

# Matchings with Group Fairness Constraints: Online and Offline Algorithms

Govind S. Sankar<sup>1</sup>, Anand Louis<sup>\*2</sup>, Meghana Nasre<sup>\*1</sup> and Prajakta Nimbhorkar<sup>\*3,4</sup>

<sup>1</sup>Indian Institute of Technology Madras, Chennai

<sup>2</sup>Indian Institute of Science, Bangalore

<sup>3</sup>Chennai Mathematical Institute, Chennai

<sup>4</sup>UMI ReLaX

govindbose@gmail.com, anandl@iisc.ac.in, meghana@cse.iitm.ac.in, prajakta@cmi.ac.in

## Abstract

We consider the problem of assigning items to platforms in the presence of group fairness constraints. In the input, each item belongs to certain categories, called *classes* in this paper. Each platform specifies the group fairness constraints through an upper bound on the number of items it can serve from each class. Additionally, each platform also has an upper bound on the total number of items it can serve. The goal is to assign items to platforms so as to maximize the number of items assigned while satisfying the upper bounds of each class. This problem models several important real-world problems like ad-auctions, scheduling, resource allocations, school choice etc. We show that if the classes are arbitrary, then the problem is NP-hard and has a strong inapproximability. We consider the problem in both online and offline settings under natural restrictions on the classes. Under these restrictions, the problem continues to remain NP-hard but admits approximation algorithms with small approximation factors. We also implement some of the algorithms. Our experiments show that the algorithms work well in practice both in terms of efficiency and the number of items that get assigned to some platform.

## 1 Introduction

Graph matching is a fundamental problem in graph theory and theoretical computer science that has been studied extensively over the years. Computing the maximum matching in bipartite graphs, both in the online and the offline setting is an important building block in many applications in allocation problems such as ad-auctions [Mehta, 2013; Mehta *et al.*, 2007], scheduling [McKeown *et al.*, 1999], resource allocation [Halabian *et al.*, 2012], school choice [Abdulkadiroglu and Sönmez, 2003] etc. Since the notation used in these various problems differ, we use the general terms *items* and *platforms* to refer to the two parts of the bipartite graph. In practice, items may be classified based on different properties and hence may belong to certain groups or classes.

Modeling the allocation problems as a vanilla matching problem seeks to optimize the cost of the solution alone and may inadvertently be “unfair” to some classes of items. Necessitated by the need to be fair and unbiased towards any group of items in the input, there has been a lot of recent work studying algorithms for various problems augmented with fairness constraints, such as [Celis *et al.*, 2018; Kay *et al.*, 2015; Costello *et al.*, 2016; Bolukbasi *et al.*, 2016].

In this paper, we enforce *group fairness* through constraints that place an upper bound on the number of items that can be matched from a particular class to a platform. We note that group fairness constraints usually involve both upper and lower bounds. This is incompatible with the practical applications that we have in mind, namely ad-allocation and course allocation problems. For this reason, we focus only on upper bounds. We formally define the problem as follows.

**Classified Maximum Matching (CMM).** We have a set  $A$  of items and a set  $P$  of platforms, and these sets form the two parts of a bipartite graph. The presence of an edge  $(a, p)$  indicates that item  $a$  can be matched to platform  $p$ . Let  $N(p)$  denote the neighborhood of  $p$ . Each platform  $p$  has a collection of classes  $\mathcal{C}_p \subseteq 2^{N(p)}$ , i.e., each item in  $N(p)$  may belong to some of the classes in  $\mathcal{C}_p$ . Each class  $C \in \mathcal{C}_p$  has an associated quota  $q_p^C$  denoting the maximum number of items from  $C$  that can be assigned to  $p$ . In addition, each platform  $p$  has a quota  $q_p$ , which is an upper bound on the total number of items it can serve. Our goal is to compute an assignment of items to platforms so as to maximize the number of items assigned, while satisfying all the quotas.

Classes allow platforms to specify group fairness constraints – for instance the classes can be seen as properties or categories and the quotas impose the constraints that not too many items from one category are assigned to a platform. These types of fairness constraints have been studied in many practical applications. In [Abdulkadiroglu and Sönmez, 2003], the authors address the school choice problem where fairness constraints are imposed to achieve racial, ethnic, and gender balance. In the assignment of medical residents to hospitals in Japanese Residency Matching Program (JRMP), regional quotas are introduced to ensure fairness amongst urban and rural hospitals [Kamada and Kojima, 2012; Kamada and Kojima, 2015]. Huang [Huang, 2010] motivates classifications from the perspective of enabling diver-

\*These three authors contributed equally.

sity in academic hiring. Apart from matchings, group fairness constraints have also been studied for many other problems like the knapsack problem [Patel *et al.*, 2021], and clustering problems [Bera *et al.*, 2019], to name a few.

Some recent pre-prints have discussed fairness in matching problems. Soriano and Bonchi [García-Soriano and Bonchi, 2020] study a different notion of individual fairness that they call maxmin-fairness. Their goal is to output a solution such that the *satisfaction* of one agent cannot be improved without making another agent worse-off. Ma and Xu [Ma and Xu, 2020] measure fairness by the ratio of expected number of agents matched from a particular group to the expected number of agents from that group and their goal is to maximize the minimum of this ratio over all groups. Basu *et al.* [Basu *et al.*, 2020] also measure fairness based on metrics involving the ratio of agents across groups and the utility they provide. While qualitatively similar, our constraints can be seen as being orthogonal to such notions of fairness.

In different applications, the fairness constraints can have a different structure. In [Kamada and Kojima, 2012; Kamada and Kojima, 2015], each hospital belongs to exactly one region, hence fairness constraints partition the set of hospitals. On the other hand, in the school choice problem in [Abdulkadiroglu and Sönmez, 2003], a student belongs to multiple constraints. The structure of the constraints crucially affects the computational complexity of finding a fair allocation. This has been illustrated in the context of bipartite matchings where one or both sides of the bipartition express preferences over the other side. Huang [Huang, 2010] and Fleiner and Kamiyama [Fleiner and Kamiyama, 2012], address the CMM problem when both sides of the bipartition have preferences over each other and the notion of optimality is *stability*, whereas [Nasre *et al.*, 2019] study the CMM problem under the optimality criteria of *rank-maximality* and *popularity*. In all these cases, it has been shown that the CMM problem can be efficiently solved if the constraints have a *laminar*<sup>1</sup> structure, and is NP-hard in general [Nasre *et al.*, 2019]. Such a restriction has been considered before in the literature, such as in the hospital-resident problem [Kamada and Kojima, 2012] or the college admissions problem [Biró *et al.*, 2010; Goto *et al.*, 2016]. However, a finer relation between the structure of the class constraints and the computational efficiency has not received much attention in literature. In this paper, we address this issue by focusing on a quantification of non-laminarity in the classes and its effect on computational efficiency. We strengthen the hardness results in [Nasre *et al.*, 2019] and obtain new approximation algorithms for the CMM problem in the offline setting.

Next, we turn our attention to the online version of the problem. Online matching problems have numerous practical applications, such as in ad-allocations [Mehta *et al.*, 2007], resource allocation [Devanur *et al.*, 2011], etc. See [Mehta, 2013] for a survey on online bipartite matchings. Fairness has also been studied in online settings such as online learning [Gillen *et al.*, 2018] or ride-hailing platforms [Sühr *et al.*, 2019]. ‘Fairness’ in another form has been considered pre-

viously in the online literature. For example, the ‘frequency caps’ mentioned in [Feldman *et al.*, 2009] places a restriction on the total number of ads that are shown to the same user, or users from a particular demographic. We study some natural online versions of the CMM problem. We first show that one of our approximation algorithms for the offline non-laminar case also works as an online algorithm, regardless of the input model. For the setting where we restrict classes to be laminar, we show that existing algorithms for online bipartite matching carry over to our setting.

## 1.1 Models

We study CMM problem in various settings. In practice, assigning an item  $a$  to a platform  $p$  may generate a revenue, which can be modelled as the weight of the edge  $(a, p)$ . In such a case the goal of the weighted CMM problem is to compute an assignment of items to platforms so as to maximize the total weight of edges in the matching, while satisfying the quotas of all the classes. We now formally define our models.

**Model 1 (Many-to-one).** This is the setting described earlier. In this setting, items can match to at most one platform.

**Model 2 (Many-to-many).** This is a more general setting in which items may be matched to multiple platforms. In addition to the classes of platforms, each item  $i$  also has a collection of classes  $\mathcal{C}_i \subseteq 2^{N(i)}$ , i.e., each platform in  $N(i)$  belongs to some of the classes in  $\mathcal{C}_i$ , and the items also have quotas for their classes. This model arises in scenarios like course allocation, where students may be allotted multiple courses subject to various restrictions. Courses may have restrictions over the number of students allotted to it from each department or from each batch.

In the setting of online matchings, the platforms are available offline and the items arrive online. When an item  $a \in A$  arrives online, its neighbours in  $P$ , and the classes that it participates in are revealed. It must be immediately decided if we match  $a$  to some platform and any edges matched cannot be unmatched later. In the literature, the order in which the items arrive has been studied in various models. In the adversarial model, the items can arrive in an arbitrary order. We study a natural online arrival model for the items, called the Random input model. See [Mehta, 2013] for a survey of other work on such models.

**Model 3 (Random input).** In this setting, there is an underlying graph  $G = (A \cup P, E)$ . The vertices of  $A$  arrive according to a permutation chosen uniformly at random.

## 1.2 Our Results

In most applications, an item typically belongs to small number of classes, hence we first study this setting. For example, [Abdulkadiroglu and Sönmez, 2003] discusses the Boston student assignment mechanism which divides students into two categories based on whether they already have a sibling in the school and whether they are within walking distance to the school. Similarly, in a faculty hiring scenario, the number of classes (which would correspond to specializations) is independent of the number of applicants. For the scenario when there are constant number of classes we show the following result.

<sup>1</sup>A family  $\mathcal{C}$  of subsets of a set  $S$  is laminar if, for every pair of sets  $X, Y \in \mathcal{C}$ , either  $X \subseteq Y$  or  $Y \subseteq X$  or  $X \cap Y = \emptyset$ .

**Theorem 1** (Informal version of Theorem 6). *The CMM problem can be solved in polynomial time if there is a constant number of platforms, each with a constant number of classes. This leads to a  $\frac{1}{2}$ -approximation algorithm for an arbitrary number of platforms, each with a constant number of classes.*

Now we turn to a more general setting where the number of classes is arbitrary and exploit the structural relation amongst the classes. We know from [Nasre *et al.*, 2019] that when the classes of every platform form exactly one laminar family then the CMM problem is solvable in polynomial time. We prove the following theorem.

**Theorem 2.** *There is a polynomial-time algorithm achieving an approximation ratio of  $\frac{1}{\Delta+1}$  for the many-to-one setting (Model 1) where each item belongs to at most  $\Delta$  laminar families of classes per platform. This generalizes to the weighted many-to-many setting (Model 2) where for each edge  $(a, p)$ , the classes of  $a$  and  $p$  that contain  $(a, p)$  can be partitioned into  $\Delta + 1$  laminar families.*

The above result is applicable in scenarios like ad-allocation where the number of classes can be arbitrary, but any ad belongs to a few of them. Complementing this, we also obtain hardness results for computing the optimal CMM.

**Proposition 1.** (i) *CMM cannot be approximated to a factor of  $n^{\epsilon-1}$  for any  $\epsilon > 0$  unless  $P = NP$ , where  $n = |A|$ , even when there is a single platform and all edge weights are one.*

(ii) *When there is a single platform, and additionally, each item appears in at most  $\Delta$  classes, the problem is NP-hard to approximate within a factor  $O\left(\frac{\log^2 \Delta}{\Delta}\right)$ .*

The proof of Proposition 1 follows from a reduction from the MAXIMUM INDEPENDENT SET problem.

In the online setting, we first remark that our algorithm from Theorem 2 works as an online algorithm in the unweighted case, even when the input is adversarially chosen.

**Theorem 3.** *There is a polynomial-time online algorithm achieving, in any input model, a competitive ratio of  $\frac{1}{\Delta+1}$  for the many-to-one setting (Model 1) where each item belongs to at most  $\Delta$  laminar families of classes per platform. The algorithm extends to the many-to-many setting (Model 2) where the classes of  $a$  and  $p$  containing each edge  $(a, p)$  can be partitioned into  $\Delta + 1$  laminar families.*

Having achieved a competitive ratio that is close to the lower bound (from Proposition 1 (ii)), we consider the case where classes are restricted to be laminar. We consider the random order input model (Model 3) and show that a simple greedy algorithm from the literature also works for CMM and that it achieves the same competitive ratio. We use the technique of randomized dual fitting which has been used to analyse competitive ratios in works such as [Devanur *et al.*, 2013; Huang *et al.*, 2018].

**Theorem 4.** *There is a polynomial-time algorithm achieving a competitive ratio of  $1 - \frac{1}{e}$  for CMM with laminar classes in the random input model.*

### 1.3 Implications for Other Problems

Although the CMM problem is modelled as a matching of items to platforms, we show that the classes capture problems which are well studied and are of independent interest.

**Maximum Independent Set on Hypergraphs.** Given a hypergraph  $H = (V, E)$ , and a function  $f : E \rightarrow \mathbb{Z}^+$ , compute the largest set of vertices  $S$  such that for every  $e \in E$ ,  $|S \cap e| \leq f(e)$ . We note that when  $f(e) = 1$  for each edge  $e$ , this is the problem of computing the *strong* maximum independent set and when  $f(e) = |e| - 1$ , this is the *weak* maximum independent set problem. These problems are well-studied for bounded-degree hypergraphs; [Halldórsson and Losievskaja, 2009] describe algorithms achieving factors of  $\frac{1}{\Delta}$  and  $\frac{5}{\Delta+3}$  for the strong and weak cases respectively, where  $\Delta$  denotes the maximum degree of a vertex in  $H$ . For the weak independent set, this was further improved to  $O\left(\frac{\log \Delta}{\Delta \log \log \Delta}\right)$  in [Agnarsson *et al.*, 2013]. However, to the best of our knowledge, there is no known approximation algorithm for the case when  $f(e)$  is an arbitrary value – we call this the GENERALIZED MAXIMUM INDEPENDENT SET on hypergraphs. We state our approximation result for independent sets on hypergraphs below, which is a consequence of Theorem 2.

**Proposition 2.** *There is a polynomial time  $\frac{1}{\Delta}$  approximation algorithm for the problem of computing a GENERALIZED MAXIMUM INDEPENDENT SET on hypergraphs with maximum degree  $\Delta$ .*

For the case when average degree of the vertices is  $\Delta$ , we get the following:

**Theorem 5.** *There is a  $\frac{r}{4\Delta}$  approximation algorithm for the generalized independent set where  $r = \frac{OPT}{n}$  and  $\Delta$  denotes the average degree of a vertex.*

For the CMM problem, this implies an  $\frac{OPT}{4\Delta n}$  approximation algorithm when we only have an upper bound on the average number of laminar families of classes an item belongs to, and there is only one platform.

**Ranking and Group Fairness.** In an apparently different model, Celis *et al.* [Celis *et al.*, 2018] consider ranking  $n$  items from a universe of  $m$  items, where  $n \ll m$ . Items are assigned properties, and upper quotas for the number of items from any property in the top  $k$  ranks. When items have at most  $\Delta$  properties each, they give a  $\frac{1}{\Delta+2}$  approximation while allowing constraints to be violated by a factor of 2. This problem can be reduced to the CMM problem and our algorithm from Theorem 2 achieves the same approximation factor without violating any class constraints. We leave the reduction to the full version of this paper [Sankar *et al.*, 2021].

**Simultaneous Matchings.** Kutz *et al.* [Kutz *et al.*, 2008] study the problem called simultaneous matchings which is defined as follows: given a bipartite graph  $G = (X \cup D, E)$  and a collection  $\mathcal{F} \subseteq 2^X$ , find the largest solution  $M \subseteq E$  such that  $\forall C \in \mathcal{F}, M \cap (C \times D)$  is a matching. This problem can be reduced to the CMM problem where every vertex  $d$  in  $D$  has constraints  $\mathcal{F}$  (excluding vertices to which  $d$  has no edge), and each class has quota 1. The approximation factor

in [Kutz *et al.*, 2008] is better but the constraints are significantly more restricted than ours.

## 2 Offline Approximation Algorithms

In light of the strong inapproximability result for the general CMM problem (Proposition 1), we describe approximation algorithms for some special cases.

### 2.1 Proof of Theorem 2

In this section, we consider the case when, for each platform  $p$  and item  $a$ , the classes containing  $a$  can be partitioned into at most  $\Delta$  laminar families. We first present a  $\frac{1}{\Delta}$ -approximation algorithm for the case when there is only one platform. This algorithm also generalizes the maximum independent set in hypergraphs (Proposition 2). We extend this algorithm to a  $\frac{1}{\Delta+1}$ -approximation algorithm for the case with multiple platforms and even to the many-to-many setting.

Let  $G = (A \cup \{p\}, E)$  be an instance of the CMM problem with a single platform  $p$  and a family of classes  $\mathcal{C}$  with the above restriction.

*Reduction to the GENERALIZED MAXIMUM INDEPENDENT SET problem:* We construct an instance  $H = (V, E_H)$  by setting  $V = \{v_i \mid a_i \in A\}$  and  $E_H = \{e_C \mid C \in \mathcal{C}\}$ , and  $f(e_C) = q(C)$ . We call a set  $S \subseteq V$  *feasible* if for every  $e \in E$ ,  $|S \cap e| \leq f(e)$ . We call a set  $S \subseteq V$  *maximal* if  $S$  is feasible and  $S \cup \{v\}$  is not feasible for every  $v \in V \setminus S$ . Our algorithm is a simple greedy algorithm: output a maximal set of vertices. To prove the approximation, we use the following lemma. See the full version of this paper for the proof.

**Lemma 1.** *Consider a set  $S \subseteq V$  and a set  $B \subseteq V \setminus S$  such that  $S \cup B$  is a maximal set of vertices. Then for every feasible set  $C$  such that  $S \subseteq C$  and  $C \cap B = \emptyset$ , we have  $|C| \leq |S| + \Delta|B|$ .*

Let  $ALG$  denote any maximal independent set of  $H$  and  $OPT$  be the optimal independent set. In the above lemma, set  $S = ALG \cap OPT$ ,  $B = ALG \setminus OPT$ ,  $C = OPT$ . The lemma implies  $|OPT| \leq \Delta|ALG|$ . This proves Proposition 2. We note that this also gives us a  $\frac{1}{\Delta}$ -approximation for the CMM problem in the single platform case when every item belongs to at most  $\Delta$  laminar families of the platform. It is also easy to see that this algorithm runs in time  $O(|V||E_H|)$ . For every vertex, we add it if it does not exceed the quota of any edge it belongs to.

**Multiple Platforms.** We can use the previous result to obtain a  $\frac{1}{\Delta+1}$  approximation for the multiple platforms case via a simple  $O(|E|)$ -time reduction to the single platform case: For every edge  $(a, p)$ , make a new item  $e_{a,p}$ . Replace all the platforms by a single dummy platform. Since classes are subsets of the neighbourhood sets of items or platforms, they can also be seen as subsets of edges of the graph. These naturally form classes over the items  $e_{a,p}$ . This combined with the result for the single platform case gives an  $O(|E| \cdot |\mathcal{C}|)$  algorithm for the multiple platform case where  $|\mathcal{C}|$  is the total number of classes, establishing Theorem 2.

*Remark 1* (Weights on items). We remark that the same analysis goes through even if items have weights and the goal is

to compute a maximum weight matching. The algorithm simply keeps matching the highest weight item that can feasibly match to the platform.

### 2.2 Constant Number of Classes

We can also prove Theorem 2 via the following general statement combined with Proposition 2. We leave the proof to the full version of this paper [Sankar *et al.*, 2021]. We will also need this to prove Theorem 1.

**Lemma 2.** *Given a polynomial-time  $\alpha$ -approximation algorithm for the many-to-one matching problem with a single platform, we can obtain a polynomial-time  $\frac{\alpha}{1+\alpha}$ -approximation for the matching problem with multiple platforms.*

**Theorem 6** (Formal version of Theorem 1). *The CMM problem can be represented as an IP with  $2^\Delta$  variables if there is only one platform with  $\Delta$  classes. This can be solved in time  $O(2^{2^\Delta} \text{poly}(n))$ , and also gives rise to a  $\frac{1}{2}$ -approximation in time  $O(2^{2^\Delta} \text{poly}(n))$  for multiple platforms, each with  $\Delta$  classes of items.*

*Sketch.* For an instance with  $\Delta$  classes, every item can be represented by a  $\Delta$ -bit incidence vector of the classes it belongs to. This partitions the items into  $2^\Delta$  types. Our ILP has one variable for each type. The runtime is obtained via [Lenstra, 1983]. Theorem 1 follows by setting  $\Delta = O(1)$ . In practice, there exist integer program heuristics that are faster (for example, see [Fischetti *et al.*, 2005]). For arbitrarily large number of platforms, each with a constant number of classes, we can use this with Lemma 2 to get a  $\frac{1}{2}$ -approximation.  $\square$

### 2.3 Bounded Average Degree

We extend the result from the previous section for a single platform to the case when the average number of laminar families of classes an item belongs to is bounded by  $\Delta$ . We state it in terms of GENERALIZED MAXIMUM INDEPENDENT SET here. Now consider the case where the hypergraph  $H$  constructed above has only bounded average degree of  $\Delta$ .

*Proof of Theorem 5.* Since the average degree is  $\Delta$ , for any  $f$ , there cannot be more than  $\frac{n}{f}$  vertices of degree more than  $f\Delta$ . Suppose we estimate  $r$  and set  $f = \frac{2}{r}$ . We call a vertex *low degree* if its degree is at most  $f\Delta$ , otherwise the vertex is *high degree*. Then the number of low degree vertices is  $\geq n(1 - \frac{r}{2})$ . In the graph induced by the low degree vertices, the size of the optimal independent set is at least  $\frac{OPT}{2}$ , since at most  $\frac{OPT}{2}$  vertices of high degree. We use our  $\frac{1}{\Delta}$  approximation algorithm on the graph induced by the low degree vertices. Since this graph has maximum degree  $\leq \frac{2\Delta}{r}$ , the size of the independent set has size  $\geq \frac{r \cdot OPT}{4\Delta}$ .

Thus, our approximation ratio is at least  $\frac{r}{4\Delta}$ . We finally need to estimate  $r$ . We guess a value of  $OPT$  from 1 to  $n$  and run the above procedure for each of the guesses. Amongst all the solutions that we obtain, we pick the one with the highest cardinality. This is guaranteed to do at least as well as the case when we picked the correct value of  $OPT$ .  $\square$

### 3 Online Algorithms

The online algorithm for Theorem 3 is essentially the same as the one in Proposition 2 and works even for an arbitrary input model. Whenever an item arrives online, we match it to a platform such that the matching remains feasible. If there is no such platform, we leave it unmatched. The output is a maximal set, which by Proposition 2, gives us the required competitive ratio. However, we point out that this only works for the unweighted version.

We now look at an ‘easier’ version of CMM where the classes form a laminar family. This version is known to have an efficient offline algorithm, via the construction of a simple flow network. A similar construction is used to compute the rank-maximal matching in [Nasre *et al.*, 2019]. In this setting, we look at the many-to-many CMM model (Model 2) and assume an input model where the item set arrives in a uniformly random permutation (Model 3). For the sake of the analysis, we assume that a random variable  $y_i$  picked uniformly at random from  $[0, 1]$  for every item  $a_i$ , and the items arrive in the increasing order of  $y_i$ . Therefore the random vector  $\vec{y} := (y_1, y_2, \dots, y_n)$  fully describes the order of arrival of the items. We use  $\vec{y}_{-i}$  to represent the vector after removing  $y_i$  from  $\vec{y}$ . We use the following greedy algorithm, and analyze its competitive ratio (in expectation): Keep an arbitrary, fixed ranking of all the platforms in  $P$ . When an item arrives online, match it to as many platforms as possible, picking the highest ranked ones.

*Sketch of Theorem 4.* We use a linear programming relaxation of our problem to analyze our algorithm. We set the primal values according to the output of our algorithm, thereby ensuring the feasibility of the primal solution. Now we need to construct an appropriate dual solution. We use the following folklore fact about the well-known method of dual fitting in designing algorithms. This technique is used in [Devanur *et al.*, 2013; Huang *et al.*, 2018] among others.

In the primal LP, we have a variable  $x_{ij} = 1 \iff$  item  $a_i$  is matched to platform  $p_j$ . We also have constraints for both the item and platform classes. In the dual LP, we have variables corresponding to constraints in the primal LP. We describe the LP formally in the full version of this paper.

**Fact 1 (Folklore).** Suppose we can set the dual variables such that the primal objective is equal to the dual objective, and the dual constraints of the form  $\alpha \geq 1$  instead satisfy  $\mathbb{E}[\alpha] \geq F$ . Then, our algorithm has a competitive ratio  $F$  in expectation.

Now, we set the dual variables so that the dual constraints have a lower bound of  $F$ . The following is the key lemma in analyzing how the algorithm behaves depending on  $\vec{y}$ . Although it is inspired by [Devanur *et al.*, 2013; Huang *et al.*, 2018; Goel and Mehta, 2008], our many-to-many model (Model 2) is more complicated in that moving one vertex up the ranking can cause more changes to the matching because an item can match to multiple platforms. Even apart from the platform classes, we must take care of item classes as well. To that end, we use the following lemma.

**Lemma 3.** For any  $i, j$  such that  $j \neq i$ , if an item  $a_j$  is matched to some platforms at  $y_i = 1$ , then it cannot be unmatched from any platform at  $y_i = \theta \in [0, 1]$  due to an item class.

Once we have the lemma, we show Theorem 4 the following way. Let  $\alpha \geq 1$  be such a dual constraint. We want to show that  $\mathbb{E}[\alpha] \geq F$  or equivalently,  $\mathbb{E}_{y_{-i}}[\mathbb{E}_{y_i}[\alpha]] \geq F$ . We look at the inner expectation. We fix  $y_{-i}$  and vary  $y_i$  from 0 to 1, and show using dual-fitting arguments and Lemma 3 that  $\mathbb{E}[\alpha] \geq (1 - \frac{1}{e})$ . From Fact 1, this proves Theorem 4.  $\square$

### 4 Experiments

In this section, we present the experimental evaluation of our offline algorithms from Theorem 1 and Theorem 2. We use a total of seven datasets which we categorize as real-world and synthetic datasets. The three real-world datasets are sourced from an elective allocation process at an educational institution. The four synthetic datasets are generated as described below. All experiments were run on a laptop running on a 64-bit Windows 10 Home edition, and equipped with an Intel Core i7-7500U CPU @2.7GHz and 12GB of RAM. For solving integer programs, we used IBM ILOG CPLEX Optimization Studio 20.1 through its Python API. All code was written to run on Python 3.8.

**Real-World Datasets.** We use data from three course-registration periods at an educational institution. Each dataset has around 100 courses and 2000 students. The students and the courses correspond to items and platforms respectively in our model. The edges represent the courses that a student is interested in. The students are partitioned into 13 departments (majors) as well as 5 batches (1st year–5th year). Each course has an overall quota denoting the maximum number of students that can be allotted to it. For each course, we introduce a quota for each department and a quota for each batch. Each course belongs to one of two categories, and each student can be matched to at most one course of each category. The goal is to maximize the number of edges selected subject to these constraints. This can be immediately viewed as an instance of CMM.

**Synthetic Datasets.** Modelled on the real-world datasets, we synthetically generate large instances and compare the performance of our algorithms to the optimal algorithm implemented using a matching Integer Linear Program. The synthetic datasets are generated as follows. Datasets labelled ‘large’ have 500 courses, and 20 departments with 10,000 students in each department. The datasets labelled ‘small’ have 300 courses, and 20 departments with 2,000 students in each department. The students have a degree that is chosen uniformly at random between 3 and 10 in the ‘dense’ datasets and between 3 and 5 in the ‘sparse’ datasets. Students choose their courses randomly based on a non-uniform probability distribution. This distribution is defined by assigning a random ‘popularity’ value to each course. We observe this feature in the real-world dataset, where all courses are not equally popular. We also experiment without this feature, and obtain similar results.

Dataset	$\frac{1}{2}$ -approx	$\Delta$ -approx	OPT
Real-1	1871.5 (0.92)	1899.8 (0.93)	2035 (1)
Real-2	1988.6 (0.92)	2014.0 (0.93)	2170 (1)
Real-3	1938.6 (0.92)	1936.7 (0.92)	2107 (1)

Table 1: Comparison of (average) solution values on the real-world datasets. Relative values are in parentheses.

Dataset	$\frac{1}{2}$ -approx	$\Delta$ -approx	OPT
Real-1	0.39 (1.23)	0.11 (4.29)	0.48 (1)
Real-2	0.43 (1.03)	0.11 (3.89)	0.44 (1)
Real-3	0.33 (1.23)	0.10 (3.90)	0.40 (1)

Table 2: Comparison of (average) running-times in seconds on the real-world datasets. Relative speedups are in parentheses.

We compare our performance and running-time with the optimal solution obtained by solving the standard Matching ILP augmented with the constraints for each class. All running-times include the time taken for file I/O. The solution values and running-times were averaged over 10 runs. Though our algorithms are deterministic, these implementations utilize some randomness because of the use of hash-tables. Observe that since we have two laminar families of classes, Theorem 1 and Theorem 2 provide theoretical guarantees of only  $\frac{1}{2}$  and  $\frac{1}{3}$  respectively. However, the performance of the algorithms on both real-world and random data are close to optimal. All our tables provide absolute values of the solution value and running-time of the algorithm from Theorem 1 (column  $\frac{1}{2}$ -approx) and algorithm from Theorem 2 (column  $\Delta$ -approx), as well as the relative value or relative speedup in comparison to that of the Matching ILP (column OPT).

#### 4.1 Observations

Tables 1, 2 provide the solution values and running times for real-world instances and Tables 3, 4 provide the same for the synthetic datasets. In both these datasets, both of our algorithms output solutions with value at least 90% of the optimum value. This seems to suggest that both real-world or random settings are ‘easier’ than the worst-case instances for our algorithms. Furthermore, we believe that the significantly improved running-time more than makes up for loss of 10% in the output value. The biggest speedups are observed in the ‘large’ datasets, where our algorithms achieve speedups of  $15\times$  and  $30\times$  respectively. This is expected because the ILP takes time exponential in the size of the graph.

## 5 Conclusion

In this paper we gave approximation algorithms for the CMM problem in various offline and online settings. Improving these approximation factors or showing matching lower bounds are natural open questions. There are existing algorithms that break the  $1 - 1/e$  barrier for online bipartite matching without group fairness constraints such as [Feldman *et al.*, 2009]; obtaining similar bounds for online CMM

Dataset	$\frac{1}{2}$ -approx	$\Delta$ -approx	OPT
large-dense	239552 (0.97)	239566.4 (0.97)	247537 (1)
large-sparse	212600.1 (0.97)	211885.1 (0.97)	218622 (1)
small-sparse	72676.4 (0.93)	72821.5 (0.93)	78279 (1)
small-dense	75887.7 (0.95)	76133.4 (0.95)	79827 (1)

Table 3: Comparison of (average) solution values in the synthetic datasets. Relative values are in parentheses.

Dataset	$\frac{1}{2}$ -approx	$\Delta$ -approx	OPT
large-dense	5.68 (14.41)	2.90 (28.21)	81.99 (1)
large-sparse	4.67 (15.14)	2.19 (32.19)	70.73 (1)
small-sparse	1.55 (3.00)	0.46 (10.07)	4.68 (1)
small-dense	1.73 (5.39)	0.58 (16.14)	9.37 (1)

Table 4: Comparison of (average) running-times in seconds in the synthetic datasets. Relative speedups are in parentheses.

is also an interesting open problem.

## Acknowledgements

We acknowledge some initial discussions with Ajay Saju Jacob. We are grateful to the anonymous reviewers for their comments. AL was supported in part by SERB Award ECR/2017/003296 and a Pratiksha Trust Young Investigator Award. MN and PN are supported in part by SERB Award CRG/2019/004757.

## References

- [Abdulkadiroglu and Sönmez, 2003] Atila Abdulkadiroglu and T. Sönmez. School choice: A mechanism design approach. *Am. Econ. Rev.*, 93(3):729–747, 2003.
- [Agnarsson *et al.*, 2013] Geir Agnarsson, Magnús M. Halldórsson, and Elena Losievskaja. Sdp-based algorithms for maximum independent set problems on hypergraphs. *Theor. Comput. Sc.*, 470:1–9, 2013.
- [Basu *et al.*, 2020] Kinjal Basu, Cyrus DiCiccio, Heloise Logan, and Noureddine El Karoui. A framework for fairness in two-sided marketplaces. *arXiv preprint arXiv:2006.12756*, 2020.
- [Bera *et al.*, 2019] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *NeurIPS*, pages 4955–4966, 2019.
- [Bíró *et al.*, 2010] Péter Bíró, Tamás Fleiner, Robert W. Irving, and David F. Manlove. The college admissions problem with lower and common quotas. *Theor. Comput. Sc.*, 411(34-36):3136–3153, 2010.
- [Bolukbasi *et al.*, 2016] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker?

- debiasing word embeddings. In *NIPS*, page 4356–4364, 2016.
- [Celis *et al.*, 2018] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, volume 107, pages 28:1–28:15, 2018.
- [Costello *et al.*, 2016] Matthew Costello, James Hawdon, Thomas Ratliff, and Tyler Grantham. Who views online extremism? individual attributes leading to exposure. *Comput. Hum. Behav.*, 63:311–320, 2016.
- [Devanur *et al.*, 2011] Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *EC*, pages 29–38, 2011.
- [Devanur *et al.*, 2013] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *SODA*, pages 101–107, 2013.
- [Feldman *et al.*, 2009] Jon Feldman, Aranyak Mehta, Vahab S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *FOCS*, pages 117–126, 2009.
- [Fischetti *et al.*, 2005] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Math. Program.*, 104(1):91–104, 2005.
- [Fleiner and Kamiyama, 2012] Tamás Fleiner and Naoyuki Kamiyama. A matroid approach to stable matchings with lower quotas. In *SODA*, pages 135–142, 2012.
- [García-Soriano and Bonchi, 2020] David García-Soriano and Francesco Bonchi. Fair-by-design matching. *Data Min. Knowl. Disc.*, pages 1–45, 2020.
- [Gillen *et al.*, 2018] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In *NIPS*, pages 2605–2614, 2018.
- [Goel and Mehta, 2008] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.
- [Goto *et al.*, 2016] Masahiro Goto, Atsushi Iwasaki, Yujiro Kawasaki, Ryoji Kurata, Yosuke Yasuda, and Makoto Yokoo. Strategyproof matching with regional minimum and maximum quotas. *Artif. Intell.*, 235:40–57, 2016.
- [Halabian *et al.*, 2012] Hassan Halabian, Ioannis Lambadaris, and Chung-Horng Lung. Optimal server assignment in multi-server parallel queueing systems with random connectivities and random service failures. In *ICC*, pages 1219–1224, 2012.
- [Halldórsson and Losievskaja, 2009] Magnús M. Halldórsson and Elena Losievskaja. Independent sets in bounded-degree hypergraphs. *Discrete Appl. Math.*, 157(8):1773–1786, 2009.
- [Huang *et al.*, 2018] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. How to match when all vertices arrive online. In *STOC*, pages 17–29, 2018.
- [Huang, 2010] Chien-Chung Huang. Classified stable matching. In *SODA*, pages 1235–1253, 2010.
- [Kamada and Kojima, 2012] Yuichiro Kamada and Fuhito Kojima. Stability and strategy-proofness for matching with constraints: A problem in the Japanese medical match and its solution. *Am. Econ. Rev.*, 102(3):366–70, 2012.
- [Kamada and Kojima, 2015] Yuichiro Kamada and Fuhito Kojima. Efficient matching under distributional constraints: Theory and applications. *Am. Econ. Rev.*, 105(1):67–99, 2015.
- [Kay *et al.*, 2015] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *CHI*, pages 3819–3828, 2015.
- [Kutz *et al.*, 2008] Martin Kutz, Khaled Elbassioni, Irit Katriel, and Meena Mahajan. Simultaneous matchings: Hardness and approximation. *J. Comput. and Syst. Sc.*, 74(5):884–897, 2008.
- [Lenstra, 1983] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.
- [Ma and Xu, 2020] Will Ma and Pan Xu. Group-level fairness maximization in online bipartite matching. *arXiv preprint arXiv:2011.13908*, 2020.
- [McKeown *et al.*, 1999] Nick McKeown, Venkat Anantharam, and Jean Walrand. Achieving 100% throughput in an input-queued switch. *IEEE T. Comm.*, 47(8):1260–1267, 1999.
- [Mehta *et al.*, 2007] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22–es, 2007.
- [Mehta, 2013] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.
- [Nasre *et al.*, 2019] Meghana Nasre, Prajakta Nimbhorkar, and Nada Pulath. Classified rank-maximal matchings and popular matchings – algorithms and hardness. In *WG*, pages 244–257, 2019.
- [Patel *et al.*, 2021] Deval Patel, Arindam Khan, and Anand Louis. Group fairness for knapsack problems. In *AAMAS*, page 1001–1009, 2021.
- [Sankar *et al.*, 2021] Govind S. Sankar, Anand Louis, Meghana Nasre, and Prajakta Nimbhorkar. Matchings with group fairness constraints: Online and offline algorithms. *arXiv preprint arXiv:2105.09522*, 2021.
- [Sühr *et al.*, 2019] Tom Sühr, Asia J. Biega, Meike Zehlike, Krishna P. Gummadi, and Abhijnan Chakraborty. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *KDD*, pages 3082–3092, 2019.