

Rawlsian Fairness in Online Bipartite Matching: Two-Sided, Group, and Individual

Seyed Esmaeili¹, Sharmila Duppala¹, Davidson Cheng², Vedant Nanda¹, Aravind Srinivasan¹, John P. Dickerson¹

¹ University of Maryland, College Park

² Colorado College

{esmaeili,sduppala,vedant,srin,john}@cs.umd.edu, d.cheng@coloradocollege.edu

Abstract

Online bipartite-matching platforms are ubiquitous and find applications in important areas such as crowdsourcing and ridesharing. In the most general form, the platform consists of three entities: two sides to be matched and a platform operator that decides the matching. The design of algorithms for such platforms has traditionally focused on the operator’s (expected) profit. Since fairness has become an important consideration that was ignored in the existing algorithms a collection of online matching algorithms have been developed that give a fair treatment guarantee for one side of the market at the expense of a drop in the operator’s profit. In this paper, we generalize the existing work to offer fair treatment guarantees to *both* sides of the market *simultaneously*, at a calculated worst case drop to operator profit. We consider group and individual Rawlsian fairness criteria. Moreover, our algorithms have theoretical guarantees and have adjustable parameters that can be tuned as desired to balance the trade-off between the utilities of the three sides. We also derive hardness results that give clear upper bounds over the performance of any algorithm.

1 Introduction

Online bipartite matching has been used to model many important applications such as crowdsourcing (Ho and Vaughan 2012; Tong et al. 2016; Dickerson et al. 2019), rideshare (Lowalekar, Varakantham, and Jaillet 2018; Dickerson et al. 2021; Ma, Xu, and Xu 2021), and online ad allocation (Goel and Mehta 2008; Mehta 2013). In the most general version of the problem, there are three interacting entities: two sides of the market to be matched and a platform operator which assigns the matches. For example, in rideshare, riders on one side of the market submit requests, drivers on the other side of the market can take requests, and a platform operator such as Uber or Lyft matches the riders’ requests to one or more available drivers. In the case of crowdsourcing, organizations offer tasks, workers look for tasks to complete, and a platform operator such as Amazon Mechanical Turk (MTurk) or Upwork matches tasks to workers.

Online bipartite matching algorithms are often designed to optimize a performance measure—usually, maximizing

overall profit for the platform operator or a proxy of that objective. However, fairness considerations were largely ignored. This is troubling especially given that recent reports have indicated that different demographic groups may not receive similar treatment. For example, in rideshare platforms once the platform assigns a driver to a rider’s request, both the rider and the driver have the option of rejecting the assignment and it has been observed that membership in a demographic group may cause adverse treatment in the form of higher rejection. Indeed, (Cook 2018; White 2016; Wirtschafter 2019) report that drivers could reject riders based on attributes such as gender, race, or disability. Conversely, (Rosenblat et al. 2016) reports that drivers are likely to receive less favorable ratings if they belong to certain demographic groups. A similar phenomenon exists in crowdsourcing (Galperin and Greppi 2017). Moreover, even in the absence of such evidence of discrimination, as algorithms become more prevalent in making decisions that directly affect the welfare of individuals (Barocas, Hardt, and Narayanan 2019; Dwork et al. 2012), it becomes important to guarantee a standard of fairness. Also, while much of our discussion focuses on the for-profit setting for concreteness, similar fairness issues hold in not-for-profit scenarios such as the fair matching of individuals with health-care facilities, e.g., in the time of a pandemic.

In response, a recent line of research has been concerned with the issue of designing fair algorithms for online bipartite matching. (Lesmana, Zhang, and Bei 2019; Ma and Xu 2022; Xu and Xu 2020) present algorithms which give a minimum utility guarantee for the drivers at a bounded drop to the operator’s profit. Conversely, (Nanda et al. 2020) give guarantees for both the platform operator and the riders instead. Finally, (Sühr et al. 2019) shows empirical methods that achieve fairness for both the riders and drivers simultaneously but lacks theoretical guarantees and ignores the operator’s profit.

Nevertheless, the existing work has a major drawback in terms of optimality guarantees. Specifically, the two sides being matched along with the platform operator constitute the three main interacting entities in online matching and despite the significant progress in fair online matching none of the previous work considers all three sides simultaneously. In this paper, we derive algorithms with theoretical guarantees for the platform operator’s profit as well as fairness

guarantees for the two sides of the market. Unlike the previous work we not only consider the size of the matching but also its quality. Further, we consider two online arrival settings: the **KIID** and the richer **KAD** setting (see Section 3 for definitions). We consider both group and individual notions of Rawlsian fairness and interestingly show a reduction from individual fairness to group fairness in the **KAD** setting. Moreover, we show upper bounds on the optimality guarantees of any algorithm and derive impossibility results that show a conflict between group and individual notions of fairness. Finally, we empirically test our algorithms on a real-world dataset.

2 Related Work

It is worth noting that similar to our work, (Patro et al. 2020) and (Basu et al. 2020) have considered two-sided fairness as well, although in the setting of recommendation systems where a different model is applied—and, critically, a separate objective for the operator’s profit was not considered.

Fairness in bipartite matching has seen significant interest recently. The fairness definition employed has consistently been the well-known Rawlsian fairness (Rawls 1958) (i.e. max-min fairness) or its generalization Leximin fairness.* We note that the objective to be maximized (other than the fairness objective) represents operator profit in our setting.

The case of offline and unweighted maximum cardinality matching is addressed by (García-Soriano and Bonchi 2020), who give an algorithm with Leximin fairness guarantees for one side of the market (one side of the bipartite graph) and show that this can be achieved without sacrificing the size of the match. Motivated by fairness consideration for drivers in ridesharing, (Lesmana, Zhang, and Bei 2019) considers the problem of offline and weighted matching. Specifically, they show an algorithm with a provable trade-off between the operator’s profit and the minimum utility guaranteed to any vertex in one-side of the market.

Recently, (Ma, Xu, and Xu 2020) considered fairness for the online part of the graph through a group notion of fairness. In particular, the utility for a group is added across the different types and is minimized for the group worst off, in rough terms their notion translates to maximizing the minimum utility accumulated by a group throughout the matching. Their notion of fairness is very similar to the one we consider here. However, (Ma, Xu, and Xu 2020) considers fairness only on one side of the graph and ignores the operator’s profit. Further, only the matching size is considered to measure utility, i.e. edges are unweighted.

A new notion of group fairness in online matching is considered in (Sankar et al. 2021). In rough terms, their group fairness criterion amounts to establishing a quota for each group and ensuring that the matching does not exceed that quota. This notion can be seen as ensuring that the system is not dominated by a specific group and is in some sense an opposite to max-min fairness as the utility is upper bounded

*Leximin fairness maximizes the minimum utility like max-min fairness. However, it proceeds to maximize the second worst utility, and so on until the list is exhausted.

instead of being lower bounded. Further, the fairness guarantees considered are one-sided as well.

On the empirical side of fair online matching, (Mattei, Saffidine, and Walsh 2017) and (Lee et al. 2019) give application-specific treatments in the context of deceased-donor organ allocation and food bank provisioning, respectively. More related to our work is that of (Sühr et al. 2019; Zhou, Marecek, and Shorten 2021) which consider the rideshare problem and provide algorithms to achieve fairness for both sides of the graph simultaneously, however both papers lack theoretical guarantees and in the case of (Sühr et al. 2019) the operator’s profit is not considered.

3 Online Model & Optimization Objectives

Our model follows that of (Mehta 2013; Feldman et al. 2009; Bansal et al. 2010; Alaei, Hajiaghayi, and Liaghat 2013) and others. We have a bipartite graph $G = (U, V, E)$ where U represents the set of static (offline) vertices (workers) and V represents the set of online vertex types (job types) which arrive dynamically in each round. The online matching is done over T rounds. In a given round t , a vertex of type v is sampled from V with probability $p_{v,t}$ with $\sum_{v \in V} p_{v,t} = 1, \forall t \in [T]$ the probability $p_{v,t}$ is known beforehand for each type v and each round t . This arrival setting is referred to as the known adversarial distribution (**KAD**) setting (Alaei, Hajiaghayi, and Liaghat 2013; Dickerson et al. 2021). When the distribution is stationary, i.e. $p_{v,t} = p_v, \forall t \in [T]$, we have the arrival setting of the known independent identical distribution (**KIID**). Accordingly, the expected number of arrivals of type v in T rounds is $n_v = \sum_{t \in [T]} p_{v,t}$, which reduces to $n_v = Tp_v$ in the **KIID** setting. We assume that $n_v \in \mathbb{Z}^+$ for **KIID** (Bansal et al. 2010). Every vertex u (v) has a group membership,[†] with \mathcal{G} being the set of all group memberships; for any vertex $u \in U$, we denote its group memberships by $g(u) \in \mathcal{G}$ (similarly, we have $g(v)$ for $v \in V$). Conversely, for a group g , $U(g)$ ($V(g)$) denotes the subset of U (V) with group membership g . A vertex u (v) has a set of incident edges E_u (E_v) which connect it to vertices in the opposite side of the graph. In a given round, once a vertex (job) v arrives, an irrevocable decision has to be made on whether to reject v or assign it to a neighbouring vertex u (where $(u, v) \in E_v$) which has not been matched before. Suppose, that v is assigned to u , then the assignment is not necessarily successful rather it succeeds with probability $p_e = p_{(u,v)} \in [0, 1]$. This models the fact that an assignment could fail for some reason such as the worker refusing the assigned job. Furthermore, each vertex u has patience parameter $\Delta_u \in \mathbb{Z}^+$ which indicates the number of failed assignments it can tolerate before leaving the system, i.e. if u receives Δ_u failed assignments then it is deleted from the graph. Similarly, a vertex v has patience $\Delta_v \in \mathbb{Z}^+$, if a vertex v arrives in a given round, then it would tolerate at most Δ_v many failed assignments in that round before leaving the system.

[†]For a clearer representation we assume each vertex belongs to one group although our algorithms apply to the case where a vertex can belong to multiple groups.

For a given edge $e = (u, v) \in E$, we let each entity assign its own utility to that edge. In particular, the platform operator assigns a utility of w_e^O , whereas the offline vertex u assigns a utility of w_e^U , and the online vertex v assigns a utility of w_e^V . This captures entities' heterogeneous wants. For example, in ridesharing, drivers may desire long trips from nearby riders, whereas the platform operator would not be concerned with the driver's proximity to the rider, although this maybe the only consideration the rider has. Similar motivations exist in crowdsourcing as well. We finally note that most of the details of our model such the **KIID** and **KAD** arrival settings as well as the vertex patience follow well-established and practically motivated model choices in online matching, see Appendix (??) for more details.

Letting \mathcal{M} denote the set of successful matchings made in the T rounds, then we consider the following optimization objectives:

- **Operator's Utility (Profit):** The operator's expected profit is simply the expected sums of the profits across the matched edges, this leads to $\mathbb{E}[\sum_{e \in \mathcal{M}} w_e^O]$.
- **Rawlsian Group Fairness:**
 - **Offline Side:** Denote by \mathcal{M}_u the subset of edges in the matching that are incident on u . Then our fairness criterion is equal to

$$\min_{g \in \mathcal{G}} \frac{\mathbb{E}[\sum_{u \in U(g)} (\sum_{e \in \mathcal{M}_u} w_e^U)]}{|U(g)|}.$$

this value equals the minimum average expected utility received by a group in the offline side U .

- **Online Side:** Similarly, we denote by \mathcal{M}_v the subset of edges in the matching that are incident on vertex v , and define the fairness criterion to be

$$\min_{g \in \mathcal{G}} \frac{\mathbb{E}[\sum_{v \in V(g)} (\sum_{e \in \mathcal{M}_v} w_e^V)]}{\sum_{v \in V(g)} n_v}.$$

this value equals the minimum average expected utility received throughout the matching by any group in the online side V .

- **Rawlsian Individual Fairness:**

- **Offline Side:** The definition here follows from the group fairness definition for the offline side by simply considering that each vertex u belongs to its own distinct group. Therefore, the objective is $\min_{u \in U} \mathbb{E}[\sum_{e \in \mathcal{M}_u} w_e^U]$.
- **Online Side:** Unlike the offline side, the definition does not follow as straightforwardly. Here we cannot obtain a valid definition by simply assigning each vertex type its own group. Rather, we note that a given individual is actually a given arriving vertex at a given round $t \in [T]$, accordingly our fairness criterion is the minimum expected utility an individual receives in a given round, i.e. $\min_{t \in [T]} \mathbb{E}[\sum_{e \in \mathcal{M}_{v_t}} w_e^V]$, where v_t is the vertex that arrived in round t .

4 Main Results

Performance Criterion: We note that we are in the online setting, therefore our performance criterion is the competitive ratio. Denote by \mathcal{I} the distribution for the instances of matching problems, then $\text{OPT}(\mathcal{I}) = \mathbb{E}_{I \sim \mathcal{I}}[\text{OPT}(I)]$ where $\text{OPT}(I)$ is the optimal value of the sampled instance I . Similarly, for a given algorithm ALG , we define the value of its objective over the distribution \mathcal{I} by $\text{ALG}(\mathcal{I}) = \mathbb{E}_{\mathcal{D}}[\text{ALG}(I)]$ where the expectation $\mathbb{E}_{\mathcal{D}}[\cdot]$ is over the randomness of the instance and the algorithm. The competitive ratio is then defined as $\min_{\mathcal{I}} \frac{\text{ALG}(\mathcal{I})}{\text{OPT}(\mathcal{I})}$.

In our work, we address optimality guarantees for each of the three sides of the matching market by providing algorithms with competitive ratio guarantees for the operator's profit and the fairness objectives of the static and online side of the market simultaneously. Specifically, for the **KIID** arrival setting we have:

Theorem 4.1. *For the **KIID** setting, algorithm $\text{TSGF}_{\text{KIID}}(\alpha, \beta, \gamma)$ achieves a competitive ratio of $(\frac{\alpha}{2e}, \frac{\beta}{2e}, \frac{\gamma}{2e})^\dagger$ simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where $\alpha, \beta, \gamma > 0$ and $\alpha + \beta + \gamma \leq 1$.*

The following two theorems hold under the condition that $p_e = 1, \forall e \in E$. Specifically for the **KAD** setting we have:

Theorem 4.2. *For the **KAD** setting, algorithm $\text{TSGF}_{\text{KAD}}(\alpha, \beta, \gamma)$ achieves a competitive ratio of $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where $\alpha, \beta, \gamma > 0$ and $\alpha + \beta + \gamma \leq 1$.*

Moreover, for the case of individual fairness whether in the **KIID** or **KAD** arrival setting we have:

Theorem 4.3. *For the **KIID** or **KAD** setting, we can achieve a competitive ratio of $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ simultaneously over the operator's profit, the individual fairness objective for the offline side, and the individual fairness objective for the online side, where $\alpha, \beta, \gamma > 0$ and $\alpha + \beta + \gamma \leq 1$.*

We also give the following hardness results. In particular, for a given arrival (**KIID** or **KAD**) setting and fairness criterion (group or individual), the competitive ratios for all sides cannot exceed 1 simultaneously:

Theorem 4.4. *For all arrival models, given the three objectives: operator's profit, offline side group (individual) fairness, and online side group (individual) fairness. No algorithm can achieve a competitive ratio of (α, β, γ) over the three objectives simultaneously such that $\alpha + \beta + \gamma > 1$.*

It is natural to wonder if we can combine individual and group fairness. Though it is possible to extend our algorithms to this setting. The follow theorem shows that they can conflict with one another:

Theorem 4.5. *Ignoring the operator's profit and focusing either on the offline side alone or the online side alone.*

[†]Here, e denotes the Euler number, not an edge in the graph.

With α_G and α_I denoting the group and individual fairness competitive ratios, respectively. No algorithm can achieve competitive ratios (α_G, α_I) over the group and individual fairness objectives of one side simultaneously such that $\alpha_G + \alpha_I > 1$.

Finally, we carry experiments on real-world datasets in Section 6.

5 Algorithms and Theoretical Guarantees

Our algorithms use linear programming (LP) based techniques (Bansal et al. 2010; Nanda et al. 2020; Xu and Xu 2020; Brubach et al. 2016) where first a *benchmark* LP is set up to upper bound the optimal value of the problem, then an LP solution is sampled from to produce an algorithm with guarantees. Due to space constraints, all proofs and the technical details of Theorems (4.4 and 4.5) are in Appendix (??).

5.1 Group Fairness for the KIID Setting:

Before we discuss the details of the algorithm, we note that for a given vertex type $v \in V$, the expected arrival rate n_v could be greater than one. However, it is not difficult to modify the instance by “fragmenting” each type with $n_v > 1$ such that in the new instance $n_v = 1$ for each type. This can be done with the operator’s profit, offline group fairness, and online group fairness having the same values. Therefore, in what remains for the **KIID** setting $n_v = 1, \forall v \in V$ and therefore for any round t , each vertex v arrives with probability $\frac{1}{T}$. It also follows that for a given group g , $\sum_{v \in V(g)} n_v = \sum_{v \in V(g)} 1 = |V(g)|$.

For each edge $e = (u, v) \in E$ we use x_e to denote the expected number of probes (i.e., assignments from u to type v not necessarily successful) made to edge e in the LP benchmark. We have a total of three LPs each having the same set of constraints of (4), but differing by the objective. For compactness we do not repeat these constraints and instead write them once. Specifically, LP objective (1) along with the constraints of (4) give the optimal benchmark value of the operator’s profit. Similarly, with the same set of constraints (4) LP objective (2) and LP objective (3) give the optimal group max-min fair assignment for the offline and online sides, respectively. Note that the expected max-min objectives of (2) and (3), can be written in the form of a linear objective. For example, the max-min objective of (2) can be replaced with an LP with objective $\max \eta$ subject to the additional constraints that $\forall g \in \mathcal{G}, \eta \leq \frac{\sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_e p_e}{|U(g)|}$. Having introduced the LPs, we will use LP(1), LP(2), and LP(3) to refer to the platform’s profit LP, the offline side group fairness LP, and the online side group fairness LP, respectively.

$$\max \sum_{e \in E} w_e^O x_e p_e \quad (1)$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_e p_e}{|U(g)|} \quad (2)$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{v \in V(g)} \sum_{e \in E_v} w_e^V x_e p_e}{|V(g)|} \quad (3)$$

$$\text{s.t. } \forall e \in E : 0 \leq x_e \leq 1 \quad (4a)$$

$$\forall u \in U : \sum_{e \in E_u} x_e p_e \leq 1 \quad (4b)$$

$$\forall u \in U : \sum_{e \in E_u} x_e \leq \Delta_u \quad (4c)$$

$$\forall v \in V : \sum_{e \in E_v} x_e p_e \leq 1 \quad (4d)$$

$$\forall v \in V : \sum_{e \in E_v} x_e \leq \Delta_v \quad (4e)$$

Now we prove that LP(1), LP(2) and LP(3) indeed provide valid upper bounds (benchmarks) for the optimal solution for the operator’s profit and expected max-min fairness for the offline and online sides of the matching.

Lemma 5.1. *For the KIID setting, the optimal solutions of LP (1), LP (2), and LP (3) are upper bounds on the expected optimal that can be achieved by any algorithm for the operator’s profit, the offline side group fairness objective, and the online side group fairness objective, respectively.*

Our algorithm makes use of the dependent rounding subroutine (Gandhi et al. 2006). We mention the main properties of dependent rounding. In particular, given a fractional vector $\vec{x} = (x_1, x_2, \dots, x_t)$ where each $x_i \in [0, 1]$, let $k = \sum_{i \in [t]} x_i$, dependent rounding rounds x_i (possibly fractional) to $X_i \in \{0, 1\}$ for each $i \in [t]$ such that the resulting vector $\vec{X} = (X_1, X_2, X_3, \dots, X_t)$ has the following properties: (1) **Marginal Distribution:** The probability that $X_i = 1$ is equal to x_i , i.e. $Pr[X_i = 1] = x_i$ for each $i \in [t]$. (2) **Degree Preservation:** Sum of X_i ’s should be equal to either $\lfloor k \rfloor$ or $\lceil k \rceil$ with probability one, i.e. $Pr[\sum_{i \in [t]} X_i \in \{\lfloor k \rfloor, \lceil k \rceil\}] = 1$. (3) **Negative Correlation:** For any $S \subseteq [t]$, (1) $Pr[\wedge_{i \in S} X_i = 0] \leq \prod_{i \in S} Pr[X_i = 0]$ (2) $Pr[\wedge_{i \in S} X_i = 1] \leq \prod_{i \in S} Pr[X_i = 1]$. It follows that for any $x_i, x_j \in \vec{x}$, $E[X_i = 1 | X_j = 1] \leq x_i$.

Going back to the LPs (1,2,3), we denote the optimal solutions to LP (1), LP (2), and LP (3) by \vec{x}^*, \vec{y}^* and \vec{z}^* respectively. Further, we introduce the parameters $\alpha, \beta, \gamma \in [0, 1]$ where $\alpha + \beta + \gamma \leq 1$ and each of these parameters decide the “weight” the algorithm places on each objective (the operator’s profit, the offline group fairness, and the online group fairness objectives). We note that our algorithm makes use of the subroutine **PPDR** (Probe with Permuted Dependent Rounding) shown in Algorithm 1.

Algorithm 1: **PPDR**(\vec{x}_v)

- 1: Apply dependent rounding to the fractional solution \vec{x}_v to get a binary vector \vec{X}_v .
 - 2: Choose a random permutation π over the set E_v .
 - 3: **for** $i = 1$ to $|E_v|$ **do**
 - 4: Probe vertex $\pi(i)$ if it is available and $\vec{X}_v(\pi(i)) = 1$
 - 5: **if** Probe is successful (i.e., a match) **then**
 - 6: **break**
-

The procedure of our parameterized sampling algorithm **TSGF_{KIID}** is shown in Algorithm 2. Specifically, when a vertex of type v arrives at any time step we run **PPDR**(\vec{x}_v^*), **PPDR**(\vec{y}_v^*), or **PPDR**(\vec{z}_v^*) with probabilities α, β , and γ , respectively. We do not run any of the **PPDR** subroutines and instead reject the vertex with probability $1 - (\alpha + \beta + \gamma)$. The LP constraint (4e) guarantees that $\forall v \in V : \sum_{e \in E_v} s_e^* \leq$

Δ_v where s^* could be \vec{x}^* , \vec{y}^* , or \vec{z}^* . Therefore, when **PPDR** is invoked by the **degree preservation** property of dependent rounding the number of edges probed will not exceed Δ_v , i.e. it would be within the patience limit.

Algorithm 2: TSGF_{KIID}(α, β, γ)

- 1: Let v be the vertex type arriving at time t .
 - 2: With probability α run the subroutine, **PPDR**(\vec{x}_v^*).
 - 3: With probability β run the subroutine, **PPDR**(\vec{y}_v^*).
 - 4: With probability γ run the subroutine, **PPDR**(\vec{z}_v^*).
 - 5: Reject the arriving vertex with probability $1 - (\alpha + \beta + \gamma)$.
-

Now we analyze TSGF_{KIID} to prove Theorem 4.1. It would suffice to prove that for each edge e the expected number of successful probes is at least $\alpha \frac{x_e^*}{2e}$, $\beta \frac{y_e^*}{2e}$ and $\gamma \frac{z_e^*}{2e}$. And finally from the linearity of expectation we show that the worst case competitive ratio of the proposed online algorithm with parameters α, β and γ is at least $(\frac{\alpha}{2e}, \frac{\beta}{2e}, \frac{\gamma}{2e})$ for the operator's profit and group fairness objectives on the offline and online sides of the matching, respectively.

A critical step is to lower bound the probability that a vertex u is available (safe) at the beginning of round $t \in [T]$. Let us denote the indicator random variable for that event by $SF_{u,t}$. The following lemma enables us to lower bound for the probability of $SF_{u,t}$.

Lemma 5.2. $Pr[SF_{u,t}] \geq \left(1 - \frac{t-1}{T}\right) \left(1 - \frac{1}{T}\right)^{t-1}$.

Now that we have established a lower bound on $Pr[SF_{u,t}]$, we lower bound the probability that an edge e is probed by one of the **PPDR** subroutines conditioned on the fact that u is available (Lemma 5.3). Let $1_{e,t}$ be the indicator that $e = (u, v)$ is probed by the TSGF_{KIID} Algorithm at time t . Note that event $1_{e,t}$ occurs when (1) a vertex of type v arrives at time t and (2) e is sampled by **PPDR**(\vec{x}_v^*), **PPDR**(\vec{y}_v^*), or **PPDR**(\vec{z}_v^*).

Lemma 5.3. $Pr[1_{e,t} \mid SF_{u,t}] \geq \alpha \frac{x_{e,t}^*}{2T}, Pr[1_{e,t} \mid SF_{u,t}] \geq \beta \frac{y_{e,t}^*}{2T}, Pr[1_{e,t} \mid SF_{u,t}] \geq \gamma \frac{z_{e,t}^*}{2T}$

Given the above lemmas Theorem 4.1 can be proved.

5.2 Group Fairness for the KAD Setting:

For the **KAD** setting, the distribution over V is time dependent and hence the probability of sampling a type v in round t is $p_{v,t} \in [0, 1]$ with $\sum_{v \in V} p_{v,t} = 1$. Further, we assume for the **KAD** setting that for every edge $e \in E$ we have $p_e = 1$. This means that whenever an incoming vertex v is assigned to a safe-to-add vertex u the assignment is successful. This also means that any non-trivial values for the patience parameters Δ_u and Δ_v become meaningless and hence we can WLOG assume that $\forall u \in U, \forall v \in V, \Delta_u = \Delta_v = 1$. From the above discussion, we have the following LP benchmarks for the operator's profit, the group fairness for the offline side and the group fairness for the online side:

$$\max \sum_{t \in [T]} \sum_{e \in E} w_e^O x_{e,t} \quad (5)$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{t \in [T]} \sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_{e,t}}{|U(g)|} \quad (6)$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{t \in [T]} \sum_{v \in V(g)} \sum_{e \in E_v} w_e^V x_{e,t}}{\sum_{v \in V(g)} n_v} \quad (7)$$

$$\text{s.t } \forall e \in E, \forall t \in [T] : 0 \leq x_{e,t} \leq 1 \quad (8a)$$

$$\forall u \in U : \sum_{t \in [T]} \sum_{e \in E_u} x_{e,t} \leq 1 \quad (8b)$$

$$\forall v \in V, \forall t \in [T] : \sum_{e \in E_v} x_{e,t} \leq p_{v,t} \quad (8c)$$

Lemma 5.4. For the **KAD** setting, the optimal solutions of LP (5), LP (6) and LP (7) are upper bounds on the expected optimal that can be achieved by any algorithm for the operator's profit, the offline side group fairness objective, and the online side group fairness objective, respectively.

Note that in the above LP we have $x_{e,t}$ as the probability for successfully assigning an edge in round t (with an explicit dependence on t), unlike in the **KIID** setting where we had x_e instead to denote the expected number of times edge e is probed through all rounds. Similar to our solution for the **KIID** setting, we denote by $x_{e,t}^*$, $y_{e,t}^*$, and $z_{e,t}^*$ the optimal solutions of the LP benchmarks for the operator's profit, offline side group fairness, and online side group fairness, respectively.

Having the optimal solutions to the LPs, we use algorithm TSGF_{KAD} shown in Algorithm 3. In TSGF_{KAD} new parameters are introduced, specifically λ and $\rho_{e,t}$ where $\rho_{e,t}$ is the probability that edge $e = (u, v)$ is safe to add in round t , i.e. the probability that u is unmatched at the beginning of round t . For now we assume that we have the precise values of $\rho_{e,t}$ for all rounds and discuss how to obtain these values at the end of this subsection. Now conditioned on v arriving at round t and $e = (u, v)$ being safe to add, it follows that e is sampled with probability $\alpha \frac{x_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}} + \beta \frac{y_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}} + \gamma \frac{z_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}}$ which would be a valid probability (positive and not exceeding 1) if $\rho_{e,t} \geq \lambda$. This follows from the fact that $\alpha, \beta, \gamma \in [0, 1]$ and $\alpha + \beta + \gamma \leq 1$ and also by constraint (8c) which leads to $\frac{\sum_{e \in E_v} x_{e,t}}{p_{v,t}} \leq 1$. Further, if $\rho_{e,t} \geq \lambda$ then by constraint (8c) we have $\sum_{e \in E_v} \left(\alpha \frac{x_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}} + \beta \frac{y_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}} + \gamma \frac{z_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}} \right) \leq 1$ and therefore the distribution is valid. Clearly, the value of λ is important for the validity of the algorithm, the following lemma shows that $\lambda = \frac{1}{2}$ leads to a valid algorithm.

Lemma 5.5. Algorithm TSGF_{KAD} is valid for $\lambda = \frac{1}{2}$.

We now return to the issue of how to obtain the values of $\rho_{e,t}$ for all rounds. This can be done by using the simulation technique as done in (Dickerson et al. 2021; Adamczyk, Grandoni, and Mukherjee 2015). To elaborate, we note that we first solve the LPs (5,6,7) and hence have the values of $x_{e,t}^*$, $y_{e,t}^*$, and $z_{e,t}^*$. Now, for the first round $t = 1$, clearly $\rho_{e,t} = 1, \forall e \in E$. To obtain $\rho_{e,t}$ for $t = 2$, we simulate the arrivals and algorithm a collection of times, and use the empirically estimated probability. More precisely, for $t = 1$

Algorithm 3: TSGF_{KAD}(α, β, γ)

- 1: Let v be the vertex type arriving at time t .
 - 2: **if** $E_{v,t} = \phi$ **then**
 - 3: Reject v
 - 4: **else**
 - 5: With probability α probe e with probability $\frac{x_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}}$.
 - 6: With probability β probe e with probability $\frac{y_{e,t} \lambda}{p_{v,t} \rho_{e,t}}$.
 - 7: With probability γ probe e with probability $\frac{z_{e,t}^* \lambda}{p_{v,t} \rho_{e,t}}$.
 - 8: With probability $[1 - (\alpha + \beta + \gamma)]$ reject v .
-

we sample the arrival of vertex v from $p_{v,t}$ with $t = 1$ ($p_{v,t}$ values are given as part of the model), then we run our algorithm for the values of α, β, γ that we have chosen. Accordingly, at $t = 2$ some vertex in U might be matched. We do this simulation a number of times and then we take $\rho_{e,t}$ for $t = 2$ to be the average of all runs. Now having the values of $\rho_{e,t}$ for $t = 1$ and $t = 2$, we further simulate the arrivals and the algorithm to obtain $\rho_{e,t}$ for $t = 3$ and so on until we get $\rho_{e,t}$ for the last round T . We note that using the Chernoff bound (Mitzenmacher and Upfal 2017) we can rigorously characterize the error in this estimation, however by doing this simulation a number of times that is polynomial in the problem size, the error in the estimation would only affect the lower order terms in the competitive ration analysis (Dickerson et al. 2021) and hence for simplicity it is ignored. Now, with Lemma 5.5 Theorem 4.2 can be proved (see Appendix (??)).

5.3 Individual Fairness KIID and KAD Settings:

For the case of Rawlsian (max-min) individual fairness, we consider each vertex of the offline side to belong to its own distinct group and the definition of group max-min fairness would lead to individual max-min fairness. On the other hand, for the online side a similar trick would not yield a meaningful criterion, we instead define the individual max-min fairness for the online side to equal $\min_{t \in [T]} \mathbb{E}[\text{util}(v_t)] =$

$\min_{t \in [T]} \mathbb{E}[\sum_{e \in \mathcal{M}_{v_t}} w_e^V]$ where $\text{util}(v_t)$ is the utility received by the vertex arriving in round t . If we were to denote by $x_{e,t}$ the probability that the algorithm would successfully match e in round t , then it follows straightforwardly that $\mathbb{E}[\text{util}(v_t)] = \sum_{e \in E_{v_t}} w_e^V x_{e,t}$. We consider this definition to be the valid extension of max-min fairness for the online side as we are now concerned with the minimum utility across the online individuals (arriving vertices) which are T many. The following lemma shows that we can solve two-sided individual max-min fairness by a reduction to two-sided group max-min fairness in the **KAD** arrival setting:

Lemma 5.6. *Whether in the **KIID** or **KAD** setting, a given instance of two-sided individual max-min fairness can be converted to an instance of two-sided group max-min fairness in the **KAD** setting.*

The above Lemma with algorithm TSGF_{KAD} can be used

to prove Theorem 4.3 as shown in Appendix (??).

6 Experiments

In this section, we verify the performance of our algorithm and our theoretical lower bounds for the KIID and group fairness setting using algorithm TSGF_{KIID} (Section 5.1). We note that none of the previous work consider our three-sided setting. We use rideshare as an application example of online bipartite matching (see also, e.g., Dickerson et al. 2021; Nanda et al. 2020; Xu and Xu 2020; Barann, Beverungen, and Müller 2017). We expect similar results and performance to hold in other matching applications such as crowd-sourcing.

Experimental Setup: As done in previous work, the drivers' side is the offline (static) side whereas the riders' side is the online side. We run our experiments over the widely used New York City (NYC) yellow cabs dataset (Sekulić, Long, and Demšar 2021; Nanda et al. 2020; Xu and Xu 2020; Alonso-Mora, Wallar, and Rus 2017) which contains records of taxi trips in the NYC area from 2013. Each record contains a unique (anonymized) ID of the driver, the coordinates of start and end locations of the trip, distance of the trip, and additional metadata.

Similar to (Dickerson et al. 2021; Nanda et al. 2020), we bin the starting and ending latitudes and longitudes by dividing the latitudes from 40.4° to 40.95° and longitudes from -73° to -75° into equally spaced grids of step size 0.005. This enables us to define each driver and request type based on its starting and ending bins. We pick out the trips between 7pm and 8pm on January 31, 2013, which is a rush hour with 10,814 drivers and 35,109 trips. We set driver patience Δ_u to 3. Following (Xu and Xu 2020), we uniformly sample rider patience Δ_v from $\{1, 2\}$.

Since the dataset does not include demographic information, for each vertex we randomly sample the group membership (Nanda et al. 2020). Specifically, we randomly assign 70% of the riders and drivers to be advantaged and the rest to be disadvantaged. The value of p_e for $e = (u, v)$ depends on whether the vertices belong to the advantaged or disadvantaged group. Specifically, $p_e = 0.6$ if both vertices are advantaged, $p_e = 0.3$ if both are disadvantaged, and $p_e = 0.1$ for other cases.

In addition to this, a key component of our work is the use of driver and rider specific utilities. We follow the work of (Sühr et al. 2019) to set the utilities. We adopt the Manhattan distance metric rather than the Euclidean distance metric since the former is a better proxy for length of taxi trips in New York City. We set the operator's utility to the rider's trip length $w_e^O = \text{tripLength}(v)$ —a rough proxy for profit. In addition, the rider's utility over an edge $e = (u, v)$ is set to $w_e^V = -\text{dist}(u, v)$ where $\text{dist}(u, v)$ is the distance between the rider and the driver. The driver's utility is set to $w_e^U = \text{tripLength}(v) - \text{dist}(u, v)$. Whereas the trip length $\text{tripLength}(v)$ is available in the dataset, the distance between the rider and the driver $\text{dist}(u, v)$ is not. We therefore simulate the distance, by creating an equally spaced grid with step size 0.005 around the starting coordinates of the trip. This results in 81 possible coordinates in the vicin-

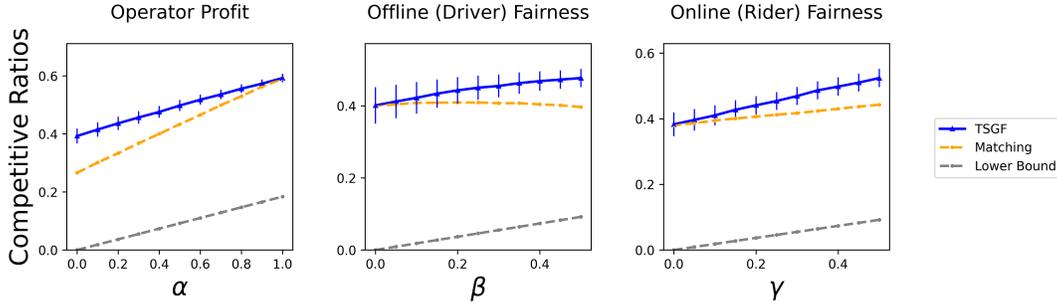


Figure 1: Competitive ratios for TSGF_{KIID} over the operator’s profit, offline (driver) fairness objective, and online (rider) fairness objective with different values of α, β, γ . Note that “Matching” refers to the case where driver and rider utilities are set to 1 across all edges. The experiment is run with $\alpha = \{0, 0.1, 0.2, \dots, 1\}$, and $\beta = \gamma = \frac{1-\alpha}{2}$. Higher competitive ratio indicates better performance.

ity of the starting coordinates of the trip. We then randomly choose one of these 81 coordinates to be the location of the driver when the trip was requested. Then $dist(u, v)$ is the distance between this coordinate to the start coordinate of the trip. This is a valid approximation since the platform would not assign drivers unreasonably far away to pickup a rider. Lastly, we scale the utilities by a constant to prevent them from being negative.

We run TSGF_{KIID} at the scale of $|U| = 49, |V| = 172$ for 100 trials. During each trial, we randomly sample 49 drivers and 172 requests between 7 and 8pm, and run TSGF_{KIID} 100 times to measure the expected competitive ratios of this trial. We then averaged the competitive ratios over all trials, and the results are reported in figure 1. Code to reproduce our experiments is available in the blinded format[‡]; we will release that code in deblinded form upon acceptance.

Performance of TSGF_{KIID} with Varied Parameters:

Figure 1 shows the performance of our algorithm over the three objectives: operator’s profit, offline (driver) group fairness, and online (rider) group fairness. It is clear that the algorithm behaves as expected with all objectives being steadily above their theoretical lower bound. More importantly, we see that increasing the weight for an objective leads to better performance for that objective. I.e., a higher weight for β leads to better performance for the offline side fairness and similar observations follow in the case of α for the operator’s objective and in the case of γ for the online-fairness. This also indicates the limitation in previous work which only considered fairness for one-side since their algorithms would not be able to improve the fairness for the other ignored side.

Furthermore, previous work (e.g., Nanda et al. 2020; Xu and Xu 2020; Ma and Xu 2022) only considered the matching size when optimizing the fairness objective for the offline (drivers) or online (riders) side. This is in contrast to our setting where we consider the matching quality. To see the effect of ignoring the matching quality and only considering the size, we run the same experiments with $w_e^U = w_e^V = 1, \forall e \in E$, i.e. the quality is ignored. The results are shown shown in the graph labelled “Matching” in figure 1,

	Profit	Driver Fairness	Rider Fairness
Greedy-O	0.431	0.549	0.503
TSGF _{KIID} ($\alpha = 1$)	0.595	0.398	0.384
Greedy-D	0.371	0.609	0.563
TSGF _{KIID} ($\beta = 1$)	0.517	0.571	0.44
Greedy-R	0.316	0.504	0.513
TSGF _{KIID} ($\gamma = 1$)	0.252	0.353	0.574

Table 1: Competitive ratios of TSGF_{KIID} with Greedy heuristics on the NYC dataset at $|U| = 49, |V| = 172$. Higher competitive ratio indicates better performance.

it is clear that ignoring the match quality leads to noticeably worse results.

Comparison to Heuristics: We also compare the performance of TSGF_{KIID} against three other heuristics. In particular, we consider Greedy-O which is a greedy algorithm that upon the arrival of an online vertex (rider) v picks the edge $e \in E_v$ with maximum value of $p_e w_e^O$ until it either results in a match or the patience quota is reached. We also consider Greedy-R which is identical to Greedy-O except that it greedily picks the edge with maximum value of $p_e w_e^V$ instead, therefore maximizing the rider’s utility in a greedy fashion. Moreover, we consider Greedy-D which is a greedy algorithm that upon the arrival of an online vertex v , first finds the group on the offline side with the lowest average utility so far, then it greedily picks an offline vertex (driver) $u \in E_v$ from this group (if possible) which has the maximum utility until it either results in a match or the patience limit is reached. We carried out 100 trials to compare the performance of TSGF_{KIID} with the greedy algorithms, where each trial contains 49 randomly sampled drivers and 172 requests and is repeated 100 times. The aggregated results are displayed in table ?? We see that TSGF_{KIID} outperforms the heuristics with the exception of a small under-performance in comparison to Greedy-D. However, using Greedy-D we cannot tune the weights (α, β , and γ) to balance the objectives as we can in the case of TSGF_{KIID}.

[‡]<https://github.com/anonymousUser634534/TSGF>

Acknowledgments

This research was supported in part by NSF CAREER Award IIS-1846237, NSF Award CCF-1918749, NSF Award CCF-1852352, NSF Award SMA-2039862, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DARPA SI3-CMD #S4761, DoD WHS Award #HQ003420F0035, ARPA-E DIFFERENTIATE Award #1257037, ARL Award W911NF2120076, and gifts by research awards from Amazon, and Google. We are grateful to Pan Xu for advice and comments on earlier versions of this work.

References

- Adamczyk, M.; Grandoni, F.; and Mukherjee, J. 2015. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, 1–12. Springer.
- Alaei, S.; Hajiaghayi, M.; and Liaghat, V. 2013. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 11–25. Springer.
- Alonso-Mora, J.; Wallar, A.; and Rus, D. 2017. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3583–3590.
- Bansal, N.; Gupta, A.; Li, J.; Mestre, J.; Nagarajan, V.; and Rudra, A. 2010. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. In *European Symposium on Algorithms*, 218–229. Springer.
- Barann, B.; Beverungen, D.; and Müller, O. 2017. An open-data approach for quantifying the potential of taxi ridesharing. *Decision Support Systems*, 99: 86–95.
- Barocas, S.; Hardt, M.; and Narayanan, A. 2019. *Fairness and Machine Learning*. fairmlbook.org. Accessed: 2022-08-01.
- Basu, K.; DiCiccio, C.; Logan, H.; and Karoui, N. E. 2020. A Framework for Fairness in Two-Sided Marketplaces. *arXiv preprint arXiv:2006.12756*.
- Brubach, B.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2016. Online Stochastic Matching: New Algorithms and Bounds. *arXiv:1606.06395*.
- Cook, G. 2018. *Woman Says Uber Driver Denied Her Ride Because of Her Wheelchair*. NBC4-Washington. Available at <https://www.nbcwashington.com/news/local/woman-says-uber-driver-denied-her-ride-because-of-her-wheelchair/2029780/>, Accessed: 2023-04-23.
- Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2019. Balancing relevance and diversity in online bipartite matching via submodularity. In *AAAI*.
- Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2021. Allocation Problems in Ride-sharing Platforms: Online Matching with Offline Reusable Resources. *ACM Transactions on Economics and Computation (TEAC)*, 9(3): 1–17.
- Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *ITCS*.
- Feldman, J.; Mehta, A.; Mirrokni, V.; and Muthukrishnan, S. 2009. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, 117–126. IEEE.
- Galperin, H.; and Greppi, C. 2017. Geographical discrimination in the gig economy. Available at SSRN 2922874.
- 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*, 1–45.
- Goel, G.; and Mehta, A. 2008. Online budgeted matching in random input models with applications to Adwords. In *SODA*.
- Ho, C.-J.; and Vaughan, J. 2012. Online task assignment in crowdsourcing markets. In *AAAI*.
- Lee, M. K.; Kusbit, D.; Kahng, A.; Kim, J. T.; Yuan, X.; Chan, A.; See, D.; Noothigattu, R.; Lee, S.; Psomas, A.; and Procaccia, A. D. 2019. WeBuildAI: Participatory framework for algorithmic governance. In *CSCW*.
- Lesmana, N. S.; Zhang, X.; and Bei, X. 2019. Balancing efficiency and fairness in on-demand ridesourcing. In *NeurIPS*.
- Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2018. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence (AIJ)*, 261: 71–112.
- Ma, W.; and Xu, P. 2022. Group-level fairness maximization in online bipartite matching. In *AAMAS*.
- Ma, W.; Xu, P.; and Xu, Y. 2020. Group-level fairness maximization in online bipartite matching. *arXiv preprint arXiv:2011.13908*.
- Ma, W.; Xu, P.; and Xu, Y. 2021. Fairness Maximization among Offline Agents in Online-Matching Markets. In *WINE*.
- Mattei, N.; Saffidine, A.; and Walsh, T. 2017. Mechanisms for Online Organ Matching. In *IJCAI*.
- Mehta, A. 2013. Online Matching and Ad Allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4): 265–368.
- Mitzenmacher, M.; and Upfal, E. 2017. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press.
- Nanda, V.; Xu, P.; Sankararaman, K. A.; Dickerson, J. P.; and Srinivasan, A. 2020. Balancing the Tradeoff between Profit and Fairness in Rideshare Platforms During High-Demand Hours. In *AAAI*.
- Patro, G. K.; Biswas, A.; Ganguly, N.; Gummadi, K. P.; and Chakraborty, A. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *Proceedings of The Web Conference 2020*, 1194–1204.
- Rawls, J. 1958. Justice as fairness. *The philosophical review*, 67(2): 164–194.
- Rosenblat, A.; Levy, K. E.; Barocas, S.; and Hwang, T. 2016. Discriminating tastes: Customer ratings as vehicles for bias. Available at SSRN 2858946.

- 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*.
- Sekulić, S.; Long, J.; and Demšar, U. 2021. A spatially aware method for mapping movement-based and place-based regions from spatial flow networks. *Transactions in GIS*, 25(4): 2104–2124.
- Sühr, T.; Biega, A. J.; Zehlike, M.; Gummadi, K. P.; and Chakraborty, A. 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *KDD*.
- Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*.
- White, G. B. 2016. *Uber and Lyft Are Failing Black Riders*. The Atlantic. Available at <https://www.theatlantic.com/business/archive/2016/10/uber-lyft-and-the-false-promise-of-fair-rides/506000/>, Accessed: 2023-04-22.
- Wirtschafter, E. 2019. *Driver discrimination still a problem as Uber and Lyft prepare to go public*. KALW. Available at <https://www.kalw.org/post/driver-discrimination-still-problem-uber-and-lyft-prepare-go-public>, Accessed: 2023-04-23.
- Xu, Y.; and Xu, P. 2020. Trade the System Efficiency for the Income Equality of Drivers in Rideshare. In *IJCAI*.
- Zhou, Q.; Marecek, J.; and Shorten, R. N. 2021. Subgroup Fairness in Two-Sided Markets. *arXiv preprint arXiv:2106.02702*.