Proportionally Fair Matching via Randomized Rounding

Sharmila Duppala[1], Nathaniel Grammel[1], Juan Luque[1], Calum MacRury[2], Aravind Srinivasan[1]

[1]Department of Computer Science, University of Maryland, College Park
[2]Graduate School of Business, Columbia University
sduppala@umd.edu, ngrammel@umd.edu, jluque@umd.edu, cm4379@columbia.edu, asriniv1@umd.edu

Abstract

Given an edge-colored graph, the goal of the *proportional fair matching* problem is to find a *maximum weight matching* while ensuring proportional representation (with respect to the number of edges) of each color. The colors may correspond to demographic groups or other protected traits where we seek to ensure roughly equal representation from each group. It is known that, assuming ETH, it is impossible to approximate the problem with ℓ colors in time $2^{o(\ell)}n^{\mathcal{O}(1)}$ (i.e., subexponential in ℓ) even on *unweighted path graphs*. Further, even determining the existence of a non-empty matching satisfying proportionality is NP-Hard. To overcome this hardness, we relax the stringent proportional fairness constraints to a probabilistic notion. We introduce a notion we call δ-ProbablyFair, where we ensure proportionality up to a factor of at most $(1 \pm \delta)$ for some small $\delta > 0$ with high probability. The violation δ can be brought arbitrarily close to 0 for some *good* instances with large values of matching size. We propose and analyze simple and fast algorithms for bipartite graphs that achieve constant-factor approximation guarantees, and return a δ-ProbablyFair matching.

1 Introduction

Graph matching, in particular the special case of bipartite matching, is a classical computational problem with many applications such as ad allocation (Mehta et al. 2007; Mehta 2013), crowdsourcing (Ho and Vaughan 2021; Tong et al. 2016; Hikima et al. 2021; Dickerson et al. 2019), job hiring (Purohit, Gollapudi, and Raghavan 2019), organ exchange (Dickerson, Procaccia, and Sandholm 2013; McElfresh, Bidkhori, and Dickerson 2019; Farnadi et al. 2021; Farhadi, Gilbert, and Hajiaghayi 2022), and ride sharing (Hikima et al. 2021; Nanda et al. 2020; Dickerson et al. 2018). Matching is also a fundamental subroutine in several domains including computer vision (Belongie, Malik, and Puzicha 2002), text similarity estimation (Pang et al. 2016) in natural language processing, machine learning algorithms (Huang and Jebara 2007; Jebara, Wang, and Chang 2009; Huang and Jebara 2011; Choromanski, Jebara, and Tang 2013) and computational biology (Zaslavskiy, Bach, and Vert 2009) among others.

In many applications, algorithms are employed to make decisions that could significantly impact the lives of individuals. In such settings, we ought to ensure fairness and equity in the decision-making process. However, classical algorithms typically do not consider such socially motivated objectives and constraints such as fairness or diversity. For instance, a ride-sharing platform may provide better quality assignments (e.g., with shorter wait times, newer vehicles, or better pricing) to riders from certain demographic groups while providing lower quality assignments to others as studied by (Esmaeili et al. 2023). Similarly, cognitive biases from workers can negatively impact the result of crowdsourced data, which may potentially be mitigated by assigning a diverse set of workers to a wide range of different *task types*.

There are many ways to formalize and incorporate some notion of *fairness* into the computed solutions of matching problems (see, e.g., Esmaeili et al. (2023); Bandyapadhyay et al. (2023)). Following the work of Bandyapadhyay et al. (2023), we consider a notion of *proportional fairness*. This notion is closely related to the notion of *disparate impact* (Feldman et al. 2015) and has been explored in various fundamental problems, including matroid optimization (Chierichetti et al. 2019), clustering (Bera et al. 2019), online matching (Sankar et al. 2021), set packing (Duppala et al. 2023), spectral clustering (Kleindessner et al. 2019), and others. In the present work, we consider a formulation of proportional fairness known as (α, β)-BalancedMatching: given an edge-colored graph (where edge colors may denote membership in specific groups), the objective is to find a maximum weight matching ensuring that the proportion of edges of any color among all matched edges lies between α and β where $0 \leq \alpha \leq \beta \leq 1$. We highlight that (α, β)-BalancedMatching adheres to two key criteria: *restricted dominance*, which limits the fraction of edges selected from any given group to at most β, and *minority protection*, which ensures that the fraction of edges from any given group is at least α.

This notion of fairness can be applied to the applications discussed above, such as in ride-sharing, job hiring, and crowd sourcing. By assigning colors to edges (rather than only to vertices), the *color* is able to capture information about both sides of the match. For instance, in ridesharing, the edge color may encode information about both the driver (e.g., vehicle type, rating) and rider (e.g., race or gender), the distance between the driver and rider, and even pricing information (e.g., whether dynamic or surge pricing is applied). As a concrete example, we may choose to encode information about a rider's race or economic status as well as whether an assignment incurs increased fares due to dynamic pricing; then, an (α, β)-BALANCEDMATCHING should ensure that no racial or economic group receives increased fares disproportionately often compared to other riders. In crowdsourcing, we may use edge color to encode both the task type and demographic information about the worker to ensure that a diverse set of workers are assigned to each type of task (for cognitive or human intelligence tasks, this may increase the diversity of perspectives among responses, and thus the overall quality of the final aggregate data).

As a final example application, consider online advertising: an edge's color may encode information about the type of advertisement (or, in political advertising, the political party sponsoring the ad) and the user's personal information. Here, an (α, β)-BALANCEDMATCHING would limit the extent of *targeted* advertising based on protected demographic traits, which may violate (or appear to violate) a user's expectations and rights with regards to their privacy and use of their data; the social consequences of such indiscriminate highly targeted advertising were seen in the real world with the Cambridge Analytica data scandal around the 2016 US presidential election, which saw Facebook's CEO Mark Zuckerberg testify before congress (Lewis and Hilder 2018; Confessore 2018).

Unfortunately, although this notion of fairness seems quite powerful, it may in fact be too strong in the basic formulation: Bandyapadhyay et al. (2023) demonstrated that the problem of (α, β)-BALANCEDMATCHING is NP-hard to approximate, even when G is an *unweighted path graph*. To address this fundamental hardness, we consider a slightly relaxed probabilistic notion of proportional fairness. Roughly speaking, we allow for small violations of the α and β bounds while still guaranteeing a constant factor approximation on the overall objective, i.e., the weight of the matching. Particularly, we ask the following question: *does there exists an approximation algorithm with constant factor violations in fairness while guaranteeing a good approximation on the objective?*

We answer this question by designing a $\frac{1}{2}$-approximate algorithm for bipartite graphs and whose matching is *probably*[1] *almost fair*. That is, there exists a small constant $\delta > 0$ for which the matching is $(\alpha(1-\delta), \beta(1+\delta))$-balanced with high probability. A formal definition of this notion is given in Section 3. Thus, while the (α, β)-BALANCEDMATCHING formulation of Bandyapadhyay et al. (2023) remains hard on bipartite graphs, we can achieve our notion of *probably almost fair*. We note that for the special case when $\alpha = 0$ and $\beta < 1$, we can attain a constant approximation ratio without violating the fairness constraint at all. Combined with the hardness result of Bandyapadhyay et al. (2023), this implies that the one-sided fairness case is strictly easier than the two-sided case.

2 Related Work

A significant body of research in fair matching explored various notions of fairness. Among these, several notions of *group fairness*, such as socially-fair matching (Bandyapadhyay et al. 2023), focus on Rawlsian (maxmin) fairness, aiming to maximize the utility for the worst-off group (Esmaeili et al. 2023). On the other hand, proportional fairness, as explored in Bandyapadhyay et al. (2023); Chierichetti et al. (2019), ensures a proportional representation of edges from each group, while leximin fairness (García-Soriano and Bonchi 2020) is also considered. Most of these studies involve assigning group memberships to vertices in the graph. However, works such as (Bandyapadhyay et al. 2023; Chierichetti et al. 2019) investigate fairness in edge-colored graphs, which aligns with our area of study. We note that (Bandyapadhyay et al. 2023; Chierichetti et al. 2019) focus only on the unweighted setting where the objective is the cardinality/size of the output. That being said, our works are incomparable. While ours applies to edge weighted bipartite graphs, (Chierichetti et al. 2019) consider when the constraint system is described by the intersection of two-matroids (this generalizes unweighted bipartite matching). They focus on the case of two colors, and present a polynomial-time algorithm achieving a $3/2$-approximation. Bandyapadhyay et al. (2023) consider when input is a general graph and the number of colors may be greater than two. They show that approximating the problem for an arbitrary number of colors is NP-hard and moreover that approximating the optimal solution requires time exponential in ℓ unless the Exponential Time Hypothesis (ETH) (Impagliazzo and Paturi 1999) is false. Their main algorithm therefore runs in time exponential in ℓ, with an approximation guarantee of $1/(4\ell)$. Note that even for the easier case when $\alpha = 0$, their algorithm may still violate the fairness constraint imposed by β by a factor up to $1 + 1/\ell$. Their hardness result motivates our study of the slightly relaxed probabilistic fairness. This relaxation allows us to improve on the prior work by achieving, in polynomial time, a *constant factor* (i.e., one half) approximation in the matching weight while ensuring, with high probability, only a small violation in the fairness constraints.

3 Preliminaries and Problem Formulation

We denote $[k] = \{1, \ldots, k\}$ for any positive integer k. Consider an undirected bipartite graph $G = (U, V, E)$ with the set of edges E forming a partition $\dot\cup_{c \in [\ell]} E_c$, where $\dot\cup$ denotes a disjoint union over the sets E_c. Each color $c \in [\ell]$ is associated with the set of edges E_c, representing edges of color c. For any color class c, we define $\psi_c(e)$ such that $\psi_c(e) = 1$ if $e \in E_c$, and $\psi_c(e) = 0$ otherwise. A *matching* $M \subseteq E$ in G is defined as a subset of edges where no two edges in M share a common vertex. For a vertex v of G, let $N(v)$ and $\delta(v)$ denote the neighbors and incident edges of v, i.e., $N(v) := \{u : (u, v) \in E\}$ and $\delta(v) := \{(u, v) : (u, v) \in E\}$. For any $\alpha, \beta \in [0, 1]$ and $\alpha \leq \beta$, we define a matching $M \subseteq E$ as (α, β)-BALANCED if, for each color $c \in [\ell]$, the proportion of edges in M belonging to color c lies between α and β. In other words,

[1] i.e., with probability close to 1

M is (α, β)-BALANCED if it contains at least α and at most β fraction of edges from every color. The goal of the *proportional fair matching problem* (PFM) is to find a (α, β)-BALANCEDMATCHING of G with maximum weight, and we denote the weight of such a matching by OPT. Since (Bandyapadhyay et al. 2023) proved that it is NP-Hard to verify the existence of a (α, β)-BALANCEDMATCHING in the PFM problem even on unweighted path graphs, we focus on deriving approximation ratios which hold against OPT.

3.1 Our Contributions and Techniques

We design Algorithm 1, an efficient randomized algorithm for weighted matching on bipartite graphs with fairness constraints specified by $0 < \alpha \leq \beta < 1$. By allowing the α (respectively, β) fairness constraint to be violated up to a multiplicative factor of $1 - \delta$ (respectively, $1 + \delta$), we show that Algorithm 1 attains an asymptotic approximation ratio against OPT. (We write the exact asymptotic guarantee of Algorithm 1 in Theorem 4.1). We argue that if the input is well-behaved, then we can take $\delta \approx 0$, and so we barely violate the fairness constraints, while still attaining a constant approximation ratio. We next analyze Algorithm 1 when $\alpha = 0$ and $\beta < 1$. By allowing for a $\frac{1-\varepsilon}{2}$-approximation ratio, where $0 < \varepsilon < 1$ is a parameter of Algorithm 1 which can be chosen arbitrarily small, we show how to ensure that the β fairness constraints are satisfied exactly. This allows us to get a true approximation ratio which is arbitrarily close to $1/2$. We also note that our algorithms can be extended to the slightly more general setting where we have different proportionality requirements for each color; e.g., where we have values α_c, β_c (with $0 \leq \alpha \leq \beta \leq 1$) for each color $c \in [\ell]$, and we insist that $\alpha_c|\mathcal{M}| \leq |\mathcal{M}_c| \leq \beta_c|\mathcal{M}|$ for each c. This allows us to capture settings where we wish to represent different groups in different proportions, for example so that each group's representation in the matching is roughly proportional to its representation in the original graph.

We take a linear programming (LP) based approach to designing Algorithm 1. Below we state LP-FAIR, which extends the standard matching polytope to include the proportionality constraints described by $0 < \alpha \leq \beta < 1$:

$$\max \sum_{e \in E} w_e x_e \qquad \text{(LP-FAIR)}$$

$$\text{s.t.} \sum_{e \in \delta(v)} x_e \leq 1, \forall v \in U \cup V \qquad \text{(1a)}$$

$$\alpha \sum_{e \in E} x_e \leq \sum_{e \in E_c} x_e \leq \beta \sum_{e \in E} x_e, \qquad \forall c \in [\ell] \qquad \text{(1b)}$$

$$x_e \geq 0, \forall e \in E \qquad \text{(1c)}$$

Lemma 3.1. LP-FAIR *relaxes* OPT. *That is,* OPT $\leq \sum_{e \in E} w_e x_e$*, where* $\boldsymbol{x} = (x_e)_{e \in E}$ *is an optimal solution to* LP-FAIR.

As a first attempt to using LP-FAIR, observe that since G is bipartite, if $\boldsymbol{x} = (x_e)_{e \in E}$ is an optimal solution to the LP, then we can write $\boldsymbol{x}$ as a convex combination of integral matchings. By randomly sampling such a matching according to the coefficients of the convex combination, this yields a randomized algorithm with expected value $\sum_{e \in E} w_e x_e \geq$ OPT. Unfortunately, while this preserves the fairness constraints in expectation due to (1b), there is no guarantee that any *particular* matching we output will satisfy these constraints. In order for an algorithm to succeed with constant probability, an ideal approach would be to match each edge e with probability x_e while ensuring that the size of a matching of a color class c is concentrated about $\sum_{e \in E_c} x_e$. For instance, if we had *negative correlation* amongst the matching statuses of the edges of E_c, then this would be sufficient (see (Dubhashi and Ranjan 1996)). The GKPS dependent rounding scheme of (Gandhi et al. 2006) provides such a guarantee for *certain* edge subsets of G. However, since the edges of E_c are adversarially chosen, it is easy to construct an example where the GKPS rounding scheme induces positive correlation amongst the matched statuses.

Due to the limitations of these approaches, we need to round our LP in a different way. Let us suppose that we round $\boldsymbol{x}$ into a random matching $\mathcal{M}$, where we denote $\mathcal{M}_c := \mathcal{M} \cap E_c$ for a fixed color class $c \in [\ell]$. Roughly speaking, our goal is to ensure that the following properties simultaneously hold, where $\delta > 0$ is a fixed parameter to be specified in Theorem 4.3:

1. For each $e \in E$, $\Pr[e \in \mathcal{M}] = x_e/2$.
2. With constant probability,

$$\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \in \left[(1-\delta)\frac{\mathbb{E}[|\mathcal{M}_c|]}{\mathbb{E}[|\mathcal{M}|]}, (1+\delta)\frac{\mathbb{E}[|\mathcal{M}_c|]}{\mathbb{E}[|\mathcal{M}|]}\right].$$

Property 1 and Lemma 3.1 then imply that

$$\mathbb{E}[\sum_{e \in \mathcal{M}} w_e] = \sum_{e \in E} w_e x_e/2 \geq \text{OPT}/2.$$

Moreover, since $\mathbb{E}[|\mathcal{M}_c|] = \sum_{e \in E_c} x_e/2$ and $\mathbb{E}[|\mathcal{M}|] = \sum_{e \in E} x_e/2$, we can combine Property 2. with (1b) to conclude that $(1-\delta)\alpha|\mathcal{M}| \leq |\mathcal{M}_c| \leq (1+\delta)\beta|\mathcal{M}|$.

Our approach uses a randomized rounding tool known as a *contention resolution scheme (CRS)*. CRS's were originally introduced by (Chekuri, Vondrák, and Zenklusen 2014) to solve certain constrained sub-modular optimization problems. Motivated by the applications to prophet inequalities, they were later adapted to the online setting by (Feldman, Svensson, and Zenklusen 2021), where they are referred to as *online* contention resolution schemes (OCRS's). Since then, they have found many other applications, and have become a fundamental tool in stochastic optimization. We continue this line of research and show that this tool can be useful even for our problem which is offline and has no inherit stochasticity.

Our algorithm first has every $v \in V$ independently draw a random vertex $F_v \in N(v)$, where

$$\Pr[F_v = u] = x_{u,v} \qquad (2)$$

for each $u \in N(v)$. (For convenience, we define F_v to be a *null element* $\perp$ if no draw is made, where $\mathbb{P}[F_v = \perp] = 1 - \sum_{u \in N(v)} x_{u,v}$). If $F_v = u$, we say that v *proposes* to

u and refer to F_v as a *proposal* for u. At this point, we know that each vertex $v \in V$ has made at most one proposal. However, a fixed vertex $u \in U$ may have received multiple proposals, and so we must resolve which proposal each vertex u should take. This is precisely the purpose of the OCRS, and we use the explicit scheme of (Ezra et al. 2022). For a fixed $u \in U$, the input required of the OCRS is the edge values $(x_{u,v})_{v \in N(u)}$, together with the proposals $(F_v)_{v \in N(u)}$. Moreover, in the contention resolution framework, $(F_v)_{v \in N(u)}$ must be independent. Under these conditions, the OCRS outputs at most one edge (u, v) with $F_v = u$, while guaranteeing that for *all* $v \in N(u)$

$$\Pr[(u, v) \text{ is output} \mid F_v = u] = 1/2. \tag{3}$$

Due to guarantee of (3), the OCRS is said to be 1/2-*selectable* in the literature. By concurrently running a separate execution of this OCRS for each $u \in U$, the matching we output satisfies Property 1.

We still need the explain why our algorithm satisfies Property 2. It turns out our algorithm has a very simple description. First, we order the vertices of V arbitrarily, say $v_1, \ldots, v_n$. Then, when v_t is processed and proposes to u (i.e., $F_{v_t} = u$), we draw an *independent* random bit A_{u,v_t} We match (u, v_t) provided u was not previously matched and $A_{u,v_t} = 1$. We note that in regards to verifying Property 2, the specific Bernouli parameter of A_{u,v_t} is not important. We simply require that $(A_e)_{e \in E}$ are drawn independently.

If we let $\mathcal{M}$ be the output of the algorithm, and $\mathcal{M}_c = E_c \cap \mathcal{M}$, then we can analyze the Doob martingale of $|\mathcal{M}_c|$. Here the Doob martingale is defined by revealing the partial matching after the vertices $v_1, \ldots, v_t$ are processed. Due to the simplicity of our algorithm, we can bound the one-step changes in the Doob martingale in a very precise way that is related to $\sum_{e \in E_c} x_e$. Note that the standard Azuma–Hoeffding martingale concentration inequality does not allow for good bounds due to constant worst-case one-step changes (as we explain in Section 5.1 and for more details we refer to the full version of our paper (Duppala et al. 2024)) leading to $\Theta(n)$ over the entire process from $t = 1$ to $t = n$. We show that by controlling the *variance* of this martingale, we can get stronger bounds via Freedman's inequality. More precisely, we can tighten the concentration on the random variable $\mathcal{M}_c$ by using one-step variance which can be much smaller than the worse-case one-step changes since the events involving the bad cases occur with small probability. The total variance can be precisely computed and is upper bounded by $10 \sum_{e \in E_c} x_e$ which is much smaller than $\Theta(n)$. The concentration helps us achieve *almost* fairness, i.e., $|\mathcal{M}_c|$ violates the fairness constraints on each side by at most a δ-multiplicative factor for some small $\delta > 0$. Finally, when we only have β-sided fairness constraints i.e., $\alpha = 0, \beta < 1$, we apply an LP *perturbation trick* to recover a matching that satisfies the constraints *exactly*, meaning that without any violations with only a $0 < \epsilon < 1$ multiplicative loss in the objective. However, it is important to note that our algorithm still has a probabilistic guarantee: with a certain probability, the matching returned will *not* satisfy the fairness constraints. At the expense of additional runtime, we can improve the success probability to be arbitrarily close to 1 by repeatedly running our algorithm and taking the best solution which is feasible.

4 Our Algorithm

We begin by formally describing our algorithm to find a matching for a bipartite graph $G = (U, V, E)$ with color classes $\cup_{c \in [\ell]} E_c$ and $\alpha \leq \beta \leq 1$. Note that our algorithm takes in a parameter $0 < \varepsilon < 1$ which is only needed for the special case when $\alpha = 0$.

For $\delta > 0$, a random matching $M \subseteq E$ is said to be δ-ProbablyFair with respect to $0 \leq \alpha \leq \beta \leq 1$ if for any color class c, [2]

$$\Pr\left[(1-\delta)\alpha \leq \frac{|M_c|}{|M|} \leq (1+\delta)\beta\right] \geq 1 - f_c(\delta, G). \tag{4}$$

where $f_c(\delta, G)$ is a term that depends on the tunable parameter δ and the input instance. This definition allows us to violate the constraints by a small δ-multiplicative factor which can be adjusted to attain a reasonable probability of success.

Algorithm 1 OCRS Rounding Algorithm

```
Input: $G = (U, V, E)$ with color classes $(E_c)_{c \in [\ell]}$, $0 \leq \alpha \leq \beta \leq 1$, and $0 < \varepsilon < 1$.
Output: subset of edges forming a matching $\mathcal{M}$.
$\mathcal{M} \leftarrow \emptyset$
If $\alpha > 0$, compute an optimal solution $\boldsymbol{x} = (x_e)_{e \in E}$ to Lp-Fair with $0 < \alpha \leq \beta \leq 1$. Otherwise, compute an optimal solution $\boldsymbol{x}$ to Lp-Fair with $\beta$ in (1b) replaced by $\tilde{\beta} = (1-\varepsilon)\beta$.
Order the vertices of $V$ as $v_1, \ldots, v_n$ arbitrarily.
Draw $(F_{v_t})_{t=1}^{n}$ as described in (2).
for $t = 1, \ldots, n$ with $F_{v_t} \neq \perp$ do
    Set $u := F_{v_t}$, and $a_{u,v_t} := \frac{1/2}{1-(1/2)\sum_{i<t} x_{u,v_i}}$.
    Draw $A_{u,v_t} \sim \mathrm{Ber}(a_{u,v_t})$ independently.
    If $A_{u,v_t} = 1$ and $u$ is currently unmatched, add $(u, v_t)$ to $\mathcal{M}$
return $\mathcal{M}$.
```

Our main results are as follows:

Theorem 4.1. *Algorithm 1 returns a matching $\mathcal{M}$ that is δ-*ProbablyFair *with* $f_c(\delta, G) = 4\exp\left(-\delta^2 \frac{\sum_{e \in E_c} x_e}{28}\right)$, *and has an expected weight of at least* $\frac{1}{2}\mathrm{OPT}$.

Remark 4.2. *The violations dependent on δ in the fairness constraints are inversely exponential to the failure probability $f_c(\delta, G)$. This implies that achieving arbitrarily small violations $\delta \approx 0$ with a high success probability requires stronger assumptions on the term $\sum_{e \in E_c} x_e$ i.e., for input instances with $\sum_{e \in E_c} x_e = \omega(1)$ we can ensure high success probability with $\delta \approx 0$.*

We next state our improved guarantee for the special case when $\alpha = 0$. Recall that in this setting, we can ensure that we do *not* violate the β-fairness constraint.

[2] Taking $\delta = 1$ satisfies the Equation (4) trivially

Theorem 4.3. *Given any $0 < \varepsilon < 1$, Algorithm 1 returns a matching that satisfies the β-fairness constraints with probability at least $1 - f(\epsilon, G)$ where $f(\epsilon, G) = 2\exp\left(-\epsilon^2 \beta \frac{\sum_{e\in E} x_e}{28}\right)$, and has an expected weight of at least $\frac{1}{2}(1-\epsilon)\mathrm{OPT}$.*

In Theorem 4.3, the values $(x_e)_{e\in E}$ represent the optimal fractional solution to LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$. Additionally, $f(\epsilon, G)$ denotes the probability that the algorithm fails to produce a matching that satisfies the β-fairness constraints *exactly* i.e., with no violations.

Remark 4.4. *The probability that our algorithm fails decreases exponentially as a function of ϵ and $\beta\sum_{e\in E} x_e$. Therefore, to attain a failure probability $f(\epsilon, G) \approx 0$ and the approximation factor close to $1/2$, we require that $\beta\sum_{e\in E} x_e$ to be sufficiently large, say $\beta\sum_{e\in E} x_e \geq C$ for some constant C, say $C \geq 100$.*

Remark 4.5. *For the instances where $\beta\sum_{e\in E} x_e \leq C$, we address these cases separately using a brute force algorithm as described in Section 6.1. This involves enumerating all feasible matchings that satisfy the fairness with respect to β and having size at most $\frac{U^2}{L^2}\frac{C}{\beta(1-\epsilon)}$ where $U = \max_{e\in E} w_e$ and $L = \min_{e\in E} w_e$.*

For all non-degenerate instances, $\frac{U}{L}$ is bounded by a constant. Hence, our algorithm runs in polynomial time if the term $\frac{U^2}{L^2}\frac{C}{(1-\epsilon)\beta}$ remains constant, given that $\frac{U}{L}$is bounded. However, in the worst-case, where $\beta = \frac{1}{\ell}$, the running time will be exponential in ℓ.

5 Algorithm Analysis

Suppose that we are given an instance of proportional fair matching. We let $\boldsymbol{x} = (x_e)_{e\in E}$ denote an optimal solution to LP-FAIR used by Algorithm 1. Note that if $\alpha = 0$, then $\boldsymbol{x}$ is computed by replacing β in (1b) with $\tilde{\beta} = (1-\varepsilon)\beta$.

Recall that the vertices of V are processed in order $v_1, \ldots, v_n$. Let us say that $u \in U$ is *safe* at time t, provided u is *not* matched to any of $v_1, \ldots, v_{t-1}$. Observe then that (u, v_t) is matched if and only if u is safe at time t, $F_{v_t} = u$ and $A_{u,v_t} = 1$. Let Z_{u,v_t} be an indicator random variable for this event. For convenience, we define $\tilde{F}_{v_t} := u$ provided $F_{v_t} = u$ and $A_{u,v_t} = 1$. Otherwise, $\tilde{F}_{v_t} = \perp$. Observe then that

$$\Pr[Z_{u,v_t} = 1] = \Pr[\tilde{F}_{v_t} = u, u \text{ is safe at } t] \tag{5}$$

Let $Z(v_t)$ be an indicator random variable that an edge of E_c is matched to vertex v_t where $Z(v_t) = \sum\limits_{u\in N(v_t)} \psi_c(u, v_t) Z_{u,v_t}$. Recall that $\mathcal{M}$ is the matching returned by our algorithm, therefore,

$$|\mathcal{M}_c| = \sum_{t\geq 1} \sum_{(u,v_t)\in\delta(v_t)} \psi_c(u, v_t) Z_{u,v_t}$$

Now, the instantiations of $(\tilde{F}_{v_i})_{i=1}^t$ are sufficient to determine which edges (if any) are matched to $v_1, \ldots, v_t$. Thus, we shall work with $\mathcal{H}_t$, the sigma-algebra generated by $(\tilde{F}_{v_i})_{i=1}^t$, which intuitively corresponds to the history of the algorithm's execution after $v_1, \ldots, v_t$ are processed. We define $\mathcal{H}_0$ to be the trivial sigma-algebra, where we do not condition on anything.

Fix a color class c, and let $M_t := \mathbb{E}[|\mathcal{M}_c| \mid \mathcal{H}_t]$ denote the expected number of edges from color class c in our matching, conditional on $\mathcal{H}_t$. Observe that $M_0 = \mathbb{E}[|\mathcal{M}_c|]$, and M_n is the actual number of edges chosen from the color class c at the end of our algorithm. Since we expose strictly more information in each round, i.e., formally, $\mathcal{H}_{t-1} \subseteq \mathcal{H}_t$ for each $t \geq 1$, the sequence $(M_t)_{t=0}^n$ is a martingale with respect to $(\mathcal{H}_t)_{t=0}^n$. (This is often referred to as a Doob martingale).

We aim to understand the size of the matching produced by our algorithm as each vertex v_t is processed. More specifically, when v_t is processed, our algorithm adds the edge (u, v_t) to the current matching if u is safe and $\tilde{F}_{v_t} = u$. Observe that M_t is a function of the current matching between U and $\{v_1, v_2, \ldots, v_t\}$. Moreover, for any $1 \leq t \leq n$,

$$M_t = \mathbb{E}\Big[\sum_{k\geq 1} Z(v_k) \mid \mathcal{H}_t\Big]. \tag{6}$$

Our goal is to control the one step changes of our martingale, which will imply that $M_n = |\mathcal{M}_c|$ is concentrated about $M_0 = \mathbb{E}[|\mathcal{M}_c|]$.

5.1 Warmup: Azuma-Hoeffding Inequality

To apply concentration bounds like the Azuma–Hoeffding inequality, the martingale must be well-behaved i.e., at any step $1 \leq t \leq n$, the martingale cannot change dramatically. Thus, we need to determine the quantities c_t such that if $\Delta M_t := M_t - M_{t-1}$ is the one-step change at step t, then $|\Delta M_t| \leq c_t$.

Lemma 5.1. $|\Delta M_t| = |M_t - M_{t-1}| \leq 2$ *for all* $1 \leq t \leq n$.

Let consider the one-step change in ΔM_t when $\tilde{F}_{v_t}$ is revealed. Since we can match at most one edge at each time step, we may miss the opportunity to match at most *two* edges of a given color. Therefore, in the worst case, we have that $\sum_t |\Delta M_t| = \Theta(n)$. For example, consider the star graph, where a central vertex is connected to all other vertices. In each time step, if we reveal $\tilde{F}_{v_t}$, then the change in the maximum matching will be at most 1, as we can only match or unmatch one edge (we defer the details of the example to the full version of our paper (Duppala et al. 2024)). To remedy this situation, we use a "second order" inequality for analyzing a martingale called Freedman's inequality:

Theorem 5.2 ((Freedman 1975))**.** *Suppose $(M_t)_{t\in[n]}$ is a martingale such that $|\Delta M_t| \leq \Lambda$ for any $1 \leq t \leq n$, and $\sum_{t\in[n]} \mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] \leq \nu$. Then, for all $\lambda > 0$,*

$$\Pr\left[|M_n - M_0| \leq \lambda\right] \leq 2\exp\left(\frac{-\lambda^2}{2(\nu + \lambda\Lambda)}\right) \tag{7}$$

Remark 5.3. *The variance term $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ reduces to $\mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] = \mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}]$ since $\mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] := \mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}] - \mathbb{E}[\Delta M_t \mid \mathcal{H}_{t-1}]^2$ and $\mathbb{E}[\Delta M_t \mid \mathcal{H}_{t-1}] = 0$ due to the martingale property.*

Freedman's inequality still depends on worst case one-step change Λ i.e., an upper bound on $|\Delta M_t|$. However, its dependence on this parameter is less severe compared to the Azuma–Hoeffding inequality. The key difference is that Freedman's inequality depends on $\mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}]$, and so we get to *average* over the randomness in $\tilde{F}_{v_t}$. This is much smaller than than the pessimistic bound of Lemma 5.1, as even conditional on $\mathcal{H}_{t-1}$, the probability that a worst-case change in $|\Delta M_t|$ occurs is small.

5.2 Concentration of $|\mathcal{M}_c|$ via Freedman's Inequality

Freedman's inequality allows us to use the expected one-step changes instead of the worst case changes as shown in the Observation 5.1 below.

Observation 5.1. $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) \leq 2\mathbb{E}[|\Delta M_t| \mid \mathcal{H}_{t-1}]$

We first prove Lemma 5.4 which establishes a bound on the one-step changes in the random variable $Z(v_k)$ at any step t where $k \geq t$. The proofs can be found in the full version of the paper (Duppala et al. 2024). Notice that these one step differences are in the worst case; therefore they can be as large as 2 as discussed in the previous section. However, when we compute the one-step variances, we get to take an expectation over the randomness of $\tilde{F}_{v_t}$ as shown in Lemma 5.5 and Lemma 5.6. This allows us to bypass this worst-case bound of 2.

Lemma 5.4. *For any step* $1 \leq t \leq n$ *and* $u \in N(v_t)$*,*

- $\left|\mathbb{E}\big[Z(v_t) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z(v_t) \mid \mathcal{H}_{t-1}\big]\right| \leq \psi_c(u, v_t) + \sum\limits_{w \in N(v_t)} \psi_c(w, v_t) x_{w,v_t}$
- $\left|\mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1}\big]\right| \leq \psi_c(u, v_k) x_{u,v_k} + \sum\limits_{w \in N_{\bar{u}}(v_t)} \psi_c(w, v_k) x_{w,v_k} x_{w,v_t}$ *if* $k > t$.

Following this lemma, we can say that the one-step change in the size of the matching restricted to color class c at any step t, conditioned on the event $\tilde{F}_{v_t} = u$, is given by:

$$\begin{aligned}&\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \\ &= \sum_{k \geq t} \big(\mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}]\big).\end{aligned}$$

Therefore, by using the Lemma 5.4 we can derive the following result about the expected one-step change in the size of the matching restricted to color class c when $\tilde{F}_{v_t} = u$.

Lemma 5.5. *For any step* $1 \leq t \leq n$ *and* $u \in N(v_t)$*,*

$$\sum_{t \in [n]} \sum_{u \in N(v_t)} x_{u,v_t} |\Delta M_t(u)| \leq 4M_0$$

where $\Delta M_t(u) = \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}\big]$.

We next consider the case when $\tilde{F}_{v_t} = \perp$. Recall that this means $F_{v_t} = \perp$, or $F_{v_t} = u$ for some $u \in N(v_t)$, yet $A_{u,v_t} = 0$.

Lemma 5.6. *For any step* $1 \leq t \leq n$*,*

$$\sum_{t \in [n]} \left|\mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp\big] - \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}\big]\right| \leq 2M_0.$$

Therefore, by combining Lemma 5.5 and Lemma 5.6, we can easily derive a bound on the sum of one-step variances i.e., $\sum_{t \in [n]} \mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ used in the Freedman's inequality. Formally, we can prove the following.

Lemma 5.7. $\sum_{t \in [n]} \mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) \leq 12M_0$ *where* $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ *is the variance in the one-step change of* M_n *at time* t *and* $M_0 = \mathbb{E}[|\mathcal{M}_c|]$.

Therefore, by applying the Freedman's inequality (i.e., Theorem 5.2) we get the desired concentration for the size of matching from color class c.

Theorem 5.8. *For any color class* c*, Algorithm 1 returns a matching* $\mathcal{M}_c = \mathcal{M} \cap E_c$ *that is sharply concentrated around its expected value* $M_0 = \mathbb{E}[|\mathcal{M}_c|]$. *Specifically, for any* $\delta > 0$*, we have that:*

$$\Pr\Big[\big||\mathcal{M}_c| - M_0\big| \geq \delta M_0\Big] \leq 2\exp\Big(-\frac{\delta^2 \sum_{e \in E_c} x_e}{28}\Big)$$

This concludes that Property 2 described in Section 3 can be satisfied. Finally combining Property 2 together with the fairness constraints in (1b), we can conclude the following lemma.

Lemma 5.9. *For any* $c \in [\ell]$*, and* $\delta > 0$*, Algorithm 1 returns a matching* $\mathcal{M}$ *that satisfies the following:*

$$\begin{aligned}\Pr&\left[(1-\delta)\alpha \leq \frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq (1+\delta)\beta\right] \\ &\geq 1 - 4\exp\left(-\frac{\delta^2 \sum_{e \in E_c} x_e}{28}\right)\end{aligned}$$

Lemma 5.9 proves that Algorithm 1 returns a δ-ProbablyFair matching. This lemma, along with the edge selectability guarantees of the OCRS rounding algorithm as stated in Property 1 completes the proof of Theorem 4.1.

6 Exact Fairness for β-Fairness Constraints in Bipartite Graphs

In the previous section we showed that our algorithm provides a $1/2$-approximation of the objective, provided we violate the two-sided fairness constraints by multiplicative factors dependent on δ. Suppose that $\alpha = 0$ and only β imposes the fairness constraint, we can argue that by using a simple *LP perturbation* trick, we can remove the δ-violations and make it an *exact* guarantee while still attaining an asymptotic $1/2$-approximation of the objective. More precisely, we satisfy the β-sided fairness constraint *exactly*. We are also able to provide a better bound on our failure probability. Previously, this was $f_c(\delta, G)$, and it depended on $\sum_{e \in E_c} x_e$. We improve this to a new, *smaller* failure probability $f(\epsilon, G)$, which now only depends on $\beta \sum_{e \in E} x_e$ and a tunable parameter ϵ. This makes our algorithm more robust to the structure of the input instance.

If $\alpha = 0$, we modify our algorithm slightly. We begin by solving a perturbed LP where β in LP-FAIR is replaced by $\beta(1-\epsilon)$ where $0 < \epsilon < 1$. The perturbed LP requires that any feasible solution must satisfy a stronger β-fairness constraint, which results in a slightly weaker solution to our problem (see Lemma 6.1). Let $\boldsymbol{x}$ denote the optimal solution to the perturbed LP i.e., LP-FAIR with $\tilde{\beta}$. Since our algorithm still executes an OCRS on each $u \in U$ concurrently, Equation (3) ensures that the matching $\mathcal{M}$ has expected weight $\sum_{e\in E} w_e x_e/2$. We now relate $\sum_{e\in E} w_e x_e$ to the optimal solution to LP-FAIR when the fairness constraint is β. This will allow us to lower bound the expected weight of our matching in terms of OPT in Corollary 6.2. (Recall that OPT is the weight of the optimal matching in the original problem with fairness constraint β.)

Lemma 6.1. *For any $0 < \epsilon < 1$, the optimal LP solution to* LP-FAIR *with $\tilde{\beta} = (1-\epsilon)\beta$ is at least $\sum_{e\in E} w_e x_e \geq (1-\epsilon)\sum_{e\in E} w_e y_e$, where $\boldsymbol{y} = (y_e)_{e\in E}$ is an optimal LP solution to* LP-FAIR *with β.*

Proof. Any feasible solution to LP-FAIR with $\alpha = 0$ and $\beta(1-\epsilon) < 1$ is also a feasible solution when $\alpha = 0$ and $\beta < 1$. Let $\boldsymbol{x} = (x_e)_{e\in E}$ and $\boldsymbol{y} := (y_e)_{e\in E}$ are optimal fractional solutions to LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$ and $\tilde{\beta} = \beta$ respectively. Therefore,

$$\sum_{e\in E} w_e x_e \geq (1-\epsilon)\sum_{e\in E} w_e y_e. \tag{8}$$

□

Corollary 6.2. *Algorithm 1 returns a matching $\mathcal{M}$ with an expected weight of at least $\frac{1}{2}(1-\epsilon)\mathrm{OPT}$.*

We next argue that the matching $\mathcal{M}$ we output satisfies the β-fairness constraints exactly with a probability of at least $1 - f(\epsilon, G)$ where $f(\epsilon, G) = 2\exp\left(-\frac{\epsilon^2\beta\sum_{e\in E} x_e}{28}\right)$. Note that the failure probability of our algorithm now only depends on the parameter ϵ and $\sum_{e\in E} x_e$ (see Lemma 6.3). It is also reduced by a factor of 2, since we can use a one-sided version of Freedman's inequality when $\alpha = 0$ and $\beta > 0$. Here, $\sum_{e\in E} x_e$ refers to the size of optimal fractional matching of LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$.

Lemma 6.3. *Given an optimal solution $\boldsymbol{x}$ to* LP-FAIR *with $\tilde{\beta} = (1-\epsilon)\beta$, then we can recover a solution that satisfies the constraints exactly i.e., no violations in β-fairness constraints. Formally, we can say that for any color c, and $0 < \epsilon < 1$, the size of the matching restricted to color class c is at most $\beta|\mathcal{M}|$ with high probability,*

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq \beta\right] \geq 1 - 2\exp\left(\frac{-\epsilon^2\beta\sum_{e\in E} x_e}{28}\right).$$

Therefore we can conclude that the probability with which our algorithm fails to satisfy β-fairness constraints is now tightened from $f_c(\delta, G)$ which depends on $\sum_{e\in E_c} x_e$ of individual color classes, to a global quantity $\beta\sum_{e\in E} x_e$ in $f(\epsilon, G)$. This concludes the proof of Theorem 4.3.

6.1 Brute Force Algorithm for Small Instances

Given an instance of PROPORTIONALLYFAIRMATCHING, recall that the performance of Algorithm 1 improves the larger $\sum_{e\in E} \beta x_e$ is, where $\boldsymbol{x} = (x_e)_{e\in E}$ is an optimal solution to LP-FAIR with $\tilde{\beta} = (1-\varepsilon)\beta$.

We now handle the case where $\sum_{e\in E} \beta x_e \leq C$ by running a simple brute-force algorithm. In order to see how to do this, let $\boldsymbol{y} = (y_e)_{e\in E}$ be an optimal solution to LP-FAIR with β. (We don't actually use $\boldsymbol{y}$ in our algorithm, it is just for the analysis). Define $L = \min_{e\in E} w_e$ and $U = \max_{e\in E} w_e$. Then,

$$U\sum_{e\in E} x_e \geq \sum_{e\in E} w_e x_e \geq (1-\varepsilon)\sum_{e\in E} w_e y_e \geq (1-\varepsilon)L\sum_{e\in E} y_e,$$

where the second inequality from the left uses Lemma 6.1, and the rest use the trivial upper and lower bounds of U and L. We thus have that

$$\frac{U}{L(1-\varepsilon)}\sum_{e\in E} x_e \geq \sum_{e\in E} y_e. \tag{9}$$

Thus, if $\sum_{e\in E} \beta x_e \leq C$, then $\sum_{e\in E} y_e \leq \frac{U}{L(1-\varepsilon)}\frac{C}{\beta}$ via (9). Let us now suppose that M_{OPT} is a maximum weighted matching with respect to fairness constraint β. Then, $L|M_{\mathrm{OPT}}| \leq \sum_{e\in M_{\mathrm{OPT}}} w_e \leq \sum_{e\in E} w_e y_e \leq U\sum_{e\in E} y_e \leq U\frac{U}{\beta L}\frac{C}{1-\varepsilon}$, where the last inequality uses the assumed inequality $\sum_{e\in E} y_e \leq \frac{U}{\beta L}\frac{C}{1-\varepsilon}$. It follows that

$$|M_{OPT}| \leq \frac{U^2}{L^2}\frac{C}{\beta(1-\varepsilon)}. \tag{10}$$

Thus, if we run a brute force algorithm where we check matchings which are fair with respect to β, and which contain at most $\frac{U^2}{L^2}\frac{C}{\beta(1-\varepsilon)}$ edges, then (10) ensures that we'll return an optimal matching.

7 Conclusion and Future Work

In contrast to our current algorithm which processes vertices in a fixed or arbitrary order, Random Order Contention Resolution Scheme (RCRS) from (Adamczyk and Włodarczyk 2018) provides better selection guarantees where vertices are processed in a random order. More precisely, RCRS achieves an approximation ratio of $1 - \frac{1}{e}$. However, it appears to be much more challenging to compute the one-step variance of the martingale at each time t, making it difficult to derive a *useful bound* for the sum of variances $\sum_{t\in[n]} \mathrm{Var}(|\Delta M_t| \mid \mathcal{H}_{t-1})$.

Thus, it remains an interesting open problem to determine if we can overcome this obstacle to improve the approximation factor from $1/2$ to $1 - \frac{1}{e}$. Moreover, adapting the analysis to general graphs is also an interesting future direction.

Acknowledgments

Sharmila Duppala, Nathaniel Grammel, Juan Luque, and Aravind Srinivasan were supported in part by NSF award number CCF-1918749.

References

Adamczyk, M.; and Włodarczyk, M. 2018. Random order contention resolution schemes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 790–801. IEEE.

Bandyapadhyay, S.; Fomin, F. V.; Inamdar, T.; and Simonov, K. 2023. Proportionally Fair Matching with Multiple Groups. *arXiv preprint arXiv:2301.03862*.

Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4): 509–522.

Bera, S.; Chakrabarty, D.; Flores, N.; and Negahbani, M. 2019. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32.

Chekuri, C.; Vondrák, J.; and Zenklusen, R. 2014. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6): 1831–1879.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvtiskii, S. 2019. Matroids, matchings, and fairness. In *The 22nd international conference on artificial intelligence and statistics*, 2212–2220. PMLR.

Choromanski, K. M.; Jebara, T.; and Tang, K. 2013. Adaptive Anonymity via b-Matching. *Advances in Neural Information Processing Systems*, 26.

Confessore, N. 2018. Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. *The New York Times*.

Dickerson, J.; Sankararaman, K.; Srinivasan, A.; and Xu, P. 2018. Allocation Problems in Ride-Sharing Platforms: Online Matching With Offline Reusable Resources. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2013. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, 323–340.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2019. Balancing relevance and diversity in online bipartite matching via submodularity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1877–1884.

Dubhashi, D. P.; and Ranjan, D. 1996. Balls and Bins: A Study in Negative Dependence. *BRICS Report Series*, 3(25).

Duppala, S.; Grammel, N.; Luque, J.; MacRury, C.; and Srinivasan, A. 2024. Proportionally Fair Matching via Randomized Rounding. arXiv:2412.11238.

Duppala, S.; Luque, J.; Dickerson, J.; and Srinivasan, A. 2023. Group fairness in set packing problems. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23. ISBN 978-1-956792-03-4.

Esmaeili, S.; Duppala, S.; Cheng, D.; Nanda, V.; Srinivasan, A.; and Dickerson, J. P. 2023. Rawlsian fairness in online bipartite matching: Two-sided, group, and individual. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5624–5632.

Ezra, T.; Feldman, M.; Gravin, N.; and Tang, Z. G. 2022. Prophet Matching with General Arrivals. *Mathematics of Operations Research*, 47(2): 878–898.

Farhadi, A.; Gilbert, J.; and Hajiaghayi, M. 2022. Generalized Stochastic Matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10008–10015.

Farnadi, G.; St-Arnaud, W.; Babaki, B.; and Carvalho, M. 2021. Individual Fairness in Kidney Exchange Programs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11496–11505.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 259–268. New York, NY, USA: Association for Computing Machinery. ISBN 9781450336642.

Feldman, M.; Svensson, O.; and Zenklusen, R. 2021. Online Contention Resolution Schemes with Applications to Bayesian Selection Problems. *SIAM Journal on Computing*, 50(2): 255–300.

Freedman, D. A. 1975. On tail probabilities for martingales. *the Annals of Probability*, 100–118.

Gandhi, R.; Khuller, S.; Parthasarathy, S.; and Srinivasan, A. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3): 324–360.

García-Soriano, D.; and Bonchi, F. 2020. Fair-by-design matching. *Data Mining and Knowledge Discovery*, 34: 1291–1335.

Hikima, Y.; Akagi, Y.; Kim, H.; Kohjima, M.; Kurashima, T.; and Toda, H. 2021. Integrated Optimization of Bipartite Matching and Its Stochastic Behavior: New Formulation and Approximation Algorithm via Min-cost Flow Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5): 3796–3805.

Ho, C.-J.; and Vaughan, J. 2021. Online Task Assignment in Crowdsourcing Markets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1): 45–51.

Huang, B.; and Jebara, T. 2007. Loopy belief propagation for bipartite maximum weight b-matching. In *Artificial Intelligence and Statistics*, 195–202. PMLR.

Huang, B.; and Jebara, T. 2011. Fast b-matching via sufficient selection belief propagation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 361–369. JMLR Workshop and Conference Proceedings.

Impagliazzo, R.; and Paturi, R. 1999. Complexity of k-SAT. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*, 237–240.

Jebara, T.; Wang, J.; and Chang, S.-F. 2009. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, 441–448.

Kleindessner, M.; Samadi, S.; Awasthi, P.; and Morgenstern, J. 2019. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, 3458–3467. PMLR.

Lewis, P.; and Hilder, P. 2018. Leaked: Cambridge Analytica's blueprint for Trump victory. *The Guardian*.

McElfresh, D. C.; Bidkhori, H.; and Dickerson, J. P. 2019. Scalable Robust Kidney Exchange. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 1077–1084.

Mehta, A. 2013. Online Matching and Ad Allocation. *Foundations and Trends in Theoretical Computer Science*, 8 (4): 265–368.

Mehta, A.; Saberi, A.; Vazirani, U.; and Vazirani, V. 2007. AdWords and Generalized Online Matching. *Journal of the ACM*, 54, no. 5.

Nanda, V.; Xu, P.; Sankararaman, K. A.; Dickerson, J.; and Srinivasan, A. 2020. Balancing the Tradeoff between Profit and Fairness in Rideshare Platforms during High-Demand Hours. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02): 2210–2217.

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; and Cheng, X. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Purohit, M.; Gollapudi, S.; and Raghavan, M. 2019. Hiring Under Uncertainty. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5181–5189. PMLR.

Sankar, G. S.; Louis, A.; Nasre, M.; and Nimbhorkar, P. 2021. Matchings with group fairness constraints: Online and offline algorithms. *arXiv preprint arXiv:2105.09522*.

Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016. Online mobile Micro-Task Allocation in spatial crowdsourcing. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 49–60.

Zaslavskiy, M.; Bach, F.; and Vert, J.-P. 2009. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12): i259–1267.