+1})$ denote two signatures. We say $\rho \geq_C \rho'$ (ρ is cumulative better signature than or equal to ρ') if

$$\sum_{j=1}^{\ell} x_j \geq \sum_{j=1}^{\ell} y_j \quad \text{for all } 1 \leq \ell \leq r + 1.$$

It is clear that a pair of signatures may be incomparable with respect to $\geq_C$, for instance $(1, 2, 0)$ and $(2, 0, 1)$ are incomparable with respect to $\geq_C$. However, if for two matchings M and M' we have $\sigma(M) \geq_C \sigma(M')$, it implies that M is better than or equal to M' with respect to *both* rank-maximality and fairness.

Now we redefine our goal in the standard setting. We are given an instance of the bipartite matching problem with one sided preferences and an input signature ρ. The goal is to decide whether the instance admits a matching M which is cumulative better than ρ and if so output the matching. In the instance in Figure 1 for the input signature $(2, 3, 1, 0)$, we observe that M satisfies the criteria that $\sigma(M) = (3, 2, 1, 0) \geq_C (2, 3, 1, 0)$. We remark that $\sigma(M) \not\geq_C \sigma(M_R)$ and $\sigma(M) \not\geq_C \sigma(M_F)$.

In most literature, matching with one-sided preferences is studied with fixed input quotas as described above. However, in practical applications, the quotas are determined by considerations like resource availability, classroom sizes, and the availability of teachers. Furthermore, these quotas may not be fixed and rigid as assumed in most of the works. Recent works on the stable allocation [7, 14, 20] addresses the issues with rigid quotas and modifies or extends the capacity of using flexible quotas.

In a similar spirit, we study a cost-based quota setting for our problem, which allows us to capture the fact that quotas may not be rigid but controlled by a cost associated with a post. We denote this as the *cost-based* quota setting as opposed to the standard-setting, which we denote as the *fixed quota* setting. In the cost-based setting, the input is a bipartite graph G, preferences associated with every applicant and a cost $c(p)$ associated with every post p, instead of the quota $q(p)$. The cost $c(p)$ denotes the cost of matching a single applicant to the post p. Given the input costs, the cost of a matching M is defined as $c(M) = \sum_{p \in \mathcal{P}} c(p) \cdot |M(p)|$, where $M(p)$ denotes the set of applicants matched to the post p. Our goal is to output a matching of minimum cost w.r.t the optimality criteria specified.

We investigate the new notion of optimality in both the fixed quota setting and the cost-based setting. We are now ready to formally define our problems. We assume that the input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$ where $\mathcal{A}$ denotes the set of applicants, $\mathcal{P}$ denotes the set of posts, and an edge $(a, p) \in E$ denotes that applicant a can be matched or assigned to the post p. Applicants rank posts in order of preference with ties allowed in preference lists. When quotas are a part of the input, we are in the fixed quota setting whereas when costs are a part of the input we are in the cost-based (quota) setting.

Fixed quota setting: A first problem that we investigate in this setting is : given an instance G of the one-sided preference list problem in the fixed quota setting, and a signature ρ, decide whether G admits a matching M such that $\sigma(M) = \rho$. We call this the exact signature in the fixed quota setting EXACT-SIGN-Q problem. We show that this problem is NP-hard under severe restrictions. Next, we investigate whether there exists a matching that is cumulative better than a given input signature ρ. We call this the cumulative better signature in the fixed quota setting CUM-SIGN-Q problem. We show that the CUM-SIGN-Q problem also turns to be NP-hard. We remark that this is in contrast to analogous questions that can be asked for rank-maximality or fairness. That is, given an instance G and a signature ρ, it is possible to answer in polynomial time whether G admits a matching M whose signature is more rank-maximal than ρ. This follows from the fact that a rank-maximal matching can be computed efficiently [11, 19]. The same holds true for fair matchings. We complement the hardness results for EXACT-SIGN-Q and CUM-SIGN-Q by presenting randomized polynomial-time algorithms for the case when the number of ranks is a constant. Important real-world applications of matching problems such as the National Residency Matching Program [12] and the Scottish Foundation Allocation Scheme [21] have bounds on the length of agent preference lists.

Cost-based setting: Next, we investigate the complexity of the above questions in the cost-based setting. Given an instance G of the one-sided preference list problem in the cost-based setting, and a signature ρ, does G admit a matching M such that $\sigma(M) = \rho$? We call this the EXACT-SIGN-C problem and show that the problem admits a polynomial-time algorithm using a reduction to the network flow problem. A natural question is to compute a matching that achieves the signature ρ at the minimum cost. We denote this problem as the exact signature at minimum cost in the cost-based setting, abbreviated as EXACT-SIGN-MIN-C.

Our next problem captures the fact that we are interested in a matching of minimum cost which has signature at least as good as the given signature. Given an instance G of the one-sided bipartite matching problem in the cost-based setting, and an input signature ρ, output a matching M (if one exists), such that signature of M is cumulative better than ρ and M has minimum cost amongst all matchings satisfying the property. We denote this problem as the cumulative better signature at minimum cost in the cost-based quotas setting, abbreviated as CUM-SIGN-MIN-C. In contrast to the fixed quota setting, the problem of achieving a matching with exact signature and cumulative better signature are efficiently solvable in the cost-based setting via suitable flow networks.

1.1 Our Results

Now we state our theoretical results formally.

Theorem 1.1. *The* EXACT-SIGN-Q *problem is NP-hard even when preferences of all applicants are strict and $q(p) = 1$ for every post $p \in P$. The* EXACT-SIGN-Q *problem admits a randomized polynomial time algorithm when the number of ranks is $O(1)$.*

Theorem 1.2. *The* CUM-SIGN-Q *problem is NP-hard even when preferences of all applicants are strict and $q(p) = 1$ for every post $p \in P$. The* CUM-SIGN-Q *problem admits a randomized polynomial time algorithm when the number of ranks is $O(1)$.*

Proposition 1.3. *The* EXACT-SIGN-C *problem admits a polynomial time algorithm via a single min-cost flow computation.*

Theorem 1.4. *The* EXACT-SIGN-MIN-C *and the* CUM-SIGN-MIN-C *problem both admit polynomial time algorithms via a single min-cost flow computation.*

We remark that the algorithms for the cost-based setting in Proposition 1.3 and Theorem 1.4 use flow networks in which edges have capacities (upper-bounds) as well as demands (lower-bounds). Flow networks have been used earlier to compute pareto-optimal matchings [4, 5] and popular and rank-maximal matchings [18] in the one-sided preference list setting.

Experimental evaluation: We complement our theoretical results by an experimental evaluation of the new notion of optimality for the fixed quota setting as well as the cost-based setting. We conduct our experiments on the available real-world data sets as well as synthetically generated data sets.

- **Fixed quota setting:** For each instance, we select an input signature ρ such that the instance admits a matching with signature which is cumulatively better than ρ. The rank-maximal matching as well as the fair matching in the instance both fail to satisfy the input requirement ρ. This illustrates that even when an instance admits a matching with the desired signature, rank-maximality or fairness need not satisfy the requirement.
- **Cost-based setting:** We begin with an instance of the fixed quota setting and select signature ρ such that the instance *does not admit* a matching which is cumulatively better than ρ in the fixed quota setting. We convert the input instance to a cost based setting instance by choosing three natural cost functions. Each of the cost functions assigns a cost to a post based on input quota and other parameters. For the cost-based setting for any signature ρ, there always exists a matching which is cumulatively better than ρ. We use our algorithm in Theorem 1.4 to compute a matching with minimum cost. We measure for every post the violation in its fixed quota with respect to our output min-cost matching. We report the maximum violation across all posts and the total violation. Our experiments suggest a possibility of transforming an instance of the fixed quota setting to an instance of the cost-based setting by an appropriate choice of cost function.

1.2 Related Works

The closest to our work in the flexible quota model is the recent work by [14] in which they study the stable matching setting where there is a cost c associated with matching an agent to a program. Recently, Gajulapalli et al. in [7] studied a two-round mechanism for the school choice problem in the two-sided setting. Here a stable matching is computed in the first round using the initial fixed quotas for every school. In the second round, some schools are allowed to increase their quota, or new students may participate in the allocation process, and a stable assignment, which is an extension of the stable matching in the first round, is computed. In a similar model by Rios et al. [20] in the context of the Chilean college admission process with only one-sided ties are allowed, flexible quotas are used as a tie-breaker.

Our exact signature problem is related to the rainbow matching and exact matching problems. In rainbow matching and related problems [3, 6, 13], the input is an edge-coloured graph, which need not be properly coloured. In some variants, the objective is to find a matching with the maximum number of colours. The Exact Matching problem [17] is also related to our problem, and we utilize this connection to design our randomized algorithms. The exact matching problem considers a graph whose edges are coloured red or blue and asks to find a perfect matching with exactly k red edges. It can be seen that both the exact matching problem as well as some rainbow matching problems on bipartite graphs ask for matchings with a particular signature when we have a bijection between ranks and colours. For example, asking for a matching with all of the colours is equivalent to asking for a matching with signature $(1, 1, \ldots, 1)$ if we have a bijection between colours and ranks. Similarly, the latter problem asks for a matching with signature $(k, n-k)$, where n is the number of vertices. The algorithm from [17] for the exact matching problem readily generalizes to solve EXACT-SIGN-Q. We modify this algorithm to solve the CUM-SIGN-Q problem. To the best of our knowledge, the isolation lemma from [17] has not been used before in the context of matchings with preferences.

2 ALGORITHMIC RESULTS IN THE COST-BASED SETTING

Our main technique in the cost-based setting is to reduce the problem to the max-flow problem[2] in a network having costs as well as demands (lower-bounds) on the edges. To begin with, we assume that the input instance admits a matching with signature ρ. We then show the flow network construction for the warm-up problem EXACT-SIGN-MIN-C and build upon that to solve the problem CUM-SIGN-MIN-C. Our reduction can be considered an extension of the standard bipartite matching problem to the flow network reduction. In this standard reduction, we have one node in the network for every vertex in the input graph G, and there are edges connecting source to applicant nodes, applicant nodes to post nodes and post nodes to the sink. In our flow network, we introduce rank nodes to capture the rank of every (a, p) edge in G. Instead of the direct connection, now the post nodes are connected to the sink via rank nodes. We guarantee the capture of the ranks by duplicating the post nodes r times where r is the largest possible rank in G. Now let the input signature ρ be $(x_1, x_2, \ldots, x_r, x_{r+1})$. We use demands on certain special edges (bold edges in Figure 2) to decide whether G has a matching with exact signature ρ. The idea is to set the demand on these special edges as its capacity such that it is possible to recover a matching from the flow network with x_i edges of rank-i. Now we describe our flow network construction formally.

2.1 Algorithm for EXACT-SIGN-MIN-C

We design a min-cost flow network $H = (V_H, E_H)$ with the following vertex and edge description. We call the vertices of V_H as nodes throughout the paper. The network H has the set of nodes

$$V_H = \{s, t, d\} \cup V_A \cup V_P \cup V_R \text{ where,}$$
$$V_A = \{a_1, a_2, \ldots, a_n\}$$
$$V_P = \{p_i^{(j)} \mid 1 \leq i \leq |\mathcal{P}|,\ 1 \leq j \leq r\}$$
$$V_R = \{r^{(1)}, r^{(2)}, \ldots, r^{(r+1)}\}.$$

Assuming $|\mathcal{A}| = n$, we have n applicant nodes in V_A, one per each applicant in $\mathcal{A}$, $r+1$ rank nodes in V_R, and for every post p_i in $\mathcal{P}$, we make a copy of it for each rank, and hence we have $|\mathcal{P}|\, r$ post nodes in V_P. In addition to these nodes, we have a source s, sink t, and a dummy node d that lets us capture the number of unmatched applicants. For a rank-j, we denote the rank node as $r^{(j)}$, and for a post $p_i \in \mathcal{P}$, the corresponding post node as $p_i^{(j)}$. See Figure 2 for an illustration. Every edge e in the network has a cost $c(e)$, a demand $q^-(e)$, and a capacity $q^+(e)$, represented by

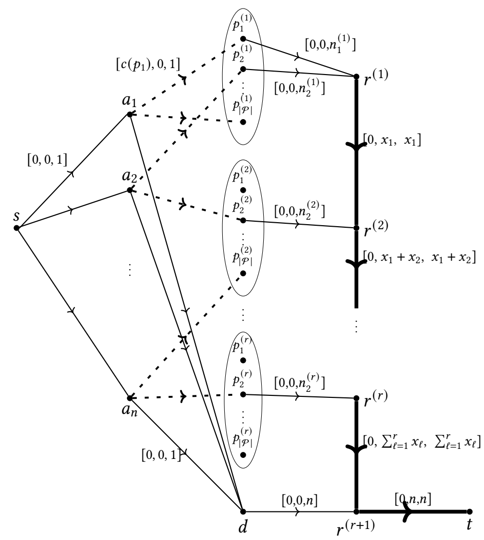

Figure 2: The min-cost flow network with demands on edges (H) for EXACT-SIGN-MIN-C**. Every edge e in H has an associated vector$[c(e), q^-(e), q^+(e)]$. All edges from s to applicants, and applicants to dummy post d have vector $[0, 0, 1]$. All edges except dashed edges has a cost of 0, and all edges except bold edges has a demand of 0.**

the vector $[c(e), q^-(e), q^+(e)]$. Now we define the edges and their corresponding vector. The edge set

$$E_H = E_{sA} \cup E_{AP} \cup E_{Ad} \cup E_{PR} \cup E_{RR} \cup \{(d, r^{(r+1)}), (r^{(r+1)}, t)\}, \text{ where}$$

- $E_{sA} = \{(s, a) \mid a \in V_A\}$ are the set of edges from source node to applicant nodes with vector as $[0, 0, 1]$.
- $E_{AP} = \{(a, p_i^{(j)}) \mid a \in V_A, p_i^{(j)} \in V_P, p_i$ is rank-j post of a in $G\}$ are the set of edges from applicant nodes to post nodes with vector as $[c(p_i), 0, 1]$. Recall that $c(p_i)$ is the cost of matching one applicant to the post p_i in G.
- $E_{Ad} = \{(a, d) \mid a \in V_A\}$ are the set of edges from applicant nodes to the dummy node d with vector as $[0, 0, 1]$.
- $E_{PR} = \{(p_i^{(j)}, r^{(j)}) \mid p_i^{(j)} \in V_P,\ r^{(j)} \in V_R \setminus r^{(r+1)}\}$, are the set of edges from post nodes to the rank nodes with vector as $[0, 0, n_i^{(j)}]$, where $n_i^{(j)}$ is the number of applicants who treat p_i as rank-j post in G.
- $E_{RR} = \{(r^{(j)}, r^{(j+1)}) \mid r^{(j)}, r^{(j+1)} \in V_R\}$, are the edges connecting node $r^{(j)}$ to the node $r^{(j+1)}$ with corresponding vector as $[0, \sum_{\ell=1}^{j} x_\ell, \sum_{\ell=1}^{j} x_\ell]$.

The edge $(d, r^{(r+1)})$ connects dummy node to the rank node with vector as $[0, 0, n]$, and the edge $(r^{(r+1)}, t)$ from rank node $r^{(r+1)}$ to sink t has vector $[0, n, n]$. In the network H, all edges except the edges in E_{AP} (see dashed edges in Figure 2) have cost 0, and these edges control the cost of the flow, which corresponds to the cost of the matching. Observe that in H, all edges except $E_{RR} \cup \{(r^{(r+1)}, t)\}$ (see bold edges in Figure 2) have a demand of 0, and these edges control the signature of the output matching. We remark that a natural way to connect posts to sink is by adding a directed edge from rank node $r^{(j)}$ to sink t with a vector $[0, x_j, x_j]$; however, we chose the connections as mentioned above since it allows us to readily adapt the network H to solve the cumulative signature problem. In order to argue the correctness of our reduction, it is useful to extend the definition of a signature of a matching to the signature of a flow in the network H.

Definition 2.1 (Signature of flow in the network H). The signature of a flow f in the network H is an $r+1$ tuple denoted by $\sigma(f) = (y_1, y_2, \ldots, y_r, y_{r+1})$. We define each component y_j of $\sigma(f)$ as:

$$y_j = \begin{cases} f\left((r^{(1)}, r^{(2)})\right), & j = 1 \\ f\left((r^{(j)}, r^{(j+1)})\right) - f\left((r^{(j-1)}, r^{(j)})\right), & 2 \le j \le r \\ f\left((r^{(r+1)}, t)\right) - f\left((r^{(r)}, r^{(r+1)})\right), & j = r+1 \end{cases}$$

We defer the detailed proofs to the full version for all the results presented in this section and give only the key ideas. We claim the following results about the flow network H.

Lemma 2.2. *Every feasible flow f in H has the signature $\sigma(f) = \rho$.*

Proof Sketch. Observe that the bold edges in Figure 2 capture the signature of the flow. Note that the demand on these edges is exactly equal to their capacity; hence, from Definition 2.1 any feasible flow f in H has the signature $\sigma(f) = \rho$. □

Lemma 2.3. *M is a matching in G with signature $\sigma(M) = \rho$ iff there is a feasible flow f in H such that $c(f) = c(M)$*

Proof Sketch. Given a matching M in G with $\sigma(M) = \rho$, we construct a feasible flow f in H as follows. If an applicant $a \in M(p_i)$, and p_i is the rank-j post of a, route 1 unit of flow through the path $\langle s, a, p_i^{(j)}, r^{(j)}, r^{(j+1)}, \ldots, r^{(r+1)}, t\rangle$. For an unmatched applicant a, route 1 unit of flow through the path $\langle s, a, d, r^{(r+1)}, t\rangle$.

For the other direction, we construct the matching M in G corresponding to a feasible flow f in H as follows. For an applicant a, consider the corresponding simple s to t path carrying 1 unit of flow. If the path is $\langle s, a, p_i^{(j)}, r^{(j)}, r^{(j+1)}, \ldots, r^{(r+1)}, t\rangle$, add (a, p_i) in M. Observe that $c(M) = c(f)$ in both cases. □

Thus, we can solve EXACT-SIGN-MIN-C using a single min-cost flow computation with demands on edges – this proves the first part of Theorem 1.4. The time required to solve EXACT-SIGN-MIN-C is at most the time needed to solve the min-cost flow network H with demands. The running time required to solve min-cost flow network problem is $O((|E_H| \log |V_H|)(|E_H| + |V_H| \log |V_H|))$ using the enhanced capacity scaling algorithm [2]. If we omit the costs in H, the question of whether the instance admits a feasible flow will solve the EXACT-SIGN-C problem using a single min-cost flow computation with demands on edges – this proves Proposition 1.3.

2.2 Algorithm for CUM-SIGN-MIN-C

In our EXACT-SIGN-MIN-C problem, we were looking for exactly x_i many applicants to be matched to rank-i, and to ensure this, we set the capacity of our bold edges equal to its demand. However, in CUM-SIGN-MIN-C problem, it is desirable to match more applicants to a particular rank-i while maintaining the cumulative sum of the number of applicants matched up to rank-i at least as good as that in the input signature. To maintain the cumulative better signature property, we keep the demand on the bold edges as the same but, to allow the possibility of matching more applicants, we set the capacity on these edges as n. That is, we do a simple modification to the network H by setting the capacity of all the edges in E_{RR} as n, and we keep everything else the same. We call the modified network $\tilde{H}$ and claim the following results, and these results, in turn, prove the second part of Theorem 1.4.

Lemma 2.4. *Every feasible flow f in $\tilde{H}$ has the signature $\sigma(f) \ge_C \rho$.*

Corollary 2.5. *Every feasible flow in $\tilde{H}$ has signature $\sigma(f) \ge_C \rho$ and $\sigma(f)$ is better than ρ with respect to rank-maximality and fairness. That is $\sigma(f) \ge_{\mathcal{R}} \rho$ and $\sigma(f) \ge_{\mathcal{F}} \rho$*

Given an instance G and a signature ρ, our goal is to output a min-cost matching M with signature $\sigma(M) \ge_C \rho$. Recall that there can be many matchings with their signature cumulatively better than ρ, and the network $\tilde{H}$ can output any one of these. However, we can ask for some additional criteria such as amongst all the min-cost matchings that have a signature cumulatively better than ρ, output one which is rank-maximal or fair. We defer these details in the full version of the paper.

3 RANDOMIZED ALGORITHMS IN THE FIXED QUOTA SETTING

Having considered the cost-based setting, we now return to the fixed quota setting. Given a graph $G = (\mathcal{A} \cup \mathcal{P}, E)$ and an input signature ρ, we say that any matching M with $\sigma(M) = \rho$ is an ESM (or Exact Signature Matching) for (G, ρ) and any matching M with $\sigma(M) = \rho' \ge_C \rho$ is a CSM (or Cumulative Signature Matching) for (G, ρ). The problems EXACT-SIGN-Q and CUM-SIGN-Q turn out to be related, and we solve their decision version using similar techniques. As mentioned earlier, we utilize the connection between our problem and the exact matching problem and design a randomized algorithm that outputs YES with probability $> 1/2$ for YES instances and outputs NO with probability 1 for NO instances. By applying standard techniques, this algorithm can be repeatedly used to recover the desired matching with high probability.

3.1 Algorithm for EXACT-SIGN-Q

We can assume without loss of generality that we are in the one-to-one setting, i.e. that every post has capacity 1. If not, create $q(p)$ copies of that post, each of capacity 1 and the same neighbourhood. Let the applicants rank each of these posts with the same rank. This adds a polynomial overhead to the instance size since the number of posts increases by at most a factor of $|\mathcal{A}|$. We may also assume $\sum_{1 \le i \le r} \rho_i = |\mathcal{A}| = |\mathcal{P}|$, so that we are effectively testing the existence of some perfect matching. We remark that this is without loss of generality since we may add some dummy applicants and posts to ensure this condition. We defer the details to the full version. We now describe the randomized algorithm from Theorem 1.1.

Proof of Theorem 1.1(Algorithm). We first assign weights according to a process from [17] to get a weight w_e for every edge e. We then come up with a modified version of their isolation lemma.

The process of adding weights is as follows : for every edge, assign to it a weight randomly chosen from $\{1, 2, \ldots, 2|E|\}$. We use Lemma 3.1 in the same spirit as in [17]. The proof is identical to theirs, and follows by restricting calculations to ESMs instead of perfect matchings. See full version of the paper for the proof.

Lemma 3.1 (Isolation Lemma for ESMs).

$$\Pr[G \text{ has a unique minimum weight ESM}] \geq \frac{1}{2}.$$

Now, we construct a modified adjacency matrix B for the graph G. Define

$$B_{u,v} = \begin{cases} 0 & \text{if } (u,v) \notin E \\ 2^{w_{(u,v)}} y_i & \text{if } (u,v) \in E \text{ and has rank-}i \end{cases}$$

where y_i is a new variable that we define. This is enough for us to solve the exact version. Since we assumed that $|\mathcal{A}| = |\mathcal{P}|$, we have that B is a square matrix. Let it have dimension $n \times n$. Now consider $\det(B)$. Let S_n be the set of all permutations of $[n]$. Then we have

$$\det(B) = \sum_{s \in S_n} (-1)^{\text{sgn}(s)} \prod_{u \in [n]} B_{u,s(u)}.$$

It can be seen as a polynomial in the variables $y_1, y_2, \ldots, y_r$. We claim that this polynomial has a non-zero coefficient for the monomial $\pi = \prod_i y_i^{\rho_i}$ with probability at least $\frac{1}{2}$ if G has a matching with signature ρ. The reverse direction is immediate, since π can appear only if there is a matching with exactly ρ_i edges of rank-i for every i.

Now we look at the terms that make up coefficient of π. If there is a unique ESM, then we are done since the coefficient has to be non-zero. Suppose not, assume that there are ESMs $M_1, M_2, \ldots, M_t$. Let them be ordered by weight so that $w(M_1) \leq w(M_2) \leq \ldots$. Then, we have that the coefficient of π is

$$\begin{aligned} &\pm 2^{w(M_1)} \pm 2^{w(M_2)} \pm \ldots \pm 2^{w(M_t)} \\ &= 2^{w(M_1)} \left(\pm 1 \pm 2^{w(M_2) - w(M_1)} \pm \ldots \pm 2^{w(M_t) - w(M_1)} \right). \end{aligned}$$

Note that the signs ($\pm$) depend on the parity of the corresponding permutation. By Lemma 3.1, we have that there is a unique minimum weight matching with probability at least $\frac{1}{2}$. Then with the same probability, we have that the term inside the parentheses must be an odd integer, since it is an odd integer (± 1) added to a sequence of even integers ($2^{w(M_i) - w(M_1)}$ is even since the term in the exponent is a non-zero positive integer). Thus, the coefficient is non-zero with probability at least $\frac{1}{2}$. □

3.2 Algorithm for CUM-SIGN-Q

We can again assume without loss of generality that we are testing for a perfect matching. We now describe the randomized algorithm from Theorem 1.2. The main difficulty in generalizing the algorithm for EXACT-SIGN-Q is that an input signature ρ to CUM-SIGN-Q does not give us any handle on the signature of the output matching. For example, an input $\rho = (1, 1)$ gives us two possible matchings that would satisfy the constraints. One with signature ρ itself and one with signature $(2, 0)$. We can possibly iterate over all possible signatures ρ' such that $\rho' \geq_C \rho$ and apply the algorithm from Theorem 1.1. This requires possibly n^r calls to the previous algorithm. We design an alternate algorithm that avoids this brute-force method but does not improve the complexity owing to overhead due to arithmetic on large numbers. We do so by letting a rank-i edge act as a rank-j for all $j \geq i$. This solves the previous problem because now the matching with signature $(2, 0)$ also 'acts' as a matching of signature $(1, 1)$. We achieve this by setting the adjacency matrix have the term $\sum_{j \leq i \leq r} y_i$ instead of just y_j for a rank-j edge. The problem here is that these terms give rise to additional binomial coefficients which require a new analysis of the algorithm. As an illustrative example, consider a matching with signature $(2, 0)$. This would contribute $(y_1 + y_2)^2 = y_1^2 + 2y_1y_2 + y_2^2$ to the determinant. Observe that the coefficient of the y_1y_2 term is different than that of the y_1^2 term.

Proof of Theorem 1.2(Algorithm). This time, we need a different allocation of weights to edges. For rank-i edges, we give its weight randomly from $\{k_i + 1, k_i + 2, \ldots, k_i + 2|E|\}$ for $k_i = 2r \cdot |E| (|E| + 1) \cdot n^{2(r-i)+1}$. We first come up with a similar isolation lemma as before, and we define some additional notation. Suppose that two matchings M_1, M_2 are both CSMs with $\sigma(M_1) = \rho_1$, and $\sigma(M_2) = \rho_2$. Observe that we can easily compare ρ_1, ρ_2 with respect to rank-maximality. We call a matching M to be a BestCSM if it is rank-maximal among all CSMs.

Lemma 3.2 (Isolation Lemma for BestCSMs).

$$\Pr[G \text{ has a unique min weight BestCSM}] \geq \frac{1}{2}.$$

The proof of the lemma is similar to that of Lemma 3.1. Now, we define the matrix B as

$$B_{u,v} = \begin{cases} 0 & \text{if } (u,v) \notin E \\ 2^{w_{(u,v)}} \left(\sum_{j \leq i \leq r} y_i \right) & \text{if } (u,v) \in E \text{ with rank-}j \end{cases}$$

The idea here is that a rank-j edge can contribute y_i for any $i \geq j$ to a monomial. The problem is that there will be additional binomial coefficients that contribute to the coefficient of some monomials. We avoid this by looking at BestCSMs in a certain way. Like before, our claim is that $\det(B)$ has a non-zero coefficient for the monomial $\pi = \prod_i y_i^{\rho_i}$ with probability at least $\frac{1}{2}$ if G has CSM.

Look at the coefficient of π. If there is a unique CSM, then we are done because the coefficient will be non-zero. If not, then look at the BestCSMs. Since they all have the same signature, the binomial coefficient associated with them, say α, will be the same. Now, look at the coefficient of π in $\det(B)$, we have that it is consists of contributions by BestCSMs, which we call β_1 and contributions by other CSMs, which we call β_2. We will look at these separately. Consider β_1. Let $M_1, M_2, \ldots$ be the BestCSMs arranged in increasing order of weight. Since we argued that they will have the same signature, their contribution will be of the form

$$\alpha \cdot 2^{w(M_1)} \left(\pm 1 \pm 2^{w(M_2) - w(M_1)} \pm 2^{w(M_3) - w(M_1)} \ldots \right).$$

With probability at least $\frac{1}{2}$, there is a unique minimum weight BestCSM, so all terms except the first one will be even. Like before, this will thus contribute a non-zero term. Also observe that we have that $|\beta_1| \geq 2^{w(M_1)}$ since all the terms and α are at least 1. Now, we look at β_2. We only need to show that $|\beta_2| < |\beta_1|$. This will ensure that the coefficient of π is non-zero because β_2 cannot cancel out β_1. It can be shown by a simple combinatorial argument that $|\beta_2| \leq 2^{w(M')+r|E|}$. See full version for the details. The main argument is that since these matchings are not BestCSMs, they have signature strictly worse than the BestCSMs and contribute a much smaller amount to the coefficient. Then we have that $|\beta_1| - |\beta_2| > 0$. Thus, with probability at least $\frac{1}{2}$, the coefficient of π is non-zero. □

Running time: In both of these cases, the overall running time is primarily decided by the step where we check the coefficient of π. This can be done with multi-variate polynomial interpolation. For example, through the approach in [8], we can do this in time n^{r^2}. The other manipulations of the determinant and the instance require at most $O(n^4)$ operations, giving an overall runtime of $O(n^{r^2+4})$. The arithmetic on large numbers adds an overhead of $O(n^r)$ for the case of cumulative better signatures, giving an overall runtime of $O(n^{r^2+r+4})$. When $r = O(1)$, the algorithm runs in polynomial time in either case.

4 HARDNESS RESULTS IN THE FIXED QUOTA SETTING

We prove our hardness results when every post $p \in \mathcal{P}$ has a unit quota, and every applicant has strict preferences over the posts. All our hardness results are based on the hardness of a special case of labelled perfect matching defined in [16]. We note that there is a rich literature on similar problems under various other names such as rainbow matching (for example, see [13]). We show that a variant of the problem in [16], that we call LABELLED MATCHING, is also NP-hard. We define this as follows.

Problem 4.1 (LABELLED MATCHING). *Given a bipartite graph $G = (A \cup B, E)$ with $|A| = |B| = n$, and $E = E_1 \cup E_2 \cup \ldots \cup E_r$ be a partition of edges such that every vertex is adjacent to at most one edge from each E_i, does the instance admit a matching M such that $|E_i \cap M| \geq 1$ for every $i \in [r]$?*

Proposition 4.2. LABELLED MATCHING *is NP-hard even on 2-regular graphs.*

Now we go back to our original problem of EXACT-SIGN-Q. We show the hardness reduction from LABELLED MATCHING to EXACT-SIGN-Q to prove our hardness part of Theorem 1.1. We defer the proof to the full version of the paper. The main idea is to treat the partition of edges $E_1, E_2, \ldots, E_r$ from LABELLED MATCHING as the ranks for the applicants. However, this might cause there to be 'gaps' in the preference lists of applicants. For example, a vertex incident to an edge from E_1 and E_3 but not E_2 would have a rank-1 and rank-3 post, but not a rank-2 post. This can be avoided by adding dummy applicants and posts in a careful way. Next, we show the hardness of CUM-SIGN-Q. We show a reduction from EXACT-SIGN-Q.

Proof of Theorem 1.2(Hardness). Given an instance of EXACT-SIGN-Q on a graph G with input signature $\rho = (\rho_1, \rho_2, \ldots, \rho_r)$ with $r \geq 2$, we reduce to an instance of CUM-SIGN-Q with $2r + 1$ ranks. We assume without loss of generality that $V = \mathcal{A} \cup \mathcal{P}$ with $|\mathcal{A}| = |\mathcal{P}| = n$ and $\sum_{1 \leq j \leq r} \rho_j = n$. Let the applicants and posts be numbered from 1 to n.

Let $G = (V, E_1 \cup E_2 \ldots \cup E_r)$ where E_j is the set of rank-j edges. We construct a graph H as follows. Make r copies of the vertex set $V = \mathcal{A} \cup \mathcal{P}$ times to get graphs $G_1, G_2, \ldots, G_r$ where G_j has vertex set $(\mathcal{A}^{(j)} \cup \mathcal{P}^{(j)})$ and only contains rank-j edges, i.e. E_j. Let the edges of E_j have rank-$2j$ in G_j. Let $a_i^{(j)}$ represent the copy of a_i in G_j and similarly let $p_i^{(j)}$ represent the copy of p_i in G_j. We add $(r-1) \cdot n$ dummy posts and applicants

$$D_P = \{d_p^{(1,1)}, \ldots, d_p^{(1,n)}, d_p^{(2,1)} \ldots, d_p^{(r-1,n)}\}$$

$$D_A = \{d_a^{(1,1)}, \ldots, d_a^{(1,n)}, d_a^{(2,1)} \ldots, d_a^{(r-1,n)}\}$$

such that for every i, j, $d_p^{(1,i)}, \ldots, d_p^{(r-1,i)}$ are all connected to copies of $a_i^{(j)}$ through rank-1 edges and for every i, j, $d_a^{(1,i)}, \ldots, d_a^{(r-1,i)}$ are all connected to $p_i^{(j)}$ through rank-$2j+1$ edges.

Lemma 4.3. *There is a matching M in G with $\sigma(M) = \rho$ iff there is a matching M' in H such that $\sigma(M') \geq_C ((r-1) \cdot n, \rho_1, n - \rho_1, \rho_2, n - \rho_2, \ldots, n - \rho_r) = \pi$.*

Given M in G, it is easy to recover M' in H. We do the following: for all rank-j edges matched in G, match the corresponding edge in G_j in H. For all the unmatched vertices, match them to dummy applicants or posts. It can be observed that $\sigma(M') = \pi \geq_C \pi$.

Now suppose we have a matching M in H with $\sigma(M) = \pi' \geq_C \pi$. We will show that $\pi' = \pi$. This can be done via a series of claims whose proofs we defer to the full version.

Claim 4.4. *M is a perfect matching for H.*

Claim 4.5. *For every $a_i \in \mathcal{A}$, exactly $r-1$ copies of a_i are matched to the dummy posts D_P. For every $p_i \in \mathcal{P}$, exactly $r-1$ copies of p_i are matched to the dummy applicants D_A.*

Claim 4.6. *For every $j \geq 1$, $\pi'_{2j} \geq \rho_j$ and $\pi'_{2j} + \pi'_{2j+1} = n$.*

Lemma 4.7. *$\pi' = \pi$.*

Now, we need to show that there is a matching in G that achieves signature ρ. We do this in the following way: for every applicant $a \in \mathcal{A}$, match it to $p \in \mathcal{P}$ such that some copy of a is matched to some copy of p in M. From Claim 4.5, this gives us a matching since every applicant is matched to exactly one non-dummy post and vice versa. Consider Lemma 4.7 and the observation that rank-$2j$ edges in H correspond to rank-j edges in G, the corresponding matching has signature ρ. This concludes the proof of Lemma 4.3. We observe that the above constructed instances have ties and gaps in the preference lists. These can be removed via some minor bookkeeping. This can be done in a manner similar to that in the proof of hardness part of Theorem 1.1. We defer these details to the full version. This then concludes the proof of hardness part of Theorem 1.2. □

5 EXPERIMENTS

We present the empirical evaluation of the cumulative better signature for the fixed quota setting as well as the cost-based setting. We report results on available real-world data sets as well as synthetically generated data sets. The experiments were conducted on a laptop running on a 64-bit Windows 10 Home edition and equipped with an Intel Core i5-8250U CPU @1.60GHz and 8GB of RAM. We used IBM ILOG CPLEX Optimization Studio 20.1 with Python APIs

Table 1: Fixed quota setting results on the real-world data sets. All values are given in percentage. The cells with background green meet the input requirement whereas the ones with background pink fail to meet the input requirement.

Data set	Input requirement			CSM			RMM			FM		
	#rank 1	#top 3 ranks	size	#rank 1	#top 3 ranks	size	#rank 1	#top 3 ranks	size	#rank 1	#top 3 ranks	size
Real-1	≥ 60	≥ 85	≥ 90	60.04	85.00	91.23	66.03	81.41	85.96	48.73	84.48	93.82
Real-2	≥ 60	≥ 80	≥ 85	60.02	80.01	85.55	61.88	78.02	82.99	51.82	80.59	88.74
Real-3	≥ 70	≥ 90	≥ 95	70.01	90.02	95.49	73.02	87.55	93.13	60.84	90.45	97.42

Table 2: Cost-based setting results on the real-world data sets. All values are given in percentage.

Data set	Input requirement			CF-1		CF-2		CF-3	
	#rank 1	#top 3 ranks	size	max violation	total violation	max violation	total violation	max violation	total violation
Real-1	≥ 65	≥ 85	≥ 90	**3.21**	14.81	**5.37**	10.25	**10.73**	17.73
Real-2	≥ 65	≥ 80	≥ 85	**3.97**	17.38	**4.43**	13.33	**4.06**	14.86
Real-3	≥ 75	≥ 90	≥ 95	**5.04**	10.46	**4.56**	12.34	**3.65**	11.21

to solve integer linear programs. The results on synthetic data sets are deferred to the full version.

Real-world data sets: The data sets Real-1, Real-2 and Real-3 are obtained from the elective allocation at an educational institution for three different periods. Each data set has around 2000 students (applicants in our model) and 100 courses (posts in our model). Each course has an upper bound on the number of students it can take, and every student has a strict preference ordering over the courses the student is interested in. A student needs to be assigned to at most one course.

Fixed quota setting: For each data set we select an input signature ρ such that the instance admits a matching which is cumulatively better than ρ. The signature is selected such that it is practically appealing in real-world applications. However, we observe that, neither the rank-maximal matching nor the fair matching are able to meet the requirement in terms of the signature. Table 1 shows our results in this setting. For the data set Real-1 suppose the requirement is to match at least 60% students to their rank-1 courses, at least 85% students to one of their top 3 courses, and the size of the matching must be at least 90% (see row-1 column Input requirement Table 1). The size of the rank-maximal matching (RMM) is 86% thus not meeting the requirement of 90% students being matched. Similarly, the fair matching (FM) matches only 49% students to their rank-1 courses thus failing to meet the 60% requirement of rank-1 matches. See row-1 columns RMM and FM in Table 1. The RMM and FM are obtained by implementing the known algorithms in [10, 11] respectively. The CSM is obtained by an linear linear program formulation (CSM ILP). By choice of the signature, the CSM satisfies the requirement.

Cost-based setting: We begin with an instance of the fixed quota setting. We obtain the cost for every post p by defining a function which takes as input the input quota $q(p)$ and the number of applicants $\#N(p)$ who applied to p. In fact we define three natural cost functions called CF-1, CF-2, CF-3. Using these we derive different instances of the cost-based setting from a single instance of the fixed quota setting. For an instance in the fixed quota setting, we select a signature ρ such that the input fixed quota instance *does not* admit a matching that is cumulatively better than ρ. For each of the three instances derived from this instance, we use our algorithm in Theorem 1.4 and compute a matching with minimum cost. The absolute cost of the matching obtained is not of significance. Since we started with a fixed quota setting instance, we measure the violation in the upper quota for every course, and we report the *max violation* (bold font) and *total violation*. Table 2 gives our results for the real-world data sets. We observe that the maximum violation for any course is around 5% (except for Real-1 with cost function CF-3) and the total violation is around 15%. We believe these are acceptable values in practice in order to meet the signature requirement of the allocation. We briefly describe our cost functions below:

- CF-1: $c(p) = max\{0, (\#N(p) - q(p))\}$
- CF-2: linear cost model that assign costs in non-decreasing order of the ratio $\frac{\#N(p)}{q(p)}$.
- CF-3: $c(p) = \frac{\#N(p).LCM}{q(p)}$, where LCM represents the least common multiple of all the quotas.

We make the following observations from our experiments:

- The cumulatively better signature allows us to express requirements which the input instance may admit and is not captured by the standard measures like rank-maximality or fairness.
- The cost based setting experiments open up the possibility of converting a fixed quota setting instance to a cost-based instance by selecting an appropriate cost function. Thus, although the input instance may not admit a matching with the desired signature, we may be able to achieve the same by violating the quotas by a small value.

ACKNOWLEDGMENTS

This work was partially supported by the grant CRG/2019/004757. We thank Girija Limaye for fruitful discussions that helped us shape our work in the initial phases. We also thank the anonymous reviewers for their valuable feedback.

REFERENCES

[1] David J Abraham, Katarína Cechlárová, David F Manlove, and Kurt Mehlhorn. 2004. Pareto optimality in house allocation problems. In *International Symposium on Algorithms and Computation*. Springer, 3–15.

[2] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. 1988. Network flows. (1988).

[3] Noga Alon, Tao Jiang, Zevi Miller, and Dan Pritikin. 2003. Properly colored subgraphs and rainbow subgraphs in edge-colorings with local constraints. *Random Structures & Algorithms* 23, 4 (2003), 409–433. https://doi.org/10.1002/rsa.10102 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.10102

[4] Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, David Manlove, Ioannis Mourtos, Eva Oceláková, and Baharak Rastegari. 2016. Pareto optimal matchings in many-to-many markets with ties. *Theory of Computing Systems* 59, 4 (2016), 700–721.

[5] Katarína Cechlárová and Tamás Fleiner. 2017. Pareto optimal matchings with lower quotas. *Mathematical Social Sciences* 88 (2017), 3–10.

[6] Alan Frieze and Michael Krivelevich. 2008. On rainbow trees and cycles. *Electronic Journal of Combinatorics* (2008), 59.

[7] Karthik Gajulapalli, James Liu, Tung Mai, and Vijay V Vazirani. 2019. Stability-Preserving, Time-Efficient Mechanisms for School Choice in Two Rounds. *arXiv e-prints* (2019), arXiv–1904.

[8] M Hecht, Bevan L Cheeseman, Karl B Hoffmann, and Ivo F Sbalzarini. 2017. A quadratic-time algorithm for general multivariate polynomial interpolation. *arXiv preprint arXiv:1710.10846* (2017).

[9] Hadi Hosseini, Vijay Menon, Nisarg Shah, and Sujoy Sikdar. 2021. Necessarily Optimal One-Sided Matchings. In *Proceedings of Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 5481–5488. https://ojs.aaai.org/index.php/AAAI/article/view/16690

[10] Chien-Chung Huang, Telikepalli Kavitha, Kurt Mehlhorn, and Dimitrios Michail. 2016. Fair matchings and related problems. *Algorithmica* 74, 3 (2016), 1184–1203.

[11] Robert W Irving, Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna E Paluch. 2006. Rank-maximal matchings. *ACM Transactions on Algorithms (TALG)* 2, 4 (2006), 602–610.

[12] Fuhito Kojima, Parag A Pathak, and Alvin E Roth. 2013. Matching with couples: Stability and incentives in large markets. *The Quarterly Journal of Economics* 128, 4 (2013), 1585–1632.

[13] Van Bang Le and Florian Pfender. 2014. Complexity results for rainbow matchings. *Theoretical Computer Science* 524 (2014), 27–33. https://doi.org/10.1016/j.tcs.2013.12.013

[14] Girija Limaye and Meghana Nasre. 2021. Stable Matchings with Flexible Quotas. *arXiv preprint arXiv:2101.04425* (2021).

[15] David F. Manlove and Colin T. S. Sng. 2006. Popular Matchings in the Capacitated House Allocation Problem. In *Algorithms – ESA 2006*, Yossi Azar and Thomas Erlebach (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 492–503.

[16] Jérôme Monnot. 2005. The labeled perfect matching in bipartite graphs. *Inform. Process. Lett.* 96, 3 (2005), 81–88.

[17] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. 1987. Matching is as Easy as Matrix Inversion. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC '87)*. 345–354.

[18] Meghana Nasre, Prajakta Nimbhorkar, and Nada Pulath. 2019. Classified Rank-Maximal Matchings and Popular Matchings - Algorithms and Hardness. In *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG 2019, Vall de Núria, Spain, June 19-21, 2019, Revised Papers (Lecture Notes in Computer Science, Vol. 11789)*, Ignasi Sau and Dimitrios M. Thilikos (Eds.). Springer, 244–257. https://doi.org/10.1007/978-3-030-30786-8_19

[19] Katarzyna Paluch. 2013. Capacitated rank-maximal matchings. In *International Conference on Algorithms and Complexity*. Springer, 324–335.

[20] Ignacio Rios, Tomas Larroucau, Giorgiogiulio Parra, and Roberto Cominetti. 2021. Improving the Chilean college admissions system. *Operations Research* (2021).

[21] Scottish Foundation Allocation Scheme. 2011 - 2022. *Matching in Practice,European network for research on matching practices in education and related markets @online*. https://www.matching-in-practice.eu/the-scottish-foundation-allocation-scheme-sfas/