Balancing the Tradeoff between Profit and Fairness in Rideshare Platforms during High-Demand Hours

Vedant Nanda,[1,2] **Pan Xu,**[3] **Karthik Abinav Sankararaman,**[1,4]
John P. Dickerson,[1] **Aravind Srinivasan**[1]

[1]University of Maryland, College Park, [2]Max Planck Institute for Software Systems
[3]New Jersey Institute of Technology, New Jersey, [4]Facebook
{vedant, john, srin}@cs.umd.edu, pxu@njit.edu, karthikabinavs@gmail.com

Abstract

Rideshare platforms, when assigning requests to drivers, tend to maximize profit for the system and/or minimize waiting time for riders. Such platforms can exacerbate biases that drivers may have over certain types of requests. We consider the case of peak hours when the demand for rides is more than the supply of drivers. Drivers are well aware of their advantage during the peak hours and can choose to be selective about which rides to accept. Moreover, if in such a scenario, the assignment of requests to drivers (by the platform) is made only to maximize profit and/or minimize wait time for riders, requests of a certain type (*e.g.,* from a non-popular pickup location, or to a non-popular drop-off location) might never be assigned to a driver. Such a system can be highly unfair to riders. However, increasing fairness might come at a cost of the overall profit made by the rideshare platform. To balance these conflicting goals, we present a flexible, non-adaptive algorithm, NAdap, that allows the platform designer to control the profit and fairness of the system via parameters α and β respectively. We model the matching problem as an online bipartite matching where the set of drivers is offline and requests arrive online. Upon the arrival of a request, we use NAdap to assign it to a driver (the driver might then choose to accept or reject it) or reject the request. We formalize the measures of profit and fairness in our setting and show that by using NAdap, the competitive ratios for profit and fairness measures would be no worse than α/e and β/e respectively. Extensive experimental results on both real-world and synthetic datasets confirm the validity of our theoretical lower bounds. Additionally, they show that NAdap under some choice of (α, β) can beat two natural heuristics, Greedy and Uniform, on *both* fairness and profit. Code is available at: https://github.com/nvedant07/rideshare-fairness-peak/.

1 Introduction

Rideshare platforms have received significant attention in both the computer science and operations research communities. The following are the three main categories of research. The first studies matching policy design in the rideshare setting (*i.e.,* matching riders and drivers), *e.g.,* (Zhao et al. 2019; Ashlagi et al. 2019; Lowalekar, Varakantham, and Jaillet 2018; Tong et al. 2016b; 2016a; 2017;

Bei and Zhang 2018; Dickerson et al. 2018b; 2018a). The second considers the spatial-temporal pricing aspects of rideshare, *e.g.,* (Ma, Fang, and Parkes 2019; Bimpikis, Candogan, and Saban 2017; Kanoria and Qian 2019; Banerjee, Freund, and Lykouris 2017; Banerjee, Johari, and Riquelme 2016). The third focuses on applying reinforcement-learning approaches to planning and matching problems in rideshare see, *e.g.,* (Xu et al. 2018; Lin et al. 2018).

In all the aforementioned prior work, the objective is either to maximize the total profit of the system or minimize the waiting time of riders (or a combination thereof). However, both of these objectives are *global* in that they do not ensure sub-group level fairness (*e.g.,* riders of a particular protected class being systematically underserved). Consider a scenario during peak hours when there is increased demand for rides, thus, giving the drivers a bargaining advantage to drop rides. Typical criteria used by drivers to reject riders include riders' starting/ending location, trip length, gender, race and age. Recently, it has been reported that drivers also reject riders based on attributes such as gender, race and disability either intentionally or unintentionally, see, *e.g.,* (Laurence 2019; Cook 2019; Griffin 2019). Houser (2018) reported that "black passengers using ride-hailing apps have to wait an average of 1 minute and 43 seconds longer than white counterparts and are 4% more likely to have drivers cancel on them."

Thus, current rideshare platforms can enable and amplify prejudices in society. To counter this, current rideshare apps implement the following measures:[1] (1) Riders' photo and destination are hidden from the driver until they accept/reject the request; (2) Penalty is imposed if drivers cancel a certain number of trips after initially accepting them (*e.g.,* drivers' account getting temporarily deactivated). However, drivers devise new strategies to avoid the restrictions imposed by the above measures. It was reported (Paul 2018) that some Uber drivers bypass the limitation imposed by (1) by starting the trip moments before picking up a passenger to see where the passenger is going. In some other cases, drivers get around the limitation imposed by (2) by intentionally delaying the pickup, thus forcing the rider to

---

[1]See, for example, §4A "Sharing Between Users" in Lyft's privacy policy found at https://www.lyft.com/privacy.

cancel the trip instead.

Ensuring group level fairness and optimizing global profit tends to be somewhat conflicting goals in general; particularly so in rideshare platforms during peak hours. The rationale is that the driver to request ratio is small and thus, drivers can afford to be more choosy. This may lead to unfair practices such as rejecting trips to unpopular destinations and rejecting riders with disabilities. To promote group level fairness among the riders, systems should aggressively match requests which typically incur high cancellation rates. This action, however, would increase the number of cancellations by drivers. The current penalty in popular rideshare platforms is that drivers are deactivated if they cancel too many requests. Thus, prioritizing high risk (in terms of cancellation) trips could lead to more drivers being deactivated and/or leaving the system. In summary, on one hand, prioritizing group level fairness leads to limited drivers and thus, limited total trips. On the other hand, letting the market find its equilibrium leads to amplifying societal biases. Thus, the central question is the following: *Can we design policies that can smoothly tradeoff between the two conflicting objectives*?

In this paper, we answer the above question in the affirmative by providing *provably* efficient policies. As is common (see, e.g., Dickerson et al. (2018a) for additional motivation) we model the dynamics of the rideshare platform as an online-matching model as follows. We have a bipartite graph $G = (U, V, E)$, where U and V represent the set of available drivers (static or offline) and the request types (dynamic or online arrival) respectively.[2] We use the notation $m := |U|$ and $n := |V|$ throughout this paper. Each driver *type* represents a specific group (*e.g.,* gender, age and race) in a given location, while each request type represents a specific group with a given starting and ending location. There is an edge $f = (u, v)$ if u is capable of serving the request (of type) v (*i.e.,* the distance between them is below a given threshold). The online phase consists of T time-steps with the sets U and V known to the algorithm. In each time-step, a request $v \in V$ arrives and is presented to the algorithm. Upon its arrival an *immediate and irrevocable* decision is required: either reject v, or match v with an available driver in U. WLOG we assume that each u has a unit capacity (which can be matched only once).[3]

We have the following key assumptions that we use to show provably guarantees. In the experimental section, we work with real data and show that the algorithms are robust even when some of these assumptions do not necessarily hold.

Arrival of Requests. We consider a finite set of T requests $v \in V$ that are drawn from a *known* identical distribution independently; this is commonly (Dickerson et al.

2018a) called the *known identical independent distributions* (KIID). The motivation for this assumption stems from the fact that we can often learn the arrival distribution from historical logs (Yao et al. 2018; Li et al. 2018; Wang, Fu, and Ye 2018). KIID is widely used in many practical applications of online matching markets including rideshare and crowdsourcing (Zhao et al. 2019; Dickerson et al. 2018b; Singer and Mittal 2013; Singla and Krause 2013). Further, we call the expected number of times any request v is sampled from this distribution in the T rounds as the *arrival rate*, denoted by r_v. Thus, it is easy to see that $\sum_{v \in V} r_v = T$. We further assume that the total number of arrivals of online requests is far larger than that of drivers in the system, *i.e.,* $T \gg |U|$.

Edge existence probabilities. Each edge $f = (u, v)$ is associated with an existence probability $p_f \in (0, 1]$: once we assign v to u, we observe an immediate random outcome of the existence, which is present (*i.e.,* u accepts v) with probability p_f and not (u cancels v) otherwise. The probability p_f captures the statistical chance that a driver of type u would accept to serve the request of type v. We assume that (1) the randomness driving the edge existence is independent across all the edges; (2) the values p_f are provided as part of the input. The first assumption is motivated by individual choice and the second from the fact that historical logs can be used to compute such statistics with high precision.

Cancellation budget. Each driver u is associated with a given budget of cancellation, $\Delta_u \in \mathbb{Z}^+$. In other words, driver $u \in U$ will be removed from the graph G if they cancel more than Δ_u requests in the T rounds (in which case we assume that u is temporarily deactivated by the system as a penalty). Once a request v gets rejected, we assume that the system will not try any reassignment of v to other available drivers. This is without loss of generality since any reassignment can be modeled as resampling in the succeeding time-steps.

We assume that the system gains a profit w_f from $f = (u, v)$ if driver u completes (*i.e.,* is assigned and the driver accepts) the trip v (in this case, we call it a *successful* assignment or a *match*). For a given policy ALG, let $\mathcal{M}$ (possibly random) be the set of successful assignments; we interchangeably use the term *matching* to denote this set $\mathcal{M}$. We define two objectives, namely *profit* and *fairness*, as follows.

Profit: The expected total profit over all matches, which is defined as $\mathbb{E}[\sum_{e \in \mathcal{M}} w_e]$.

Fairness: Let $\mathcal{M}_v \subseteq \mathcal{M}$ be the subset of edges incident to v. Note that $|\mathcal{M}_v|$ can be larger than 1 due to multiple arrivals of type v in the T time-steps. We define the fairness achieved by ALG over all request types as $\min_{v \in V} \frac{\mathbb{E}[|\mathcal{M}_v|]}{r_v}$, which refers to the minimum ratio of the expected number of matches of type v to that of arrivals.[4] Thus, maximizing fairness corresponds to maximizing this minimum ratio.

---

[2]We consider a small time window during the peak hours, thus assuming that the set of drivers are static.

[3]We can create multiple copies of u to address the case when each u can be matched multiple times. Note that each copy corresponds to the same driver and shares same underlying variables.

---

[4]Since we are in the context of peak hours, we consider the group level fairness of riders.

We aim to design an online matching policy that balances the tradeoff between the two objectives of maximizing profit and fairness.

1.1 Preliminaries and Main Contributions

Competitive ratio. Competitive ratio is a commonly-used metric to evaluate the performance of online algorithms. Consider an online maximization problem for example. Let $\text{ALG}(\mathcal{I}) = \mathbb{E}_{I \sim \mathcal{I}}[\text{ALG}(I)]$ denote the expected performance of ALG on a given distribution $\mathcal{I}$, where the expectation is taken over the random arrival sequence I. Let $\text{OPT}(\mathcal{I}) = \mathbb{E}[\text{OPT}(I)]$ denote the expected *offline optimal*, where $\text{OPT}(I)$ refers to the optimal value after we observe the full arrival sequence I. Then, the competitive ratio is defined as $\min_{\mathcal{I}} \frac{\text{ALG}(\mathcal{I})}{\text{OPT}(\mathcal{I})}$. It is a common technique to use a Linear Program (LP) to upper bound $\text{OPT}(\mathcal{I})$ (called the benchmark LP) and hence get a valid lower bound on the target competitive ratio. In our paper, we conduct competitive-ratio analysis on both objectives.

Main contributions. This paper provides three-fold contributions.

First, we formalize the metric of fairness in rideshare. More specifically, we consider the online-matching based model with multiple objectives.

Second, we present a provably efficient algorithm and provide formal mathematical guarantees. To do so, we first propose a bi-objective linear program (LP-(1) and LP-(2)), whose optimal value is at least as large as that of any online algorithm that maximizes either of these objectives (or a combination thereof). Our main algorithm NAdap uses this bi-objective LP to guide the online decision-making process. In particular, we prove the following main theorems.

Theorem 1. NAdap(α, β) *achieves a competitive ratio at least* $\left(\alpha/e, \beta/e\right)$ *simultaneously on the profit and fairness for any given* $\alpha, \beta > 0$ *with* $\alpha + \beta \leq 1$.

Theorem 2. *No non-adaptive algorithm can achieve a* (α, β)*-competitive ratio simultaneously on the profit and fairness with* $\alpha + \beta > 1 - 1/e$ *using* LP-(1) *and* LP-(2) *as the benchmark.*

Third, we provide an extensive evaluation of the algorithm and modeling assumption on a real-world dataset collected from a large on-demand taxi dispatching platform. The experiments have many novel insights; among others, we show that even when some of the assumptions that were used to prove mathematical guarantees do not hold, NAdap performs well in practice.

2 Additional Related Work

There is a large body of work which studies fairness issues in resource allocation problems (divisible or indivisible goods), see, *e.g.,* (Ghodsi et al. 2011; Bateni et al. 2016; Parkes, Procaccia, and Shah 2015; Kash, Procaccia, and

Shah 2014; Fain, Munagala, and Shah 2018). Most of these works require the allocation policy to satisfy certain properties in fair mechanism design such as strategy-proofness and envy-freeness. These properties do not apply here, however. Recent work by Sühr et al. (2019) proposes a matching policy to balance fairness and profit over time; however, they do not provide any optimality guaranty for their proposed policy. Lesmana, Zhang, and Bei (2019) has a setting similar to ours, however they consider the case when there are more drivers than riders. Our focus is on the case when the number of drivers in the system are less than the demand for rides (*i.e.,* peak hours).

Our model belongs to a more general optimization paradigm, called *Multi-Objective Optimization*. Ravi et al. (1993) presented approximation algorithms for a variety of network-design problems. Grandoni, Ravi, and Singh (2009) designed several iterative-rounding based approximation algorithms for multi-objective optimization problems. More recently, Aggarwal et al. (2014) studies the Bi-objective Online Bipartite Matching where there is essentially one single objective: the minimum matching ratio over two disjoint sets of edges. Esfandiari, Korula, and Mirrokni (2016) considers the Bi-objective Online Submodular Optimization problem, where the two objectives are two monotone submodular functions. Our objectives are two linear functions, but our model assumes a more complicated setting (*i.e.,* edge existence probabilities and cancellation quotas on the offline-side vertices). The models studied by Bansal et al. (2012) and Brubach et al. (2017) have the closest setting to us: each edge has an independent existence probability and each vertex from the offline and/or online side has a patience constraint (similar to cancellation quota for each driver in our setting) on it. However, both those works investigated only one single objective: maximization of the total profit.

3 Valid Benchmarks for Profit and Fairness

We first present our benchmark LPs and then an LP-based parameterized algorithm. For each edge $f = (u, v)$, let x_f be the expected number of probes on edge f (*i.e.,* assignments of v to u but not necessarily matches) in the offline optimal. For each u (v), let E_u (E_v) be the set of neighboring edges incident to u (v). Consider the following bi-objective LP.

$$\max \quad \sum_f w_f x_f p_f \tag{1}$$

$$\max \min_{v \in V} \quad \frac{\sum_{f \in E_v} x_f p_f}{r_v} \tag{2}$$

$$\text{s.t.} \quad \sum_{f \in E_u} x_f p_f \leq 1 \quad \forall u \in U \tag{3}$$

$$\sum_{f \in E_u} x_f \leq \Delta_u \quad \forall u \in U \tag{4}$$

$$\sum_{f \in E_v} x_f \leq r_v \quad \forall v \in V \tag{5}$$

$$0 \leq x_f \quad \forall f \in E \tag{6}$$

Let LP-(1) and LP-(2) denote the two LPs with the respective Objective (1) and (2), each with constraints (3), (4), (5), (6). Note that we can rewrite Objective (2) as a linear one like $\max \eta$ with additional linear constraints as $\eta \leq \frac{\sum_{f \in E_v} x_f p_f}{r_v}$ for all $v \in V$. For presentation convenience, we keep the current compact version. The validity of LP-(1) and LP-(2) as benchmarks for our two objectives can be seen in the following lemma.

Lemma 1. LP-(1) *and* LP-(2) *are valid benchmarks for the two respective objectives, profit and fairness. In other words, the optimal values to* LP-(1) *and* LP-(2) *are valid upper bounds for the expected profit and fairness achieved by the offline optimal respectively.*

Proof. We can verify that objective functions (1) and (2) each captures the exact expected profit and fairness achieved by the offline optimal, according to our definition in Section 1. To prove the validity of the benchmark for each objective, it suffices to show the feasibility of all constraints for any given offline optimal.

Constraint (3) is valid since each driver u has a unit capacity; Constraint (4) is valid since each u can be probed at most Δ_u times according to our assumption; Constraint (5) is valid since the expected number of probes related to each v should be no more than that of online arrivals (recall that we can try at most one assignment upon the arrival of v). Thus we justify the feasibility of all constraints for any given offline optimal. $\qquad\square$

4 An LP-based Parameterized Algorithm

Now we present an LP-based parameterized algorithm. Let $\{x_f^*\}$ and $\{y_f^*\}$ be an optimal solution to LP-(1) and LP-(2) respectively. Consider a given pair of parameters (α, β) with $0 \leq \alpha, \beta \leq 1, \alpha + \beta \leq 1$. NAdap$(\alpha, \beta)$ will sample an assignment guided by the two LP solutions $\{x_f^*\}$ and $\{y_f^*\}$, with probabilities p and q respectively. Note that constraint (4) for the LP models the expectation of number of probes. While this number might exceed Δ_u, it's important to note that this solution is for the offline case and does not correspond to the actual assignments. During the execution of our algorithm, we explicitly check if a driver is available before making the assignment, thus ensuring that no driver is assigned more than Δ_u rides. The details are as follows.

The key part to prove Theorem 1 is the computation of the probability that each driver is available at each time. Focus on a given driver u and a time $t \in [T]$. Let $\mathsf{SF}_{u,t}$ be the event that u is *available* at (the beginning of) t. Note that the occurrence of $\mathsf{SF}_{u,t}$ can be guaranteed by these two events: (1) u never *received and simultaneously accepted* any assignment prior to t; (2) u received no more than $\Delta_u - 1$ assignments prior to t. For each $f = (u, v) \in E_u$ and $\ell < t$, let $X_{f,\ell}$ indicate if f comes (or v comes) at time ℓ, $Y_{f,t}$ indicate if f gets sampled in NAdap(p, q) at t and $Z_{f,t}$ indicate if u accepts f after the assignment. Set $A_{u,t} \doteq \sum_{\ell < t} \sum_{f \in E_u} X_{f,\ell} Y_{f,\ell} Z_{f,\ell}$ and

$B_{u,t} \doteq \sum_{\ell < t} \sum_{f \in E_u} X_{f,\ell} Y_{f,\ell}$. From our analysis, we see that $\Pr[\mathsf{SF}_{u,t}] \geq \Pr[(A_{u,t} = 0) \wedge (B_{u,t} \leq \Delta_u - 1)]$.

Lemma 2. *For any given u and $t \in [T]$, we have*

$$\Pr[A_{u,t} = 0] \geq \left(1 - \frac{1}{T}\right)^{t-1}, \quad \Pr[B_{u,t} \leq \Delta_u - 1] \geq 1 - \frac{t-1}{T}$$

Proof.

$$
\begin{aligned}
\Pr[A_{u,t} = 0] &= \prod_{\ell < t} \Pr\left[\sum_{f \in E_u} X_{f,\ell} Y_{f,\ell} Z_{f,\ell} = 0\right] \\
&= \prod_{\ell < t} \left(1 - \Pr\left[\sum_{f \in E_u} X_{f,\ell} Y_{f,\ell} Z_{f,\ell} \geq 1\right]\right) \\
&= \prod_{\ell < t} \left(1 - \sum_{f \in E_u} \frac{r_v}{T}\left(\alpha \frac{x_f^*}{r_v} + \beta \frac{y_f^*}{r_v}\right) p_f\right) \\
&= \prod_{\ell < t} \left(1 - \frac{1}{T} \sum_{f \in E_u} \left(\alpha x_f^* p_f + \beta y_f^* p_f\right)\right) \\
&\geq \left(1 - \frac{1}{T}\right)^{t-1}
\end{aligned}
$$

The last inequality follows from $\sum_{f \in E_u} x_f^* p_f \leq 1$ and $\sum_{f \in E_u} y_f^* p_f \leq 1$ due to Constraint (3), and $\alpha + \beta \leq 1$.

As for the second inequality, we have

$$\Pr[B_{u,t} \leq \Delta_u - 1] = 1 - \Pr[B_{u,t} \geq \Delta_u] \geq 1 - \frac{1}{\Delta_u}\mathbb{E}[B_{u,t}] \quad (7)$$

Note that

$$\mathbb{E}[B_{u,t}] = \sum_{\ell < t} \sum_{f \in E_u} \mathbb{E}[X_{f,\ell} Y_{f,\ell}] = \sum_{\ell < t} \sum_{f \in E_u} \frac{r_v}{T}\left(\frac{\alpha x_f^*}{r_v} + \frac{\beta y_f^*}{r_v}\right) \quad (8)$$

$$\leq \frac{1}{T} \sum_{\ell < t} (\alpha + \beta)\Delta_u \leq \frac{t-1}{T}\Delta_u \quad (9)$$

The last inequalities above follows from $\sum_{f \in E_u} x_f^* \leq \Delta_u$ and $\sum_{f \in E_u} y_f \leq \Delta_u$ due to Constraint (4), and $\alpha + \beta \leq 1$. Substituting Inequality (9) to (7), we have that $\Pr[B_{u,t} \leq \Delta_u - 1] \geq 1 - (t-1)/T$. $\qquad\square$

Lemma 3.

$$\Pr[B_{u,t} \leq \Delta_u - 1 | A_{u,t} = 0] \geq 1 - \frac{t-1}{T}$$

Lemma 3 says that the two events $(A_{u,t} = 0)$ and $(B_{u,t} \leq \Delta_u - 1)$ are positively correlated. Proof of Lemma 3 can be found in the full version of the paper (on Arxiv).

Now we have all ingredients to prove the main Theorem 1.

Proof. From Lemmas 2 and 3, we have for each u and $t \in [T]$,

$$\Pr[\mathsf{SF}_{u,t}] \geq \Pr[(A_{u,t} = 0) \wedge (B_{u,t} \leq \Delta_u - 1)]$$
$$\geq \Pr[A_{u,t} = 0] \Pr[B_{u,t} \leq \Delta_u - 1] \geq \left(1 - \frac{1}{T}\right)^{t-1}\left(1 - \frac{t-1}{T}\right)$$

For each $f \in E$, let κ_f be the expected number of *successful* assignments of f in $\mathsf{NAdap}(\alpha, \beta)$. Here an assignment $f = (v, u)$ is successful iff u is available when we assign v to u (but no necessarily means u accepts v).

$$\kappa_f = \sum_{t=1}^{T} \frac{r_v}{T} \Pr[\mathsf{SF}_{u,t}]\left(\frac{\alpha x_f^*}{r_v} + \frac{\beta y_f^*}{r_v}\right)$$
$$\geq \sum_{t=1}^{T} \frac{1}{T}\left(1 - \frac{1}{T}\right)^{t-1}\left(1 - \frac{t-1}{T}\right)\left(\alpha x_f^* + \beta y_f^*\right) \sim \frac{\alpha x_f^* + \beta y_f^*}{e}$$

The last term is obtained after taking $T \to \infty$.

Let $\mathsf{Profit}(\alpha, \beta)$ be the expected total profit obtained by $\mathsf{NAdap}(\alpha, \beta)$. By linearity of expectation, we have $\mathsf{Profit}(\alpha, \beta) \geq \sum_{f \in E} \frac{1}{e}(\alpha x_f^* + \beta y_f^*) p_e w_e$. From Lemma 1, we know that the expected profit in offline optimal is upper bounded by $\sum_{f \in E} x_f^* p_e w_e$. Thus we claim that $\mathsf{NAdap}(\alpha, \beta)$ achieves a ratio at least α/e on the profit. Similarly, we can argue that $\mathsf{NAdap}(\alpha, \beta)$ achieves a ratio at least β/e on the fairness. $\square$

5 Hardness Results

The model in our paper has two objectives which complicate the hardness analysis. To simplify it, we focus only on those non-adaptive algorithms. We characterize a non-adaptive algorithm as $\vec{z} = \{z_f | f \in E_f\}$ where each $z_f \in [0,1]^{N_v}$ (N_v is the size of E_v, the set of edges incident to v) such that $\sum_{f \in E_v} z_f \leq 1$. A non-adaptive algorithm parameterized with $\vec{z}$, denoted by $\mathsf{NADAP}(\vec{z})$ will sample an assignment $f = (u, v) \in E_v$ with probability z_f upon the arrival of v, and assign v to u if u is available. Note that our LP-based parameterized algorithm $\mathsf{NAdap}(\alpha, \beta)$ can be viewed as a specific non-adaptive with $z_f = \alpha x_f^*/r_v + \beta y_f^*/r_v$ for each f.

Example 1. *Consider a star graph where $U = \{u\}$ with $\Delta_u = 1$, $V = \{v_0, v_1, \ldots, v_K\}$ and $T = K + 1$. We use j to denote v_j and edge $f_j = (u, v_j)$ when the context is clear. Let $w_j = 1$ and $r_j = 1$ for all $j = 0, 1, \ldots, K$. In other words, our star graph is unweighted and the arrival distributions are uniform. Set $p_j = 1$ for $j = 0$ and $p_j = \epsilon$ for each $j \in [K]$. Let OPT-P and OPT-F denote the optimal LP values of LP-(1) and LP-(2) on this example respectively. We can verify that: (1) OPT-P $= 1$, where there is a unique optimal solution $x_0 = 1$ and $x_j = 0$ for all $j \in [K]$; (2) OPT-F $= \frac{\epsilon}{K+\epsilon}$, where there is a unique optimal solution $x_0 = \frac{\epsilon}{\epsilon + K}$ and $x_j = \frac{1}{\epsilon + K}$ for all $j \in [K]$.*

Consider a given non-adaptive algorithm $\mathsf{NADAP}(\vec{z})$, with $\vec{z} = \{z_0, z_1, \ldots, z_K\}$ on the above example. Let $P(\vec{z})$ and $F(\vec{z})$ be the expected total profit and fairness achieved by $\mathsf{NADAP}(\vec{z})$. Set $z = \sum_{j=0}^{K} z_j$. Note that $0 \leq z \leq 1$.

Lemma 4. *The sum of competitive ratios achieved by $\mathsf{NADAP}(\vec{z})$ for the profit and fairness on Example 1 is no larger than $1 - 1/e + 2\epsilon$.*

Proof.

$$P(\vec{z}) = \frac{z_0}{T} \sum_{t=1}^{T}\left(1 - \frac{\sum_{j=0}^{K} z_j}{T}\right)^{t-1} + \sum_{j=1}^{K} \frac{z_j \epsilon}{T} \sum_{t=1}^{T}\left(1 - \frac{\sum_{j=0}^{K} z_j}{T}\right)^{t-1}$$
$$= \frac{z_0}{z}\left(1 - \exp(-z)\right) + \frac{(\sum_{j=1}^{K} z_j)\epsilon}{z}\left(1 - \exp(-z)\right) \quad (T \to \infty)$$
$$\leq \frac{z_0}{z}\left(1 - \exp(-z)\right) + \epsilon$$

From the above computation, we see that the expected sum of acceptances of assignments over all $j \in [K]$ during the online phase is $\frac{(\sum_{j=1}^{K} z_j)\epsilon}{z}\left(1 - \exp(-z)\right)$. Thus the minimum ratio should be no larger than the average, from which we have

$$F(\vec{z}) \leq \frac{(\sum_{j=1}^{K} z_j)\epsilon}{zK}\left(1 - \exp(-z)\right)$$

Note that OPT-P $= 1$ and OPT-F $= \frac{\epsilon}{K+\epsilon}$. Thus the sum of competitive ratios on the profit and fairness should be

$$\frac{P(\vec{z})}{\text{OPT-P}} + \frac{F(\vec{z})}{\text{OPT-F}}$$
$$\leq \frac{z_0}{z}\left(1 - \exp(-z)\right) + \epsilon + \frac{(\sum_{j=1}^{K} z_j)}{z}\frac{K+\epsilon}{K}\left(1 - \exp(-z)\right)$$
$$\leq 1 - \exp(-z) + 2\epsilon \leq 1 - 1/e + 2\epsilon$$

$\square$

Notice that Lemma 4 immediately implies our main Theorem 2.

6 Experiments

Real-world dataset processing. We now provide experimental validation of our algorithms. We use a real-world dataset: the New York City yellow cabs dataset[5],[6] which

---

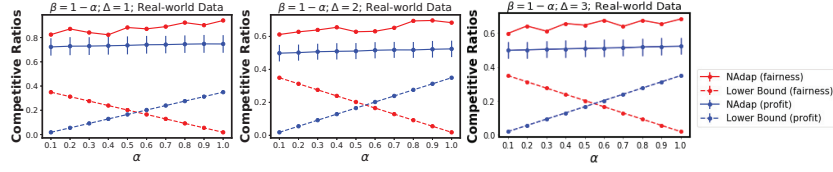[5]http://www.andresmh.com/nyctaxitrips/
[6]https://tinyurl.com/y56gonvs

Figure 1: Real dataset: competitive ratios for profit and fairness with different values of α and β with $\alpha + \beta = 1$. $|U| = 48, |V| = 24, T = 359$. $\Delta = 1$ (Left), $\Delta = 2$ (Middle), and $\Delta = 3$ (Right).
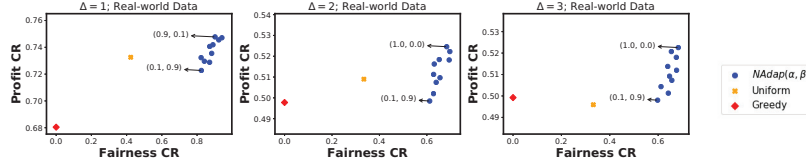

Figure 2: Real dataset: comparison of performances of NAdap against Greedy and Uniform. $|U| = 48, |V| = 24, T = 359$. $\Delta = 1$ (Left), $\Delta = 2$ (Middle), and $\Delta = 3$ (Right).

contains taxi trip records from Manhattan, Brooklyn, and Queens during the year 2013. The dataset is split into 12 months. For each month we have a the record of a completed trip that contains an anonymized hash of a driver's license, the start and end location of the trip (latitude and longitude), time and date when the trip was initiated and ended, time taken to complete the trip, distance of the trip (in miles) and additional attributes such as number of passengers and registration number. Demographics of the drivers and riders are not known; however, to demonstrate our methodology, we randomly assign a "disadvantaged" or "advantaged" race status to all requesters such that ratio of disadvantaged riders to advantaged riders is 1:2, which roughly matches the racial demographics of the city.[7] For drivers, we randomly assign race such that the ratio of disadvantaged to advantaged drivers is 3:1.[8,9] For the purpose of demonstration, we consider race to be binary; however, in reality, there would be many other races, which our model can easily incorporate, given that the distribution of p_f is known or can be learned.

We consider the rush hour of 7–8PM; in our dataset, this period typically has the largest number of trips per hour of the day. On Jan. 31st 2013, a total of 35,109 trips were completed by a total of 10,814 drivers. One driver might make multiple trips during this period. However, here we assume that once a driver starts the trip, they are out of the system. So, we assume that all 10,814 drivers are present at the start and need to serve a total of 35,109 trips (*i.e.*, $T = 35,109$). This is consistent with peak hour assumptions ($T \gg |U|$). Similar to Dickerson et al. (2018a), we focus on longitudes from $-73°$ to $-75°$ and latitudes from $40.4°$ to $40.95°$, both with a step size of 0.05. We then assign an index to each grid

and every location with latitudes and/or longitude in a particular grid is represented by the index of that grid.

We construct our graph $G = (U, V, E)$ as follows. Each $u \in U$ represents a driver type which has attributes of the starting location and race. Each $v \in V$ represents a request type which has attributes of the starting location, ending location, and race. We downsample from all driver and request types such that $|U| = 48$ and $|V| = 24$. For each u, we set a uniform cancellation quota Δ; across three experiments, we vary $\Delta \in \{1, 2, 3\}$. For each v, we generate a random value $r_v \sim \mathcal{N}(15, 1)$ (normal distribution) and assign the value as its arrival rate. Set $T = \sum_{v \in V} r_v$. We assume that the driver's starting coordinates belong to the same bin as that of the start coordinates of the trip. An edge exists between a driver type u and a request type v, iff the starting coordinates of v is in the same bin as that of u. We set the existing probabilities mainly based on the combination of driver and rider's races. When both driver and rider belong to the advantaged class, we assign $p_f = 0.6$; if driver and rider are both members of the disadvantaged class then we assign $p_f = 0.3$ and in all other cases $p_f = 0.1$. These probabilities are then scaled up by a factor κ, *i.e.*, $p_f = \kappa + (1 - \kappa) \cdot p_f$. In our experiments we set $\kappa = 0.5$. The profit associated with each edge (w_f) is defined as the normalized distance of the request type v (so $0 \leq w_f \leq 1$).

Synthetic dataset. To generate a synthetic dataset, we fix $|U| = 100, |V| = 50, T = 700$, and sample arrival rates randomly from a multinomial distribution to ensure they sum up to $T(= 700)$. Between each pair $u \in U$ and $v \in V$, an edge exists with probability 0.1 and does not exist with probability 0.9. The probabilities p_f are randomly sampled such that $p_f \in [0.5, 1]$ and the profit values (w_f) associated with each edge are randomly sampled such that $w_f \in [0, 1]$. We assign each u the same cancellation quota Δ, and vary $\Delta \in \{1, 2, 3\}$ across three experiments.

---

[7]https://tinyurl.com/y7q8dppr

[8]https://tinyurl.com/y27anseh

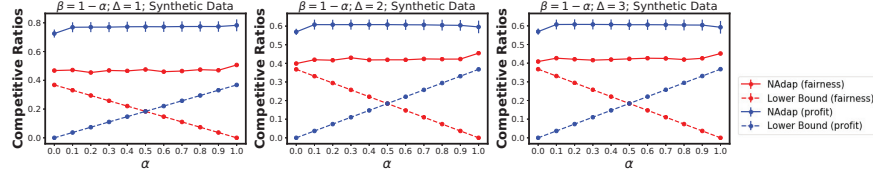[9]For simplicity, we ignore other potential discrimination factors such as gender and disability.

Figure 3: Synthetic dataset: competitive ratios for profit and fairness with different values of α and β with $\alpha + \beta = 1$. $|U| = 100, |V| = 50, T = 700, p_f \in [0.5, 1]$. $\Delta = 1$ (Left), $\Delta = 2$ (Middle), and $\Delta = 3$ (Right).
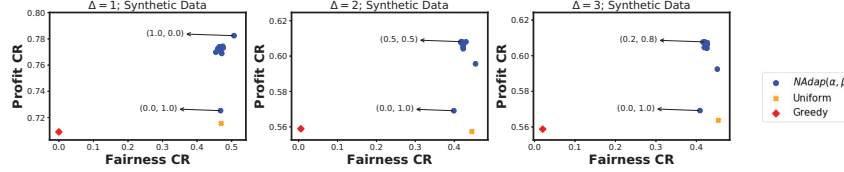

Figure 4: Synthetic dataset: comparison of performances of NAdap against Greedy and Uniform. $|U| = 100, |V| = 50, T = 700, p_f \in [0.5, 1]$. $\Delta = 1$ (Left), $\Delta = 2$ (Middle), and $\Delta = 3$ (Right).

Algorithms. We run NAdap for 5000 iterations and take the average values over these runs to be the expectations. We use profit computed via LP-(1) as the benchmark and use it to calculate the competitive ratio of profit for NAdap. Similarly, we use LP-(2) as the benchmark for fairness and use it to calculate the competitive ratio of fairness for NAdap. For Greedy and Uniform, say a request of type v arrives at time t and let $E_{v,t}$ be the set of available assignments at t and let E_v be all possible assignments for v. In Uniform, we sample an assignment uniformly from E_v and check if it exists in $E_{v,t}$. If it exists, then we make the assignment; otherwise, we reject the request. In Greedy, we select the assignment with the highest p_f value among $E_{v,t}$.

Results on the real and synthetic datasets. Figures 1 and 2 show results on the real dataset. We see that the solid lines (performance of NAdap) always stay above the dotted ones (theoretical lower bounds). Note that a higher β means that NAdap is guided more heavily by $\vec{y}^*$ and a higher α means NAdap is guided more heavily by $\vec{x}^*$. For the cases when this trend is not apparent (*e.g.,* for $\Delta = 3$ in Figure 1), we posit that the optimal solutions ($\vec{x}^*$ and $\vec{y}^*$) are highly correlated. Figure 2 shows that the effectiveness and flexibility of NAdap on both objectives compared to Greedy and Uniform. Specifically, Greedy is very weak and dominated by almost all variants of NAdap on both objectives. This is expected since Greedy as a heuristic is short-sighted and does not make assignments based on the what kind of requests the system expects to get in the future. Uniform seems to achieve relatively high fairness, which makes sense due to the nature of Uniform. However, NAdap provides a wide range of options and some of them can beat Uniform on both objectives as well. Figures 3 and 4 show results on the synthetic dataset. The trends are more consistent since all values here are randomly picked. We see that even in this case NAdap is able to do better than Greedy and Uniform on both the objectives.

7 Conclusion

In this work, we present a flexible approach for matching requests to drivers during peak hours that balances fairness and profit. Our proposed approach allows the system designer to specify how fair and how profitable they want the system to be via two separate parameters. We take a nuanced view of the ridesharing market and model the problem as an online bipartite matching problem with stochastic rewards. One highlight is the introduction of existence probability on each edge, which captures the potential acceptance rate for each pair of driver and rider types. We present an LP-based algorithm that dynamically assigns requests to drivers. Extensive experimental results on both of the real and synthetic datasets show that our proposed approach is not only above the theoretical lower bounds but also can beat natural approaches such as Greedy and Uniform on both objectives. Our work presents many interesting directions for future research. For example, we propose a non-adaptive algorithm in which once a request is rejected (either by the system or the driver), it will not be assigned again. A possible direction for future work could be to re-assign a rejected request. Finally, we assume that drivers do not appear again in the system for the peak hour, however, in reality, that might not always be the case. We hope that our work will encourage the community to look at such problems.

8 Acknowledgements

References

Aggarwal, G.; Cai, Y.; Mehta, A.; and Pierrakos, G. 2014. Biobjective online bipartite matching. In *WINE*, 218–231. Springer.

Ashlagi, I.; Burq, M.; Dutta, C.; Jaillet, P.; Sholley, C.; and Saberi, A. 2019. Edge weighted online windowed matching. In *Proceedings of EC*, 729–742.

Banerjee, S.; Freund, D.; and Lykouris, T. 2017. Pricing and optimization in shared vehicle systems: An approximation framework. EC '17, 517–517.

Banerjee, S.; Johari, R.; and Riquelme, C. 2016. Dynamic pricing in ridesharing platforms. *SIGecom Exch.* 15(1):65–70.

Bansal, N.; Gupta, A.; Li, J.; Mestre, J.; Nagarajan, V.; and Rudra, A. 2012. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica* 63(4):733–762.

Bateni, M. H.; Chen, Y.; Ciocan, D.; and Mirrokni, V. 2016. Fair resource allocation in a volatile marketplace. EC '16, 819–819.

Bei, X., and Zhang, S. 2018. Algorithms for trip-vehicle assignment in ride-sharing. AAAI '18, 3–9.

Bimpikis, K.; Candogan, O.; and Saban, D. 2017. Spatial pricing in ride-sharing networks. NetEcon '17, 5:1–5:1.

Brubach, B.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2017. Attenuate locally, win globally: An attenuation-based framework for online stochastic matching with timeouts. In *Proceedings of AAMAS*, 1223–1231.

Cook, G. 2019. Woman says uber driver denied her ride because of her wheelchair. https://tinyurl.com/yxea2r97. Accessed: 2019-06-12.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2018a. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. AAAI '18, 1007–1014.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2018b. Assigning tasks to workers based on historical data: Online task assignment with two-sided arrivals. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, 318–326.

Esfandiari, H.; Korula, N.; and Mirrokni, V. 2016. Bi-objective online matching and submodular allocations. In *Advances in Neural Information Processing Systems*, 2739–2747.

Fain, B.; Munagala, K.; and Shah, N. 2018. Fair allocation of indivisible public goods. In *Proceedings of EC*, 575–592. ACM.

Ghodsi, A.; Zaharia, M.; Hindman, B.; Konwinski, A.; Shenker, S.; and Stoica, I. 2011. Dominant resource fairness: Fair allocation of multiple resource types. In *NSDI*, volume 11, 24–24.

Grandoni, F.; Ravi, R.; and Singh, M. 2009. Iterative rounding for multi-objective optimization problems. In *European Symposium on Algorithms*, 95–106. Springer.

Griffin, K. 2019. Ont. woman says uber driver rejected her guide dog. https://tinyurl.com/y7sd268y. Accessed: 2019-06-12.

Houser, K. 2018. Uber and Lyft still allow racist behavior, but not as much as taxi services. https://tinyurl.com/yyln8e3v. Accessed: 2019-06-12.

Kanoria, Y., and Qian, P. 2019. Near optimal control of a ride-hailing platform via mirror backpressure. *arXiv preprint arXiv:1903.02764*.

Kash, I.; Procaccia, A. D.; and Shah, N. 2014. No agent left behind: Dynamic fair division of multiple resources. *JAIR* 51:579–603.

Laurence, L. 2019. Uber driver fired for refusing to drive woman to get abortion. https://tinyurl.com/y2cadftk. Accessed: 2019-06-12.

Lesmana, N. S.; Zhang, X.; and Bei, X. 2019. Balancing efficiency and fairness in on-demand ridesourcing. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc. 5310–5320.

Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; and Liu, Y. 2018. Multi-task representation learning for travel time estimation. KDD '18, 1695–1704.

Lin, K.; Zhao, R.; Xu, Z.; and Zhou, J. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. 1774–1783.

Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2018. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence* 261:71 – 112.

Ma, H.; Fang, F.; and Parkes, D. C. 2019. Spatio-temporal pricing for ridesharing platforms. EC '19, 583–583.

Parkes, D. C.; Procaccia, A. D.; and Shah, N. 2015. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM TEAC* 3(1):3.

Paul, K. 2018. There's a quiet battle of wills between uber drivers and customers over who cancels first. https://tinyurl.com/y8jx6dol. Accessed: 2019-06-12.

Ravi, R.; Marathe, M. V.; Ravi, S.; Rosenkrantz, D. J.; and Hunt III, H. B. 1993. Many birds with one stone: Multi-objective approximation algorithms. STOC '93, 438–447. Citeseer.

Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. WWW '13, 1157–1166.

Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. WWW '13, 1167–1178.

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. KDD '19, 3082–3092.

Tong, Y.; She, J.; Ding, B.; Chen, L.; Wo, T.; and Xu, K. 2016a. Online minimum matching in real-time spatial data: experiments and analysis. *Proceedings of the VLDB Endowment* 9(12):1053–1064.

Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016b. Online mobile micro-task allocation in spatial crowdsourcing. ICDE '16, 49–60.

Tong, Y.; Wang, L.; Zhou, Z.; Ding, B.; Chen, L.; Ye, J.; and Xu, K. 2017. Flexible online task assignment in real-time spatial data. *Proceedings of the VLDB Endowment* 10(11):1334–1345.

Wang, Z.; Fu, K.; and Ye, J. 2018. Learning to estimate the travel time. KDD '18, 858–866.

Xu, Z.; Li, Z.; Guan, Q.; Zhang, D.; Li, Q.; Nan, J.; Liu, C.; Bian, W.; and Ye, J. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. KDD '18, 905–913.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. AAAI '18, 2588–2595.

Zhao, B.; Xu, P.; Shi, Y.; Tong, Y.; Zhou, Z.; and Zeng, Y. 2019. Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. AAAI '19, 2245–2252.