

Group Fair Matchings Using Convex Cost Functions

Atasi Panda¹, Harsh Sharma², Anand Louis^{1*}, and Prajakta Nimbhorkar^{2*}

¹ Indian Institute of Science

² Chennai Mathematical Institute

{atasipanda, anandl}@iisc.ac.in, {harshs.ug2023, prajakta}@cmi.ac.in

Abstract

We consider the problem of assigning items to platforms where each item has a utility associated with each of the platforms to which it can be assigned. Each platform has a soft constraint over the total number of items it serves, modeled via a convex cost function. Additionally, items are partitioned into groups, and each platform also incurs group-specific convex cost over the number of items from each group that can be assigned to the platform. These costs promote group fairness by penalizing imbalances, yielding a soft variation of fairness notions introduced in prior work, such as Restricted Dominance and Minority protection. Restricted Dominance enforces upper bounds on group representation, while Minority protection enforces lower bounds. Our approach replaces such hard constraints with cost-based penalties, allowing more flexible trade-offs. Our model also captures Nash Social Welfare kind of objective.

The cost of an assignment is the sum of the values of all the cost functions across all the groups and platforms. The objective is to find an assignment that minimizes the cost while achieving a total utility that is at least a user-specified threshold. The main challenge lies in balancing the overall platform cost with group-specific costs, both governed by convex functions, while meeting the utility constraint. We present an efficient polynomial-time approximation algorithm, supported by theoretical guarantees and experimental evaluation. Our algorithm is based on techniques involving linear programming and network flows. We also provide an exact algorithm for a special case with uniform utilities and establish the hardness of the general problem when the groups can intersect arbitrarily. This work has applications in cloud computing, logistics, resource constrained machine learning deployment, federated learning, and network design, where resources must be allocated across platforms with diverse cost structures and diminishing returns.

Code — <https://github.com/codula-code/GFMCCF>

Datasets — <https://grouplens.org/datasets/movielens/100k/>

Extended version — <https://arxiv.org/abs/2508.12549>

1 Introduction

Bipartite graphs are a well-established framework for solving resource allocation and matching problems across diverse

domains. In this paper, we work with bipartite graphs where the underlying partitions are referred to as *items* and *platforms*. The goal is to compute a many-to-one matching on this graph, subject to a set of constraints. Each item has utilities for the platforms in its neighborhood, and each platform incurs a cost based on the set of items it serves. This cost is modeled using a convex function, which can capture various real-world phenomena such as increasing marginal cost due to operational overhead or resource constraints, or decreasing marginal cost up to a point, representing economies of scale until a critical mass is reached. Additionally, the items are divided into groups, and each platform also has convex cost functions associated with the number of items that it handles from each group. This allows us to encode preferences or constraints related to group fairness or diversity goals.

The significance of group fairness constraints that enforce upper and lower bounds (quotas) on the number of items from each group assigned to a platform has been widely emphasized in literature (Celis, Straszak, and Vishnoi 2018; Luss 1999; Devanur, Jain, and Kleinberg 2013; Costello et al. 2016; Segal-Halevi and Suksompong 2019; Kay, Matuszek, and Munson 2015; Bolukbasi et al. 2016; Panda, Louis, and Nimbhorkar 2024). For instance, in school choice, group fairness constraints can promote diversity among students assigned to each school based on attributes like ethnicity and socioeconomic background, as observed in practical implementations (Cowen Institute 2011). Similarly, in project teams, group fairness constraints ensure the inclusion of experts from all the relevant fields. The hospital-resident matching problem also benefits from group fairness constraints to ensure diverse specialties among assigned medical interns. These constraints thus achieve Restricted Dominance introduced in Bera et al. (2019), which asserts that the representation from any group on any platform does not exceed a user-specified cap, and Minority Protection (Bera et al. 2019), which asserts that the representation from any group, among the items matched to any platform is at least a user-specified bound. These principles restrict the over-representation or under-representation of any group on any platform by setting a threshold for each group’s representation, ensuring balanced allocation across groups on each platform. Fairness constraints with bounds on the number of items with each attribute are also studied in the context of ranking and multi-winner voting (Celis, Straszak, and Vishnoi 2018; Celis, Huang, and Vishnoi 2018).

*These authors contributed equally.

However, in literature, these constraints are given as fixed lower and upper bounds, which may not be desirable in many practical applications. For instance, the number and distribution of students applying to schools may change from year to year. Moreover, it may not be possible to meet the upper and lower bounds of all platforms simultaneously, thus making the instance infeasible. Flexible quotas are thus desirable over strict quotas in such scenarios. Our work departs from the strict quota systems by utilizing convex cost functions and introduces a softer version of the fairness conditions, where the under- and/or over-representation of groups is *penalized* in terms of costs, thus allowing flexibility while enforcing fairness indirectly through cost minimization.

In many practical applications, specifying bounds optimally in advance is challenging, particularly when total allocations depend on dynamic factors such as demand, participation, or resource availability. Consequently, relaxing lower bounds or augmenting upper bounds is often considered to allow allocations to deviate from strict constraints (A., Ravi, and Nasre 2023). This flexibility is crucial in scenarios where rigid adherence to bounds could exclude valid matches or lead to wastage. For instance, schools might relax regional quotas when there are insufficient applicants from a particular region, reallocating seats to other deserving candidates, thus dynamically increasing the quotas for those groups. The group-specific convex cost functions in our model ensure equitable representation among the groups in each platform through one of the following mechanisms:

- 1) **Over-representation Penalty (Soft Restricted Dominance):** When the cost function is convex and increasing, assigning too many items from a group to a platform increases the cost, thereby limiting the group’s representation.
- 2) **Under-representation penalty (Soft Minority Protection):** When the cost function is convex and decreasing, it discourages assigning too few items from any group by imposing decreasing costs as the number of assigned items increases below a threshold level of representation.
- 3) **Balanced representation:** Achieve both 1) and 2) by having a convex cost function that changes direction at an inflection point. This simultaneously penalizes both under- and over-representation, encouraging group sizes within a preferred range.

See the extended version for an example demonstrating how the fairness notions described above are softly enforced using convex cost functions.

Our model can also capture objectives like Nash Social Welfare (NSW) through appropriate cost functions. Nash Social Welfare (NSW) is a prominent welfare function in mathematical economics, which balances fairness and efficiency in resource distribution (Caragiannis et al. 2019). We formally show this connection in Proposition 2.6.

Convex cost functions are often used to model scenarios where the marginal cost of serving additional items increases, such as in systems with congestion, limited capacity, or group-specific handling requirements. Our problem frequently arises in applications such as cloud computing, where servers have group-specific service costs (e.g., handling different types of tasks or workloads), or logistics, where transportation hubs face rising costs when serving more goods

from certain categories. Our framework also naturally applies to resource-constrained Machine Learning deployment, where tasks or models must be assigned to servers with limited resources like compute or memory. The convex cost functions model the non-linear degradation or scaling penalties in overloaded systems. Group-specific costs enforce fairness across applications or user groups. The utility captures deployment performance goals such as responsiveness or latency for a model-server pair.

Our framework is also well suited to address data heterogeneity challenges in federated learning (FL), where client (item) participation is constrained by server (platform) bandwidth. Our convex cost framework allows for control over group-level participation, where groups may correspond to data distributions, for example, datasets in different languages or in different regions. The use of group-specific convex costs enables the system to penalize skewed participation over time and incentivize more balanced inclusion of underrepresented clients. This leads to more robust global models that generalize better across non-uniform client distributions. The utility could represent the improvement in model performance from previous rounds, updated after each training round, before assigning clients to server slots for the next round.

In this paper, our goal is to efficiently compute a many-to-one matching that keeps both group-specific and total platform costs to a minimum while ensuring that the total utility is above a given threshold.

Our Contributions: We present a framework for group fair resource allocation in bipartite matching where platforms incur convex costs based on both the total number of assigned items and the distribution of said items across groups. By replacing rigid group fairness constraints with convex penalties, our model enables flexible trade-offs between utility and fairness. Our main contributions are:

- **Modeling group fairness via convex costs:** We propose convex cost functions as a way to encode soft constraints on group representation. This also captures objectives like Nash Social Welfare (see Proposition 2.6).
- **Flow based algorithmic framework:** We reduce the group fair matching problem with convex costs to a Minimum Cost Flow problem under a total utility constraint when the groups are laminar.¹ We design a multi-layer flow network that embeds group-specific and platform-level convex costs using piecewise linear approximations.
- **Efficient approximation algorithm:** When the groups are disjoint or laminar, we develop an efficient Linear Programming based algorithm that produces an integral solution with provable runtime and approximation guarantees using the integrality of the flow polytope. This algorithm is exact when the utilities are uniform across all item-platform pairs.
- **Hardness result for general groups:** We show the NP-hardness of the problem for non-laminar group structures via a reduction from the independent set problem.

¹A family of sets, say S , is laminar if, for every pair of sets $X, Y \in S$, one of the following holds: $X \subseteq Y$ or $Y \subseteq X$ or $X \cap Y = \phi$.

2 Our Problem and Results

Convex Cost Matching with Utilities (CCMU): We are given a bipartite graph $G = (I, P, E)$ where I is the set of n items, P is the set of m platforms, and $E \subseteq I \times P$ is the set of edges representing the possible assignments of items to platforms. We use $\delta(v)$ to denote the set of all the edges incident on vertex v , $N(v)$ denotes the neighborhood of v , and $\Delta(v)$ denotes the degree of v . The items are divided into τ , possibly overlapping groups, I_1, I_2, \dots, I_τ where $I_j \subseteq I$ represents the set of items in group j , such that $I = I_1 \cup I_2 \cup \dots \cup I_\tau$ and $\Delta_j(p)$ denotes the number of items in $N(p)$ from group j i.e. $|N(p) \cap I_j|$, $j \in [\tau]$. Each $e \in E$ has an associated utility, denoted by $u_e \forall e \in E$. Each platform $p \in P$ has an associated convex cost function $f_p(\sigma_p)$ where σ_p denotes the total number of items assigned to p . It also has a convex cost function for each of the τ groups, $f_p^1(\nu_p^1), f_p^2(\nu_p^2), \dots, f_p^\tau(\nu_p^\tau)$ where ν_p^j is the number of items assigned to p from I_j . When the groups are disjoint, $\sigma_p = \sum_{j=1}^\tau \nu_p^j$ otherwise $\sigma_p \leq \sum_{j=1}^\tau \nu_p^j$. An assignment \mathcal{M} of items to platforms, referred to as a *matching* in this paper, is a subset of E such that each item is assigned to exactly one platform. Utility of \mathcal{M} is $\sum_{e \in \mathcal{M}} u_e$. Cost of \mathcal{M} is the sum of values of all the cost functions.

Let F be a $m \times (\tau + 1)$ matrix with F_{pj} denoting the convex cost function associated with platform $p \in P$ and group I_j , $f_p^j(\cdot)$, for $j \in [\tau]$, and $F_{p(\tau+1)}$ denoting the convex cost function $f_p(\cdot)$. As a part of input, we are also given a lower bound, ℓ , on the total utility of a matching. We denote the input instance as $\mathcal{I} = (G, I_1, I_2, \dots, I_\tau, \ell, \vec{u}, F)$, where \vec{u} is a $|E|$ -sized edge utility vector. The goal is to compute a matching with total utility at least ℓ while keeping the total cost to a minimum. Let x_e be a variable associated with each edge, $e \in E$, such that for an edge (i, p) :

$$x_{ip} = \begin{cases} 1 & \text{if item } i \in I \text{ is assigned to platform } p \in P \\ 0 & \text{otherwise} \end{cases}$$

Therefore, $\nu_p^j = \sum_{i \in N(p)} x_{ip}$, $\sigma_p = \sum_{i \in N(p)} x_{ip}$ and the total cost of the platforms, say, $\mathcal{C}_{\mathcal{I}}$, is

$$\mathcal{C}_{\mathcal{I}} = \sum_{p \in P} \left(\sum_{j=1}^{\tau} f_p^j(\nu_p^j) + f_p(\sigma_p) \right) \quad (1)$$

Hence, the objective is to minimize $\mathcal{C}_{\mathcal{I}}$ (Equation (1)) while satisfying the utility constraint. The following Integer Linear Program captures this problem.

ILP 2.1.

$$\min \mathcal{C}_{\mathcal{I}} \quad (2)$$

$$\text{such that } \sum_{e \in E} u_e x_e \geq \ell \quad (3)$$

$$\sum_{e \in \delta(i)} x_e \leq 1 \quad \forall i \in I \quad (4)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (5)$$

Results

Let OPT represent the minimum cost of a matching that satisfies the utility constraint, and $\nabla f(x)$ denote $f(x) - f(x - 1)$ for any function $f(\cdot)$. Our main contribution is a polynomial time algorithm that computes a matching with utility at least ℓ and the total cost close to OPT , when the groups are either disjoint or laminar, formally stated below.

Theorem 2.2. *Given an instance $\mathcal{I} = (G, I_1, I_2, \dots, I_\tau, \ell, \vec{u}, F)$ of the CCMU problem with disjoint groups, there is a polynomial-time algorithm that computes a matching on G such that the total utility is at least ℓ (Constraint 3) and the total cost of the platforms, $\mathcal{C}_{\mathcal{I}}$ (Equation (1)), is at most $OPT + \max_p [\nabla f_p(\sigma_p) + \max_j \nabla f_p^j(\nu_p^j)] - \min_p [\nabla f_p(\sigma_p) + \min_j \nabla f_p^j(\nu_p^j)]$.*

Thus our algorithm gives an additive approximation to the optimal cost, where the extra cost incurred is at most the cost of assigning one extra item to one platform. We give an analogous result, stated below, when the groups are overlapping but follow a laminar structure. Let the maximum number of groups an item belongs to be d , referred to as the *depth* of the laminar structure. Thus, the groups form a depth- d forest (extended version, Figure 2). Let j_r denote the group, I_j , $j \in [\tau]$ at level $r \in [d]$ in this forest. For laminar groups, we formally state our result below.

Theorem 2.3. *Given an instance $\mathcal{I} = (G, I_1, I_2, \dots, I_\tau, \ell, \vec{u}, F)$ of the CCMU problem, when the groups follow a laminar structure with depth d , there is a polynomial-time algorithm that computes a matching on G such that the total utility is at least ℓ and the total cost $\mathcal{C}_{\mathcal{I}}$ (Equation (1)) is at most $OPT + \max_p [\nabla f_p(\sigma_p) + \sum_{r=1}^d \max_{j_r} \nabla f_p^{j_r}(\nu_p^{j_r})] - \min_p [\nabla f_p(\sigma_p) + \min_j \nabla f_p^j(\nu_p^j)]$.*

Hardness: We give an NP-hardness result for the CCMU problem when groups have a non-laminar structure using a reduction from the independent set problem inspired by a similar reduction in (Sankar et al. 2021).

Theorem 2.4. *The CCMU problem is NP-hard when groups have a non-laminar structure. In fact, it cannot be approximated to any positive multiplicative factor.*

We also contribute an exact algorithm for a special case of the CCMU problem that has unit utilities that is $u_e = 1, \forall e \in E$, and $\ell \leq |I|$, when the groups are either disjoint or laminar. We formally state this result below.

Theorem 2.5. *Given an instance $\mathcal{I} = (G, I_1, I_2, \dots, I_\tau, \ell, 1, F)$ of the CCMU problem, when the groups are disjoint or laminar, and $\ell \leq |I|$, there is a polynomial-time algorithm that computes a matching on G such that the total utility is at least ℓ and the total cost of the platforms, $\mathcal{C}_{\mathcal{I}}$ (Equation (1)), is minimized.*

We sketch the proof of Theorem 2.2 in Section 4, and the detailed proofs of Theorems 2.2, 2.3, 2.4 and 2.5 can be found in the extended version in sections 4, 5, and 6, respectively. In fact, we prove a more general version of Theorem 2.5 where all the utilities have the same integer value, not necessarily unit. We note that the group structures considered here arise naturally e.g. when groups are based on one attribute like

age (disjoint) or on geographical boundaries like city, state, country (laminar).

Connection to Nash Social Welfare: We establish the connection between our model and Nash Social Welfare with the following proposition:

Proposition 2.6. Let ν_p^j and σ_p respectively denote the number of items from the group I_j , and total number of items matched to platform p . Let τ_p be the number of groups in platform p 's neighborhood. Then, the following holds:

1) Maximizing the Nash Social Welfare (NSW) across groups on platform p corresponds to maximizing $(\prod_j \nu_p^j)^{\frac{1}{\tau_p}}$. Its convex relaxation is the well-studied Eisenberg-Gale program, (Cole and Gkatzelis 2018) which maximizes $\frac{1}{\tau_p} \sum_{j=1}^{\tau_p} \log(\nu_p^j)$.

2) Maximizing NSW for fair load distribution across platforms corresponds to maximizing $(\prod_p \sigma_p)^{\frac{1}{m}}$ whose convex relaxation is to maximize $\frac{1}{m} \sum_{p=1}^m \log(\sigma_p)$.

Therefore, the overall relaxed objective becomes to maximize $\sum_{p=1}^m \left(\sum_{j=1}^{\tau} \frac{\log(\nu_p^j)}{\tau_p} + \frac{\log(\sigma_p)}{m} \right)$ or equivalently to minimize

$$\sum_{p=1}^m \left(\sum_{j=1}^{\tau} \frac{-\log(\nu_p^j)}{\tau_p} + \frac{-\log(\sigma_p)}{m} \right).$$

This expression matches the objective in Equation (1) when the platform cost function is set to $\frac{-\log(\cdot)}{m}$, and every group-specific convex cost function for any platform, p , is set to $\frac{-\log(\cdot)}{\tau_p}$. That is $f_p(\cdot) = \frac{-\log(\cdot)}{m}$, $\forall p \in P$ and $f_p^j(\cdot) = \frac{-\log(\cdot)}{\tau_p}$, $\forall p \in P, j \in [\tau]$ in Equation (1).

3 Related Work

Allocation problems are foundational in operations research and find applications in diverse fields such as resource allocation (Halabian, Lambadaris, and Lung 2012), kidney exchange programs (Farnadi et al. 2021), school choice (Abdulkadiroglu and Sönmez 2003), candidate selection (Bei et al. 2020), summer internship programs (Aziz, Baychkov, and Biró 2020), and matching residents to hospitals (Goko et al. 2022). A comprehensive survey of developments in matching with constraints, including those based on regions, diversity, multi-dimensional capacities, and matroids, is provided in Aziz, Biró, and Yokoo (2022).

Fairness constraints are integral to ensuring equitable outcomes in matching problems, and are captured in various forms like justified envy-freeness (Abdulkadiroglu and Sönmez 2003), proportional matching (Bei et al. 2020), and upper and lower bounds on the total representation of each group (Huang 2010; Gonczarowski et al. 2019). Fairness constraints, with varied notions of fairness, have also been considered in stability under matroid constraints (Fleiner and Kamiyama 2016), in various settings of two-sided matching markets (Beyhaghi and Éva Tardos 2021), (Patro et al. 2020), (Huang et al. 2016), recommender systems (Chen et al. 2024; Greenwood, Chiniah, and Garg 2024), and fair assortment planning (Chen, Golrezaei, and Susan 2025; Lu, Sahin, and

Wang 2023). In contexts addressing historical discrimination, vertical reservations (implemented as set-asides) and horizontal reservations (implemented as minimum guarantees or lower bounds) are prominent mechanisms used in India to safeguard disadvantaged groups, as detailed in Sönmez and Yenmez (2022).

When groups are overlapping, Sankar et al. (2021) present a polynomial-time algorithm with an approximation ratio of $\frac{1}{\Delta+1}$ where each item belongs to at most Δ laminar families of groups per platform, and Nasre, Nimbhorkar, and Pulath (2019) show the NP-hardness of the problem without a laminar structure. Both papers focus only on upper bounds, and Sankar et al. (2021) show that their problem is NP-hard to approximate within a factor of $\mathcal{O}\left(\frac{\log^2 \Delta}{\Delta}\right)$. We show that this hardness result holds for our problem as well when the groups overlap arbitrarily (see Section 2 for details). Louis et al. (2023) address proportional and diversity constraints, and Panda, Louis, and Nimbhorkar (2024) study individual fairness along with group fairness constraints for disjoint groups. All these works focus on satisfying the constraints while maximizing the matching size. In contrast, our work looks at matching in a way that the utility is above a certain threshold while minimizing the total cost imposed using convex cost functions.

Traditionally, bounds or quotas, used to enforce fairness constraints, are fixed. However, recent studies have explored capacity expansion in applications such as school choice (Bobbio et al. 2023), matching residents to hospitals (Abe, Komiyama, and Iwasaki 2022), and college admissions (Bobbio et al. 2021). Ranjan, Nasre, and Nimbhorkar (2024) looks at optimally increasing the quotas of hospitals to ensure that a strongly stable matching exists in the hospital resident matching. They show that minimizing the maximum capacity increase for any hospital is NP-hard in general and provide a polynomial-time algorithm for minimizing the total capacity increase across all hospitals. However, most of these problems become NP-hard when each hospital incurs a cost for each capacity increase, even if the costs are 0 or 1.

Various papers like Ehlers et al. (2014); Echenique and Yenmez (2015); Aziz, Gaspers, and Sun (2020); Kurata et al. (2017) consider soft diversity constraints in the controlled school choice problem, and Kurata et al. (2017) consider the setting in which each student has multiple types. For more details, we refer the reader to the survey paper Aziz, Biró, and Yokoo (2022). A., Ravi, and Nasre (2023) look at matching applicants to posts with one-sided preferences where the assignments can deviate from the range defined by the quotas.

In A., Sankar, and Nasre (2022) and Limaye and Nasre (2023), cost-controlled quotas replace fixed quotas in one-sided and two-sided preference-based matching, respectively. However, the problems considered in both these works are very different from ours. While these works also minimize costs, their notions of optimality (envy-freeness, rank-maximality) are preference-based and more aligned with individual fairness, rather than group fairness. Additionally, these models do not consider group structures, and matching costs are linear rather than convex. In contrast, we address

Algorithm 1: Min Cost ℓ -util Flow($\mathcal{I} = (G, I_1, I_2, \dots, I_\tau, F)$)

Input : \mathcal{I}

Output : A matching, \mathcal{M} , with utility at least ℓ

- 1 $\mathcal{M} = \emptyset$
 - 2 Follow the steps in Section 12 to construct the flow network Γ
 - 3 Solve LP 4.2 on the network, Γ , and store the result in x^*
 - 4 **if** x^* is fractional **then**
 - 5 $FL = \text{Rounding}(x^*)$ [extended version, Alg 2]
 - 6 **else**
 - 7 $FL = x^*$
 - 8 $FE = \{(i, v) : i \in I \text{ and the edge } (i, v) \text{ is part of the flow, } FL\}$
 - 9 **for** $(i, v) \in FE$ **do**
 - 10 Let p be the platform such that v is a group layer copy of p .
 - 11 $\mathcal{M} = \mathcal{M} \cup (i, p)$
 - 12 **Return** \mathcal{M}
-

group fairness in a cost-minimization framework by employing convex cost functions to enforce soft diversity constraints. Although fairness also appears in the objective in regularization based fairness in machine learning (Kamishima, Akaho, and Sakuma 2011), our formulation is fundamentally distinct.

4 Cost-Approximate Matching Algorithm for Disjoint Groups

We prove Theorem 2.2 in this section. We first construct a minimum-cost flow instance from the given input instance (Step 2 of Algorithm 1, see Figure 1) and then use LP 4.2 to compute a minimum-cost flow (step 3). If the solution is fractional, we invoke Algorithm 2 (extended version) to round it to an integral flow. Finally, we map this flow to a bipartite matching that satisfies the utility constraint while possibly incurring a slightly higher cost than OPT . We show in Lemma 4.5 that if an optimal solution of LP 4.2 is fractional, it consists of at most two fractional paths. Algorithm 2 resolves this by rounding up the edges in the higher-utility path and rounding down those in the lower-utility path. We now delve into the details of our algorithm, starting with the network construction.

Flow Network Construction

The network is a multi-layer directed bipartite multigraph (see Figure 1), denoted by $\Gamma = (V_\Gamma, E_\Gamma)$, constructed from the input graph, G , described below:

1. *Source and sink layer:* Create new nodes s and t .
2. *Item layer:* One node i for each $i \in I$
3. *Group layer:* One node p^j for each platform $p \in P$, each group $j \in g(p)$ pair, where $g(p)$ denote the set of groups with at least one item in the neighborhood of platform p , $N(p)$.

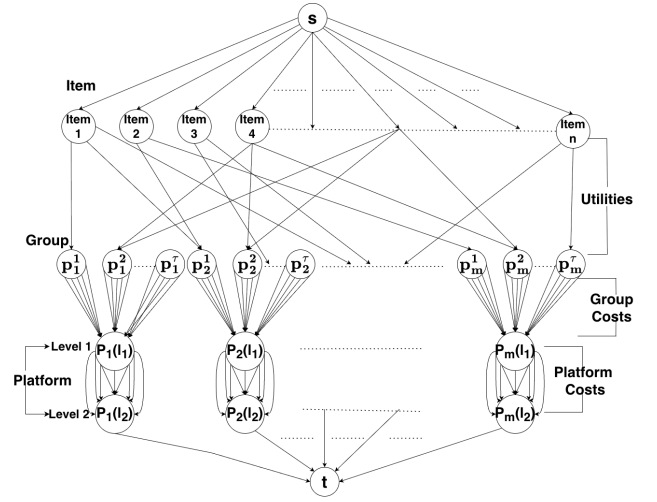


Figure 1: Flow network when the input graph G has n items, m platforms and τ groups with the following edges (Item 1, p_1), (Item 1, p_2), (Item 2, p_2), (Item 4, p_1), (Item 4, p_2), (Item 4, p_m), (Item n , p_m). In G , Items 1 and 2 are in group 1, Item 4 is in group 2 and Item n is in group τ

4. *Platform layer:* This consists of two levels, each containing a copy of every platform denoted by $p(l_1)$ and $p(l_2)$ in the first and second levels, respectively, for each $p \in P$.

1. Edge (s, i) for each $i \in I$, with $w_{si} = 0$ and $c_{si} = 1$,
2. Edges (i, p^j) for each $j \in [\tau]$, $i \in I_j$, $p \in N(i)$, with $c_{ip^j} = 1$ and $w_{ip^j} = 0$. When the groups are disjoint, it is straightforward to verify that for every edge $(i, p) \in E$, there exists exactly one corresponding $(i, p^j) \in E_\Gamma$. We will later show that this one-to-one correspondence also holds in the case of laminar groups, as discussed in the network construction for that setting (extended version, section 5).
3. L_1 : Multi-edges $(p^j, p(l_1))^k$ for $k \in [\Delta_j(p)]$ for each $p \in P$, $j \in g(p)$, with capacity 1 and weight $f_p^j(k) - f_p^j(k-1)$. Thus, these are parallel edges with the same capacity but different weights. $(p^j, p(l_1))^k$ is an edge with capacity 1 and weight $f_p^j(k) - f_p^j(k-1) \forall k \in [\Delta_j(p)]$.
4. L_2 : Multi-edges $(p(l_1), p(l_2))^k \forall p \in P$, $k \in [\Delta(p)]$. An edge $(p(l_1), p(l_2))^k$ has capacity 1 and weight $f_p(k) - f_p(k-1)$.
5. Edges (p, t) with $w_{pt} = 0$ and $c_{pt} = \Delta(p)$, $\forall p \in P$.

Additionally, the edges (i, p^j) have a utility associated with them, $u_{ip^j} = u_{ip} \forall (i, p^j) \in E_\Gamma, (i, p) \in E$. Other edges have 0 utility.

We first state the following correspondence between matchings in G and flows in Γ :

Lemma 4.1. *Let $\hat{\ell} \in \mathbb{R}_{\geq 0}$. There exists a matching in G with utility at least $\hat{\ell}$ and minimum cost $\hat{C}_\mathcal{I}$ iff there exists a flow in Γ with utility at least $\hat{\ell}$ and cost $\hat{C}_\mathcal{I}$. Such a matching is output by Steps 9–11 of Algorithm 1.*

The formal proof of Lemma 4.1 is present in the extended version (Lemma 4.1), we provide a sketch of the proof here.

Proof Sketch: Assuming disjoint groups, we establish a one-to-one correspondence between matchings in the bipartite graph, G and flows in the constructed network Γ . Each edge $(i, p) \in E$, where item i belongs to group, j , maps uniquely to an edge from i to the corresponding group-platform copy, $p^j \in V_\Gamma$, preserving utility. Hence, any matching in G with utility at least $\hat{\ell}$ corresponds directly to a flow of the same utility in Γ , and vice versa. We then show that the minimum cost of such a matching equals the minimum cost of the corresponding flow by accounting for the cost bearing edges in Γ to compute the total flow cost which exactly matches the cost expression of our objective in Equation (1).

By Lemma 4.1, it suffices to compute a flow on the constructed network, Γ , that minimizes the total flow cost while ensuring that the flow utility is at least ℓ (Equation (3)). The Integer Programming version of the Linear Program (LP) defined in LP 4.2 over Γ does exactly this. For any vertex $v \in V_\Gamma$, let $\delta^-(v)$ and $\delta^+(v)$ denote the sets of incoming and outgoing edges, respectively. It is easy to verify that the constraints 6, 8, 9 and 10 of LP 4.2 describe the flow polytope corresponding to Γ . We denote this flow polytope by Q and refer the reader to standard references such as (Ahuja, Magnanti, and Orlin 1993) for background.

LP 4.2.

$$\min \sum_{e \in E_\Gamma} w_e x_e \quad (6)$$

$$\text{such that } \sum_{e \in \delta^+(i)} x_e \leq 1 \quad \forall i \in I \quad (7)$$

$$\sum_{e \in E_\Gamma} u_e x_e \geq \ell \quad (8)$$

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e \quad \forall v \in V \setminus \{s, t\} \quad (9)$$

$$x_e \geq 0 \quad \forall e \in E_\Gamma \quad (10)$$

The constraints 6, 9 and 10 of LP 4.2 describe the flow polytope, denoted by, say Q , corresponding to Γ . More background can be found in (Ahuja, Magnanti, and Orlin 1993).

Since LP 4.2 is a linear programming relaxation, its optimal solution may be fractional. Consequently, the resulting flow may not correspond to a valid matching in the original graph. To address this, we apply a simple rounding procedure described in Algorithm 2 (extended version), which converts the fractional flow into an integral one, that incurs only a bounded increase in cost, and hence corresponds to a valid matching that satisfies the utility constraint with the same increased cost. This is possible because we show that any optimal solution to LP 4.2 contains at most two fractional paths. By rounding up the path with higher utility and rounding down the one with lower utility (if it exists), we obtain an integral solution that maintains feasibility and closely preserves the cost. Before we formally state these in Lemma 4.5 and Lemma 4.6 we first observe the following:

Observation 4.3. An optimal solution to LP 4.2 has utility exactly ℓ .

This follows because the costs are non-negative, and hence any solution with higher utility can be scaled down to another solution with utility exactly ℓ and smaller cost. The correctness of the rounding procedure relies on Lemmas 4.5 and 4.6, both of which are derived from the following result, formally proved in the extended version (Section 4.2):

Lemma 4.4. *An optimal (fractional) vertex solution² x^* of LP 4.2 is a convex combination of two integer solutions of the flow polytope on Γ , denoted by Q . Moreover, one of the two solutions has both cost and utility larger than those of x^* and the other one has both cost and utility smaller than those of x^* .*

Proof Sketch: Since the flow polytope Q is integral (see (Ahuja, Magnanti, and Orlin 1993)), every vertex of Q corresponds to an integral flow. By Observation 4.3, the optimal solution x^* to LP 4.2 lies on the hyperplane defined by constraint 8. Because LP 4.2 adds exactly one constraint to the description of Q , x^* must lie on an edge of Q , whose endpoints are two integral vertices x_1^* and x_2^* . Then x^* is a convex combination of x_1^* and x_2^* , and since it lies on the utility threshold hyperplane, x_1^* and x_2^* must lie on opposite sides of it. One has utility above ℓ , and the other below. If the lower-utility point also has higher cost, then x^* cannot be optimal, as shifting the combination toward the higher-utility, lower-cost point would improve the objective. This contradiction establishes the claim.

Following are the main lemmas in analyzing the approximation factor and their combined proof sketch.

Lemma 4.5. *If an optimal solution x^* of LP 4.2 has fractional edges, it has at most two fractional paths.*

Lemma 4.6. *Algorithm 1 returns a matching with total utility at least ℓ and cost at most $OPT + \max_p [\nabla f_p(\sigma_p) + \max_j \nabla f_p^j(\nu_p^j)] - \min_p [\nabla f_p(\sigma_p) + \min_j \nabla f_p^j(\nu_p^j)]$ when the groups are disjoint.*

Proof Sketch: We build on the framework of (Gallo and Sodini 1978), which characterizes the structure of extreme points of flow polytopes. We begin by recalling some definitions from (Gallo and Sodini 1978). A cycle in the flow network Γ is a sequence of nodes (v_1, \dots, v_r) such that for each k , either $(v_k, v_{k+1}) \in E_\Gamma$ or its reverse appears, with $v_1 = v_r$. $A(\gamma)$ denotes the set of arcs in a cycle, which is further partitioned into forward and reverse arcs based on direction. For a flow x , the set of *floating arcs* is denoted by $A_1(x) = \{e \in E_\Gamma : 0 < x_e < 1\}$. By Theorem 3.2 of (Gallo and Sodini 1978), if a cycle γ is the only cycle in $A(\gamma) \cup A_1(x)$, then the flow can be perturbed along γ to obtain an adjacent extreme point. Using properties of the flow network and the flow polytope, they show that, any pair of adjacent extreme points differs by at most one cycle.

Let x^* be the optimal (fractional) solution to LP 4.2. By Lemma 4.4, we have $x^* = \lambda x_0^* + (1 - \lambda)x_1^*$ where x_0^*, x_1^* are

²An extreme point is a point in the polytope that cannot be expressed as a convex combination of two distinct points in the polytope. Equivalently, it is also a vertex of the polytope where as many linearly independent constraints as the polytope's dimension attain equality. See any reference on linear programming for details.

Utility Threshold	Cost				Run-time (s)	
	Naive Greedy	Greedy	LP 4.2	Algorithm 1	Greedy	Algorithm 1
500	3384	1816	1788	1802	0.757	1.545
1000	11126	6810	6768	6794	0.77	1.306
1500	23992	15224	15100	15100	1.111	1.331

Table 1: Top 10 most-watched movies (Size = 4863)

Utility Threshold	Cost				Run-time (s)	
	Naive Greedy	Greedy	LP 4.2	Algorithm 1	Greedy	Algorithm 1
500	1698	1076	1072	1088	0.736	2.169
1000	5780	3698	3684	3704	1.698	3.163
1500	11834	8030	7965	7986	1.902	2.335

Table 2: Top 20 most-watched movies (Size = 8716)

extreme points of Q , and are integral. By the above result, x_0^* and x_1^* differ on a single cycle, so x^* has fractional flow only along that cycle, that is on at most two paths in Γ . Therefore, we round up the path with higher utility and round down the other, preserving feasibility. The increase in cost is bounded by the difference between the maximum and minimum total cost across all such two-edge paths, leading to the claimed additive bound in Lemma 4.6.

Lemma 4.6 establishes that Algorithm 1 returns a matching with total utility at least ℓ and cost at most $OPT + \max_p [\nabla f_p(\sigma_p) + \max_j \nabla f_p^j(\nu_p^j)] - \min_p [\nabla f_p(\sigma_p) + \min_j \nabla f_p^j(\nu_p^j)]$. To complete the proof of Theorem 2.2 it remains to show that Algorithm 1 runs in polynomial time. The total number of variables in LP 4.2 is $|E_\Gamma|$ and the total number of constraints is $n + |V_\Gamma| + |E_\Gamma| - 1$. Therefore, LP 4.2 can be solved in time polynomial in the number of vertices and edges in the constructed network, Γ . The runtime for both the rounding step and the loop from step 9 to 11 in Algorithm 1 is $\mathcal{O}(|E_\Gamma|)$. Hence, if we can show that E_Γ and V_Γ are polynomial in the number of items, platforms, and edges in the input graph, G , we are done. We show this in Lemma 4.7 stated below.

Lemma 4.7. *The flow network Γ has at most $n + 2m + |E| + 2$ nodes and exactly $n + m + 3|E|$ edges.*

We prove Lemma 4.7 in Section 4.2 of the extended version by accounting for all the nodes and edges in Γ (Figure 1).

5 Experiments

This section evaluates Algorithm 1 on a real-world dataset, the movielens 100k dataset comprising 100,000 user ratings. We assess the practical performance and efficiency of our algorithm and compare it against a natural greedy baseline. Our evaluation focuses on the cost-approximation quality of the integral solution returned by our algorithm, relative to both the greedy output and the optimal value of LP 4.2. Empirically, Algorithm 1 consistently outperforms the greedy algorithm and achieves a cost that is only marginally higher than the Linear Programming (LP) optimum. Since the LP value serves as a lower bound on the true optimum (OPT), this suggests that our algorithm is near-optimal in practice.

Dataset

The movielens 100k dataset (Harper and Konstan 2015) consists of 100,000 movie ratings from users. Please refer to section 8.1 in the extended version for more details on this dataset.

In our model, users correspond to items and movies to platforms. Each user rating is treated as the utility of the edge between the user (item) and the movie (platform) they rated. We partition users into groups based on age brackets: 15 – 29 (group 1), 30 – 44 (group 2), 45 – 59 (group 3), 60 – 74 (group 4) and < 15 (group 5). We use the number of ratings as a proxy for the number of views for each movie.

Experimental Setup and Results

We implement our algorithm in Python 3.10, and all the experiments are run using Google Colab notebook on a virtual machine with Intel(R) Xeon(R) CPU @ 2.20GHz and 13GB RAM. To encourage age group diversity within each movie screening and equitable user distribution across screenings, we employ convex cost functions, specifically, x^2 to evaluate Algorithm 1 on the top 10, 20, 50, 75, and 100 most-watched movies.

In tables 1, 2, 3, 4, and 5 (last three in section 8 of the extended version), we report a comparison of the total cost (rounded to the nearest integer) incurred by our algorithm with that of the optimal solution to LP 4.2, the greedy algorithm (Algorithm 3 in the extended version), and a naive greedy baseline. The convex cost function used for every platform and every group-platform pair is x^2 . These comparisons are conducted under different utility threshold values: 500, 1000, and 1500. Additionally, we report the execution times of Algorithm 3 (extended version) and Algorithm 1. The value labeled ‘Size’ in each table caption refers to the total number of edges in the input bipartite graph.

The naive greedy, which selects the highest utility edge iteratively until the utility threshold is met, performs the worst in terms of cost, as expected. A more meaningful comparison is between the greedy algorithm and Algorithm 1. The greedy algorithm selects edges based on the highest utility-to-cost ratio (see Algorithm 2 in the extended version for pseudocode). As anticipated, the run time of Algorithm 2 is lower than that of Algorithm 1, since our algorithm requires constructing a flow network and solving an LP. However, our algorithm consistently achieves lower total cost across all runs, except for the first two utility thresholds in Table 2.

In figures 4,5,6, and 7 in the extended version, we observe an almost uniform distribution of items across platforms when all the cost functions are set to $-\log(x)$ for the top 10, 20, 50, and 75 movies datasets respectively. This is expected, as the resulting objective resembles a Nash Social Welfare kind of objective discussed in Proposition 2.6. We also plot the group-wise distribution of matched items for each platform in the extended version (Figures 10-15) and observe that they closely reflect the group proportions in the neighborhoods of the corresponding platforms in the input bipartite graph.

Acknowledgements

AP was supported by the Walmart Center for Tech Excellence at IISc. AL was supported in part by the Walmart Center for Tech Excellence at IISc, and SERB award CRG/2023/002896.

References

- A., S. K.; Ravi, R. R.; and Nasre, M. 2023. Matchings under One-Sided Preferences with Soft Quotas. In *IJCAI, Macao, SAR, China*.
- A., S. K.; Sankar, G. S.; and Nasre, M. 2022. Optimal Matchings with One-Sided Preferences: Fixed and Cost-Based Quotas. In *AAMAS, Auckland, New Zealand*.
- Abdulkadiroglu, A.; and Sönmez, T. 2003. School Choice: A Mechanism Design Approach. *The American Economic Review*.
- Abe, K.; Komiyama, J.; and Iwasaki, A. 2022. Anytime Capacity Expansion in Medical Residency Match by Monte Carlo Tree Search. In *IJCAI*.
- Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc. ISBN 013617549X.
- Aziz, H.; Baychkov, A.; and Biró, P. 2020. Summer Internship Matching with Funding Constraints. In *AAMAS*.
- Aziz, H.; Biró, P.; and Yokoo, M. 2022. Matching Market Design with Constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Aziz, H.; Gaspers, S.; and Sun, Z. 2020. Mechanism Design for School Choice with Soft Diversity Constraints. In *IJCAI*.
- Bei, X.; Liu, S.; Poon, C. K.; and Wang, H. 2020. Candidate Selections with Proportional Fairness Constraints. In *AAMAS*.
- Bera, S. K.; Chakrabarty, D.; Flores, N.; and Negahbani, M. 2019. Fair Algorithms for Clustering. In *NeurIPS*.
- Beyhaghi, H.; and Éva Tardos. 2021. Randomness and Fairness in Two-Sided Matching with Limited Interviews. In *12th Innovations in TCS Conf. (ITCS 2021)*.
- Bobbio, F.; Carvalho, M.; Lodi, A.; Rios, I.; and Torrico, A. 2023. Capacity Planning in Stable Matching: An Application to School Choice. In *EC*, 295. ACM.
- Bobbio, F.; Carvalho, M.; Lodi, A.; and Torrico, A. 2021. Capacity Expansion in the College Admission Problem. *CoRR*, abs/2110.00734.
- Bolukbasi, T.; Chang, K.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 4349–4357.
- Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Economics and Comput.*, 7(3): 12:1–12:32.
- Celis, L. E.; Huang, L.; and Vishnoi, N. K. 2018. Multiwinner Voting with Fairness Constraints. In Lang, J., ed., *IJCAI*.
- Celis, L. E.; Straszak, D.; and Vishnoi, N. K. 2018. Ranking with Fairness Constraints. In *ICALP*.
- Chen, Q.; Golrezaei, N.; and Susan, F. 2025. Fair Assortment Planning. arXiv:2208.07341.
- Chen, Q.; Liang, J. C. N.; Golrezaei, N.; and Bouneffouf, D. 2024. Interpolating Item and User Fairness in Multi-Sided Recommendations. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Cole, R.; and Gkatzelis, V. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM J. Comput.*, 47(3): 1211–1236.
- Costello, M.; Hawdon, J.; Ratliff, T.; and Grantham, T. 2016. Who Views Online Extremism? Individual Attributes Leading to Exposure. *Comput. Hum. Behav.*
- Cowen Institute. 2011. Case Studies of School Choice and Open Enrollment in Four Cities. Technical report, Cowen Institute.
- Devanur, N. R.; Jain, K.; and Kleinberg, R. D. 2013. Randomized Primal-dual Analysis of RANKING for Online Bipartite Matching. In *SODA '13*.
- Echenique, F.; and Yenmez, M. B. 2015. How to Control Controlled School Choice. *American Economic Review*.
- Ehlers, L.; Hafalir, I. E.; Yenmez, M. B.; and Yildirim, M. A. 2014. School choice with controlled choice constraints: Hard bounds versus soft bounds. *Journal of Economic Theory*, 153: 648–683.
- Farnadi, G.; St-Arnaud, W.; Babaki, B.; and Carvalho, M. 2021. Individual Fairness in Kidney Exchange Programs. *AAAI*, 35(13): 11496–11505.
- Fleiner, T.; and Kamiyama, N. 2016. A Matroid Approach to Stable Matchings with Lower Quotas. *Math. Oper. Res.*, 41(2): 734–744.
- Gallo, G.; and Sodini, C. 1978. Extreme points and adjacency relationship in the flow polytope. *Calcolo*, 15: 277–288.
- Goko, H.; Makino, K.; Miyazaki, S.; and Yokoi, Y. 2022. Maximally Satisfying Lower Quotas in the Hospitals/Residents Problem with Ties. In *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPICs*, 31:1–31:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Gonczarowski, Y. A.; Nisan, N.; Kovalio, L.; and Romm, A. 2019. Matching for the Israeli: Handling Rich Diversity Requirements. In Karlin, A.; Immorlica, N.; and Johari, R., eds., *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, 321. ACM.
- Greenwood, S.; Chiniyah, S.; and Garg, N. 2024. User-item fairness tradeoffs in recommendations. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak,

- J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Halabian, H.; Lambadaris, I.; and Lung, C.-H. 2012. Optimal server assignment in multi-server parallel queueing systems with random connectivities and random service failures. In *IEEE International Conference on Communications (ICC)*.
- Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4).
- Huang, C. 2010. Classified Stable Matching. In Charikar, M., ed., *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, 1235–1253. SIAM.
- Huang, C.; Kavitha, T.; Mehlhorn, K.; and Michail, D. 2016. Fair Matchings and Related Problems. *Algorithmica*, 74(3): 1184–1203.
- Kamishima, T.; Akaho, S.; and Sakuma, J. 2011. Fairness-aware Learning through Regularization Approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, 643–650.
- Kay, M.; Matuszek, C.; and Munson, S. A. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for Occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*.
- Kurata, R.; Hamada, N.; Iwasaki, A.; and Yokoo, M. 2017. Controlled School Choice with Soft Bounds and Overlapping Types. *J. Artif. Intell. Res.*
- Limaye, G.; and Nasre, M. 2023. Optimal Cost-Based Allocations Under Two-Sided Preferences. In *IWOCA*.
- Louis, A.; Nasre, M.; Nimbhorkar, P.; and Sankar, G. S. 2023. Online Algorithms for Matchings with Proportional Fairness Constraints and Diversity Constraints. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *ECAI 2023 - 26th European Conference on Artificial Intelligence*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 1601–1608.
- Lu, W.; Sahin, O.; and Wang, R. 2023. A simple way towards fair assortment planning: Algorithms and welfare implications. *Available at SSRN 4514495*.
- Luss, H. 1999. On Equitable Resource Allocation Problems: A Lexicographic Minimax Approach. *Operations Research*.
- Nasre, M.; Nimbhorkar, P.; and Pulath, N. 2019. Classified Rank-Maximal Matchings and Popular Matchings – Algorithms and Hardness. In Sau, I.; and Thilikos, D. M., eds., *Graph-Theoretic Concepts in Computer Science*, 244–257. Cham: Springer International Publishing. ISBN 978-3-030-30786-8.
- Panda, A.; Louis, A.; and Nimbhorkar, P. 2024. Individual Fairness under Group Fairness Constraints in Bipartite Matching - One Framework to Approximate Them All. In *IJCAI-243*.
- Patro, G. K.; Chakraborty, A.; Ganguly, N.; and Gummadi, K. P. 2020. Fair Updates in Two-Sided Market Platforms: On Incrementally Updating Recommendations. In *AAAI 2020, IAAI 2020, EAAI 2020*.
- Ranjan, K.; Nasre, M.; and Nimbhorkar, P. 2024. Optimal Capacity Modification for Strongly Stable Matchings. *CoRR*.
- Sankar, G. S.; Louis, A.; Nasre, M.; and Nimbhorkar, P. 2021. Matchings with Group Fairness Constraints: Online and Offline Algorithms. In *IJCAI*.
- Segal-Halevi, E.; and Suksompong, W. 2019. Democratic fair allocation of indivisible goods. *Artificial Intelligence*.
- Sönmez, T.; and Yenmez, M. B. 2022. Affirmative Action in India via Vertical, Horizontal, and Overlapping Reservations. *Econometrica*, 90(3): 1143–1176.