

Using Simple Incentives to Improve Two-Sided Fairness in Ridesharing Systems

Ashwin Kumar, Yevgeniy Vorobeychik, William Yeoh

Washington University in St. Louis
 {ashwinkumar, yvorobeychik, wyeoh}@wustl.edu

Abstract

State-of-the-art order dispatching algorithms for ridesharing batch passenger requests and allocate them to a fleet of vehicles in a centralized manner, optimizing over the estimated values of each passenger-vehicle matching using integer linear programming (ILP). Using good estimates of future values, such ILP-based approaches are able to significantly increase the service rates (percentage of requests served) for a fixed fleet of vehicles. However, such approaches that focus solely on maximizing efficiency can lead to disparities for both drivers (e.g., income inequality) and passengers (e.g., inequality of service for different groups). Existing approaches that consider fairness only do it for naive assignment policies, require extensive training, or look at only single-sided fairness. We propose a simple incentive-based fairness scheme that can be implemented online as a part of this ILP formulation that allows us to improve fairness over a variety of fairness metrics. Deriving from a lens of variance minimization, we describe how these fairness incentives can be formulated for two distinct use cases for passenger groups and driver fairness. We show that under mild conditions, our approach can guarantee an improvement in the chosen metric for the worst-off individual. We also show empirically that our *Simple Incentives* approach significantly outperforms prior art, despite requiring no retraining; indeed, it often leads to a large improvement over the state-of-the-art fairness-aware approach in *both* overall service rate and fairness.

Introduction

In a ridesharing system, multi-capacity vehicles allow passengers to share rides with others or be added onto existing trips, and a single central agent aggregates all information and dynamically matches passenger requests to available vehicles. On-demand ridesharing has been gaining traction over the past few years as a solution to the growing need for urban mobility and, as a consequence, there has been extensive work on developing approaches for optimizing the efficiency of such systems. Recent approaches to this problem use dynamic programming and deep reinforcement learning to learn value functions for matching pools of passenger requests to available drivers, and have led to significant improvements in service rate (percentage of passenger requests served) as well as total passenger delay (Alonso-Mora et al.

2017; Shah, Lowalekar, and Varakantham 2020; Lowalekar, Varakantham, and Jaillet 2019; Li et al. 2019).

Optimizing the efficiency of the overall system, however, can lead to both geographic disparity in the quality of service for the passengers, potentially exacerbating historical inequalities, and income disparity of the drivers. While fairness in ridesharing has been a subject of some prior discussion (Nanda et al. 2020; Xu and Xu 2020), the particular issue of *geographic disparities* has received less attention. Raman, Shah, and Dickerson (2021) consider balancing overall efficiency and *either* geographic or income fairness, but this approach loses a great deal of overall efficiency, and requires complete retraining of the deep reinforcement learning model for any change in hyperparameters, such as the relative importance of fairness.

Instead of proposing yet another algorithm-specific approach to address fairness in ridesharing, we seek to develop a general framework that can be applied orthogonally to a large class of existing ridesharing approach. Specifically, we leverage the two-stage approach that many existing ridesharing algorithms (Alonso-Mora et al. 2017; Shah, Lowalekar, and Varakantham 2020; Lowalekar, Varakantham, and Jaillet 2021) rely on: (1) Identifying or learning good value functions for matches, and (2) Optimizing over those value functions to find a good match. While one could incorporate fairness in the value function learning stage, doing this would undermine our goal of generality since different ridesharing approaches employ different ways of identifying or learning their value functions. However, since all approaches employ similar ILP-based optimization approaches to find good matches, incorporating fairness in the second stage would result in the generality that we seek.

Therefore, in this paper, we propose *Simple Incentives* (SI), a general framework for including fairness in ILP-based matching, using any off-the-shelf value function approximations. The key ingredient in SI is a novel linear measure of relative disparity that can be associated with individuals based on their group membership (e.g, passengers' origin and destination pair or drivers' relative income level). We make the following contributions in this paper: (1) We show how we can derive a general form for this measure from the lens of minimizing variance in a metric of interest, following which we derive instantiations of passenger- and driver-side fairness functions. (2) We theoretically demon-

strate that our approaches, under mild assumptions, provably improve the service rate of passengers from the most historically disadvantaged region as well as improve the income of the driver with the least historical income. (3) We empirically demonstrate the generality of SI by showing that it significantly improves fairness for a variety of ridesharing algorithms with minimal compromises on service rate. Additionally, through extensive experiments, we show that the best SI variant achieves significantly greater fairness than the state-of-the-art fair ridesharing approach, while at the same time yielding overall system efficiency (measured by the service rate) that is nearly inline with a state-of-the-art approach that maximizes efficiency and ignores fairness. (4) We show that our passenger- and driver-side approaches can be combined to improve two-sided fairness in such systems. (5) Finally, unlike the existing state of the art (Raman, Shah, and Dickerson 2021), our approach is completely online, allowing ridesharing operators or policy makers to tune the tradeoff between fairness and efficiency in real time during execution.

Related Work

Order dispatching in ridesharing: Rideshare-matching has been extensively studied, and researchers have introduced methods that improve the quality of the matches made in terms of increasing the number of requests matched (Lowalekar, Varakantham, and Jaillet 2021; Ma, Zheng, and Wolfson 2015), reducing the pickup and detour delays (Alonso-Mora et al. 2017; Huang et al. 2014), and increasing drivers’ earnings (Lesmana, Zhang, and Bei 2019). The complexity of ridesharing algorithms increases with the increase in vehicle capacity and fleet size. As the runtime of real-time algorithms need to be relatively small, most existing work has either considered assigning one request at a time (sequentially) to available drivers for high capacities (Ma, Zheng, and Wolfson 2015; Tong et al. 2018) or assigning all active requests together in a batch for a small capacity (Yu and Shen 2019; Zheng, Chen, and Ye 2018). The sequential solution is faster to compute but the solution quality is typically poor (Uber 2018). Alonso-Mora et al. (2017) proposes ILP optimization approaches for assigning all active requests together for high-capacity ridesharing. Shah, Lowalekar, and Varakantham (2020) and Lowalekar, Varakantham, and Jaillet (2021) further improve these approaches by including information about anticipated future requests while matching current batch of requests to available drivers.

Fairness in ridesharing: Researchers have evaluated ridesharing fairness from many viewpoints. For passengers, there has been work on addressing lack of transparency (Wolfson and Lin 2017), using game-theoretic approaches to fairness (Foti, Lin, and Wolfson 2021), and benefit sharing by ensuring non-increasing disutility (Gopalakrishnan, Mukherjee, and Tulabandhula 2016). Driver-side fairness has also been explored from the economic perspective, by using a max-min approach to fairness to balance efficiency and fairness (Lesmana, Zhang, and Bei 2019), and by looking at fairness over longer periods of time by equaliz-

ing driver income proportional to the number of hours spent on the platform (Sühr et al. 2019). Fairness isn’t restricted to monetary benefits, however. Motivated by demographic and geographic fairness concerns, recent work formulates a matching problem with parameters to trade profit for fairness in terms of discriminatory cancellations, looking at factors like start/end locations, race, gender, or age of passengers (Nanda et al. 2020) and drivers (Xu and Xu 2020).

A work that is closest to ours is by Raman, Shah, and Dickerson (2021), which looks at disparate treatment of passengers and income disparity amongst drivers. While they also look at geographic zones to quantify fairness for passengers, their approach requires the training of a neural network based value function to include the fairness term in the objective, making it costly to change parameters for fairness. Our approach presents an online way to address this problem, without retraining existing value functions. Further, our approach offers better tradeoffs between efficiency and fairness as compared to the existing approach, and we show this in our empirical evaluation.

Dynamic Matching in Ridesharing Settings

A matching algorithm for ridesharing receives as inputs a continuous stream of batches of requests from passengers \mathcal{R} and the current state of all the taxis \mathcal{V} operating in a street network \mathcal{G} . The street network $\mathcal{G} = \langle \mathcal{L}, \mathcal{E}, c(e) \rangle$ is a graph containing locations \mathcal{L} connected by roads \mathcal{E} , with a cost function $c : \mathcal{E} \rightarrow \mathbb{R}^+$ that defines the cost $c(e)$ of each edge $e \in \mathcal{E}$ in the graph, which commonly corresponds to the time needed by a taxi to traverse e . A request is a tuple $r = \langle q, d, t \rangle$ that contains the pickup location q , dropoff location d , and the request arrival time t . The vehicle i is associated with a state $v_i = \langle l_i, p_i, c_i, U_i \rangle$ that includes its location l_i , current path p_i , capacity c_i , and the set of requests U_i it is currently serving.

State-of-the-art approaches to this dynamic matching problem take a discrete-time multi-agent perspective (Alonso-Mora et al. 2017; Shah, Lowalekar, and Varakantham 2020). In this framework, a decision-maker is faced in each time step with a set of outstanding requests, along with the state of vehicles at that point in time. The set of actions of this decision-maker is the set of all feasible matches of requests to vehicles. Since any matching impacts the distribution of future requests (unmatched requests can spill over into the next time step) and vehicle states, the associated expected future value is captured by a value function. This allows the decision-maker to make non-myopic decisions by considering expected rewards in addition to immediate rewards. To manage problem complexity, a value function is associated with individual vehicles and assumed to be identical for all vehicles, conditional on vehicle state; furthermore, it is approximated by a *Value Function Approximator* (VFA) – most recently, by a deep neural network in the NeurADP framework (Shah, Lowalekar, and Varakantham 2020).

A key workhorse in such approaches is an integer linear program (ILP) for computing the optimal matching of requests to vehicles in a given time step, given a value function. We now formalize the general form of this ILP, which

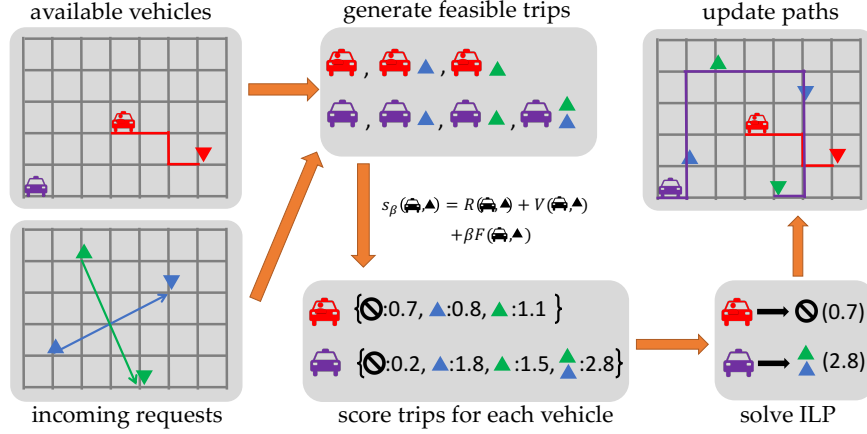


Figure 1: The rideshare-matching pipeline. Every time step, based on incoming requests and available vehicles, feasible trips for each vehicle are generated. These are assigned values using a score function, and an ILP is solved to optimize for total score.

is central to our proposed approach below. For each vehicle $i \in \mathcal{V}$, we define an *action* $a \subseteq \mathcal{R}$ as a subset of requests that could be matched to it. Each action a of vehicle i is associated with a (discounted) value $V(i, a)$ obtained using VFA, and an immediate reward:

$$R(i, a) = \sum_{r \in a} R(i, r) \quad (1)$$

where $R(i, r)$ is an immediate reward to vehicle i for servicing request r . Let A_i be the set of feasible actions of vehicle i (i.e., all feasible subsets of requests that can be matched to i). Then, for any feasible action $a \in A_i$ of vehicle i , we define the score of vehicle i associated with action a as:

$$s(i, a) = V(i, a) + R(i, a). \quad (2)$$

Let $\mathcal{A} = \{a_i\}_{i \in \mathcal{V}}$, $a_i \cap a_j = \emptyset$ for all $i \neq j$ be a matching of requests to vehicles. We say that this matching is optimal if it maximizes the total score $\sum_i s(i, a_i)$. We can compute such an optimal matching by solving the following ILP:

$$\max_{x_i(a) \in \{0,1\}} \sum_{i \in \mathcal{V}} \sum_{a \in A_i} x_i(a) s(i, a) \quad \text{s.t.} \quad (3)$$

$$\sum_{a \in A_i} x_i(a) = 1 \quad \forall i \in \mathcal{V} \quad (4)$$

$$\sum_{i \in \mathcal{V}} \sum_{a \in A_i | r \in a} x_i(a) \leq 1 \quad \forall r \in \mathcal{R} \quad (5)$$

where $x_i(a)$ is an indicator variable associated with vehicle i and its action $a \in A_i$. The constraints ensure that each vehicle is assigned exactly one action (Eq. 4) and no request is assigned to more than one vehicle (Eq. 5). In each vehicle's set of available actions, there is always the null action (i.e., accepting no new requests), so that there is always a solution. The final assignment \mathcal{A} is a concatenation of all vehicle assignments. In existing approaches (Shah, Lowalekar, and Varakantham 2020), this ILP serves to maximize the total expected number of requests served (i.e., the *service rate* when normalized by the total number of requests), and it can be readily generalized to maximize total expected profit. Figure 1 shows an illustration of the ILP matching process.

SI: A General Fairness Framework

By focusing on maximizing measures of efficiency such as service rate, traditional matching algorithms for ridesharing fail to account for the emergence of fairness issues for both the passengers and drivers. Additionally, since matching algorithms in ridesharing are often trained on real-world data, they can further exacerbate historical inequalities. For example, from a passenger's perspective, it is conceivable that while maximizing efficiency, matching algorithms focus resources towards high-traffic areas (e.g., downtown locations) while reducing service in low-traffic areas (e.g., suburbs). If such algorithms are deployed in practice, it may also lead to a feedback loop that continually reduces service to regions with low demand, which causes fewer passengers to make requests from such regions and so on. From a driver's perspective, the matching algorithms may prioritize drivers in high-traffic areas to continually service requests from those areas over drivers in low-traffic areas, leading to disparity in driver income. In this work, our goal is to tackle this kind of unfairness with minimal impact to the efficiency vis-a-vis the service rate.

We consider *statistical parity* (Feldman et al. 2015; Agarwal, Dudík, and Wu 2019) as the notion of fairness in this work, which defines that the expected value of a given metric z over a group g is the same as $\bar{z} = \mathbb{E}_{g' \in G} [z(g')]$ the expected value of that same metric over all groups $g' \in G$:

$$z(g) = \bar{z} \quad (6)$$

More generally, we can write it as:

$$|\bar{z} - z(g)| \leq \epsilon \quad \forall g \in G \quad (7)$$

where ϵ is a slack parameter.

As an example, let's say that we want parity in service rates for passenger groups defined by their origin-destination pairs and in normalized income for drivers. If a matching algorithm achieves statistical parity for both passengers and drivers, then the probability of a passenger receiving a ride is the same regardless of their origin and destination locations and the income of drivers are the same for all drivers.

While the goal of achieving parity is noble, achieving it through the matching of a single time step is rarely possible, especially when there is a large disparity. Instead, it is often better to look at amortized parity over a longer period of time (Sühr et al. 2019). To do this, we aim for a matching that “moves closer” towards parity with the goal of achieving parity in the near future. Towards that end, our framework uses variance $\text{var}(\mathbf{Z})$, where $\mathbf{Z} = \{z(g), \forall g \in G\}$ is the set of metric values for all groups, as a proxy measure for fairness and, at each time step, it takes a gradient step in the solution space, moving in the direction that minimizes variance.

If we assume that the average of the metric over all groups is stable (i.e., $\frac{\partial}{\partial \mathcal{A}} \bar{z} \simeq 0$, a reasonable assumption if a long enough history is included), then we can find an assignment for a modified score function that accounts for the gradient of the variance with respect to the assignment \mathcal{A} :

$$s'(i, a) = s(i, a) - \lambda \frac{\partial}{\partial \mathcal{A}} \text{var}(\mathbf{Z}) \quad (8)$$

$$= s(i, a) - \frac{1}{|\mathbf{Z}|} \lambda \frac{\partial}{\partial \mathcal{A}} \sum_{z_j \in \mathbf{Z}} (z_j - \bar{z})^2 \quad (9)$$

$$= s(i, a) + \frac{2}{|\mathbf{Z}|} \lambda \sum_{z_j \in \mathbf{Z}} (\bar{z} - z_j) \frac{\partial z_j}{\partial \mathcal{A}} \quad (10)$$

where λ is a hyperparameter. The general form above for the second term is our *incentive* score: A constant (weight) multiplied by the disparity of group j , scaled by a derivative term. We show later that the sum can usually be simplified within the context of a given action a and the derivative can be approximated for specific metrics, including our two passenger- and driver-side fairness metrics of interest.

The “Simple Incentives” idea is that, for each group involved in an action, provide them with an incentive (or penalty) proportional to how disadvantaged (or advantaged) their group has been historically. Given the recent abundance of black-box algorithms, we find this simplicity helpful from a transparency perspective, making it easy to explain to any stakeholder how this score is calculated.

SI(+): While there is generally a consensus for applying incentives to help disadvantaged groups, it may be controversial to impose penalties for advantaged groups in some applications. With this in mind, we also present a modification to SI, where we clip the incentive term to be larger than 0, resulting in the SI(+) variant, where the “+” indicates that we include only positive incentives.

In the following sections, we discuss how we specify the incentive score (see Eq. 10) for two use cases, one for passengers and one for drivers.

SI for Passengers (SIP): Geographic Fairness

Our notion of passenger-side fairness is defined by the idea that the probability of receiving a ride should not depend on your origin or destination. To do this, we divide the geographical area served by the fleet into a collection of areas C . Recall that each request r contains both the origin o and destination d . Thus, we can map each passenger to one of $C \times C$ groups, uniquely identified by the origin-destination

area pair based on (o, d) . Computing the service rate for each of these groups gives us the *passenger*-side metric set \mathbf{Z}_p , where $z_i \in \mathbf{Z}_p$ denotes the *historical* service rate for passenger group i . For a given request r , let $g(r)$ denote the group to which r belongs, and let $z(r) = z_{g(r)}$ (i.e., the historical service rate of the group $g(r) \in C \times C$). These historical service rates are updated every time step after an assignment is made. The goal is thus to achieve parity in the service rates for all geographic groups.

We now derive the fairness incentive for this metric, based on Eq. 10:

$$s'(i, a) = s(i, a) + \frac{2}{|\mathbf{Z}_p|} \lambda \sum_{z_j \in \mathbf{Z}_p} (\bar{z} - z_j) \frac{\partial z_j}{\partial \mathcal{A}} \quad (11)$$

$$\simeq s(i, a) + \frac{2}{|\mathbf{Z}_p|} \lambda \sum_{z_j \in \mathbf{Z}_p} (\bar{z} - z_j) \frac{\partial z_j}{\partial a} \quad (12)$$

$$= s(i, a) + \frac{2}{|\mathbf{Z}_p|} \lambda \sum_{r \in a} (\bar{z} - z(r)) \frac{\partial z(r)}{\partial a} \quad (13)$$

$$\simeq s(i, a) + \frac{2}{|\mathbf{Z}_p|} \lambda \sum_{r \in a} (\bar{z} - z(r)) \lambda' \quad (14)$$

$$= s(i, a) + \beta \sum_{r \in a} (\bar{z} - z(r)) \quad (15)$$

Notice that computing $\frac{\partial z_j}{\partial \mathcal{A}}$ is difficult due to circular dependencies: The matching \mathcal{A} depends on the score function, which depends on the service rates, which in turn depend on the matching. As such, computing it precisely will require a global optimization procedure. Therefore, we approximate it with $\frac{\partial z_j}{\partial a}$ (from Eqs. 11 to 12) based on the assumption that the change in the service rate of zone z_j that is based on action a is independent of the actions of other vehicles. Additionally, as serving a single request r can only make small changes to the service rate $z(r)$ because service rates are aggregated over a reasonably long time period, we assume that the change is a (positive) constant λ' for all requests (from Eqs. 13 to 14). We also introduce a *fairness incentive* term $F_p(i, a)$ to represent the summation in the second term of Eq. 15 and use it in the new approximated score function s_β :

$$F_p(i, a) = \sum_{r \in a} (\bar{z} - z(r)) \quad (16)$$

$$s_\beta(i, a) = s(i, a) + \beta F_p(i, a) \quad (17)$$

Here, β is a hyperparameter that controls the relative importance of geographic fairness for passengers. In summary, for our SIP score function s_β , we include a fairness incentive term $F_p(i, a)$ that increases (or decreases) the score of actions proportional to how disadvantaged (or advantaged) their respective passenger groups have been according to our geographic fairness metric.

As described earlier, we also have the corresponding SI(+) version of this passenger-side incentive, which takes into account only requests from groups with below-average service rates:

$$F_p(i, a) = \sum_{r \in a} \max\{\bar{z} - z(r), 0\} \quad (18)$$

We call this variant of the framework SIP(+).

SI for Drivers (SID): Income Fairness

Our notion of driver-side fairness is defined by the idea that all drivers should earn similar incomes. This makes our driver-side fairness metric \mathbf{Z}_d the set of all z_i , where z_i now denotes *historical* driver incomes for each driver i . We scale the driver incomes by the largest driver income to restrict it to within $[0, 1]$ and avoid scaling issues. The goal here is thus to achieve parity in the scaled income for all drivers.

We now derive the fairness incentive for this metric, based on Eq. 10, where we model the income of a driver to be proportional to the value of the request they serve $R(i, r)$:

$$s'(i, a) = s(i, a) + \frac{2}{|\mathbf{Z}_d|} \lambda \sum_{z_j \in \mathbf{Z}_d} (\bar{z} - z_j) \frac{\partial z_j}{\partial \mathcal{A}} \quad (19)$$

$$= s(i, a) + \frac{2}{|\mathbf{Z}_d|} \lambda \sum_{z_j \in \mathbf{Z}_d} (\bar{z} - z_j) \frac{\partial z_j}{\partial a} \quad (20)$$

$$= s(i, a) + \frac{2}{|\mathbf{Z}_d|} \lambda (\bar{z} - z_i) \frac{\partial z_i}{\partial a} \quad (21)$$

$$= s(i, a) + \delta (\bar{z} - z_i) R(i, a) \quad (22)$$

Eqs. 19 and 20 are equivalent because, unlike the case in passenger-side fairness, computing $\frac{\partial z_j}{\partial \mathcal{A}}$ when the metric is driver income is straightforward as the income of a driver depends solely on their actions. Next, Eqs. 20 to 22 are equivalent because $\frac{\partial z_j}{\partial a} = R(i, a)$ when $j = i$ and is 0 otherwise.

Analogous to the case for passenger-side fairness, we also introduce a similar fairness incentive term $F_d(i, a)$ and use it in the new score function s_δ :

$$F_d(i, a) = (\bar{z} - z_i) R(i, a) \quad (23)$$

$$s_\delta(i, a) = s(i, a) + \delta F_d(i, a) \quad (24)$$

Here, δ is a hyperparameter that controls the relative importance of income fairness for drivers.

We also have the variant that we call SID(+), where the fairness incentive is applied only to drivers with below-average income:

$$F_d(i, a) = \sum_{r \in a} \max\{\bar{z} - z_i, 0\} R(i, r) \quad (25)$$

Theoretical Properties

While our approach directly works to minimize variance in the selected metric across groups, *max-min* fairness is another popular notion of fairness, where we want to maximize the worst-off group. Towards that end, given a sufficient large weight (β for passenger-side fairness and δ for driver-side fairness), our approaches provide guarantees for improving worse-off groups. We provide proof sketches here for the two metrics discussed in the previous section, with complete proofs our supplemental document.¹ Both proofs assume that vehicles can only accept a single request at any time step (including the null action). With this assumption, each action a contains only one request r , allowing us to use them interchangeably.

¹The code and supplementary material can be found at: <https://github.com/YODA-Lab/Simple-Incentives-For-Ridesharing>

Let $\mathcal{A}(w)$ be the matching solution produced by the ILP with fairness weight $w \in \{\beta, \delta\}$. Further, let $z'_j(w)$ denote the updated metric value of group j after an assignment $\mathcal{A}(w)$; and $g_{\min} = \operatorname{argmin}_j z_j$ denote the group with the smallest metric value. We now describe the theoretical properties for each of our two fairness metrics below.

Passenger-side Fairness Properties

Let $\mathcal{R}_f = \{r\}_{g(r)=g_{\min}}$ denote the set of current requests corresponding to group g_{\min} . Any request $r \in \mathcal{R}_f$ would have the largest fairness incentive by definition (Eq. 16).

In what follows, we state the theorem, claiming that as long as it is possible to improve the service rate of $z_{g_{\min}}$ relative to $\mathcal{A}(0)$ – a condition that we formalize as *passenger-min-unfairness* in the following definition – we can do so for a sufficiently high β .

Definition 1. A matching \mathcal{A} is passenger-min-unfair if there exists a request $r_f \in \mathcal{R}_f$ not served in \mathcal{A} , there is a vehicle $i \in \mathcal{V}$ such that $r_f \in A_i$, but the request assigned to i is $r_i \notin \mathcal{R}_f$.

Theorem 1. If $\mathcal{A}(0)$ is passenger-min-unfair, then there exists $\beta > 0$ such that $z'_{g_{\min}}(\beta) > z'_{g_{\min}}(0)$.

Proof Sketch: It can be shown that:

1. The *highest-preferred* request of vehicle i , $r_i^* = \operatorname{argmax}_{r \in A_i} s_\beta(i, r)$ will always be assigned to some vehicle in the optimal assignment. This follows from the nature of the ILP, which maximizes the total score.
2. Any vehicle that prefers (see (1)) $r_f \in \mathcal{R}_f$ also prefers it for higher β values. This can be proven using the fact that r_f has the largest fairness incentive, and increasing β only increases the contribution of the fairness incentive towards $s_\beta(i, a)$.
3. There is a threshold $\bar{\beta}$ value such that for higher β values, the $\beta F_p(\cdot)$ term in $s_\beta(i, a)$ dominates the other factors (i.e., any action a_1 with $F_p(i, a_1) > F_p(i, a_2)$ will be preferred by vehicle i over action a_2). This can be shown if we assume some bounds on $R(\cdot)$ and $V(\cdot)$.
4. Any request r_f that was assigned in $\mathcal{A}(0)$ will also be assigned in $\mathcal{A}(\beta)$, where $\beta > \bar{\beta} \geq 0$. This follows from a combination of the previous points.

If $\mathcal{A}(0)$ is passenger-min-unfair, then we can conclude that there is some request $r \in \mathcal{R}_f$ that was not preferred in $\mathcal{A}(0)$ but will now be preferred (3) and assigned to some vehicle (1). Thus, $\mathcal{A}(\beta)$ assigns all requests in \mathcal{R}_f that were assigned in $\mathcal{A}(0)$ and at least one more, thus increasing the service rate for g_{\min} as compared to $\mathcal{A}(0)$. \square

Driver-side Fairness Properties

For drivers, note that they would see a highest improvement in income after receiving an action with the highest immediate reward. In what follows, we state the theorem, claiming that, for SID(+) with high enough δ , any worse-off driver j ($z_j < \bar{z}$) is guaranteed to get their highest preferred request r_j^* as long as no other worse-off driver can serve it.

Definition 2 (driver-min-unfair). A matching \mathcal{A} is driver-min-unfair if any worse-off driver j is assigned a request

$r \neq r_j^*$ in \mathcal{A} and there exists no other worse-off driver k that can serve r_j^* .

Theorem 2. *If $\mathcal{A}(0)$ is driver-min-unfair for any driver j , then there exists $\delta > 0$ such that $z_j'(\delta) > z_j'(0)$, when using $SID(+)$.*

Proof Sketch: It can be shown that:

1. For a worse-off driver j , their highest-preferred request r_j^* will have the highest score for some value of δ , and will be assigned in $\mathcal{A}(\delta)$ to some driver.
2. Any better-off driver that can serve r_j^* will not serve it for some large value of δ .

If $\mathcal{A}(0)$ is driver-min-unfair, then we can conclude that r_j^* will be assigned to j for some large value of δ because no other worse-off driver can serve it; no better-off driver can serve it (2); and it must be served by some driver (1). Thus, $z_j'(\delta) > z_j'(0)$ because j is getting their highest-preferred request r_j^* for some large $\delta > 0$, but not when $\delta = 0$. \square

Experimental Evaluations

To comprehensively evaluate our SI framework, we ran three sets of experiments. First, to demonstrate the generality of the framework, we evaluated it in combination with various ridesharing algorithms, which use different value function approximations, from the literature. Second, to demonstrate the competitiveness of the framework, we evaluated it against a state-of-the-art ridesharing fairness approach. Finally, to demonstrate the flexibility of the framework, we evaluated it on two-sided fairness with both of our passenger- and driver-side fairness approaches combined.

We evaluate the performance and fairness metrics after running the matching algorithms over a 24-hour period on the island of Manhattan using demand data from the NY Yellow Taxi dataset (NYC Taxi & Limousine Commission 2020). The locations in the road network correspond to street intersections, with edges as roads connecting them. We define areas with respect to a standard partition of Manhattan into neighborhoods, where passenger groups correspond to pairs of areas in which passengers are to be picked up and dropped off. Our efficiency objective is to maximize service rate, and thus, correspondingly, we set the value of each request to 1. For each hyperparameter β and δ , we performed a logarithmic search in the $[0.5, 20]$ range to capture a wide range of behaviors. Consistent with literature (Shah, Lowalekar, and Varakantham 2020; Lowalekar, Varakantham, and Jaillet 2021; Raman, Shah, and Dickerson 2021), we use a fleet size of 1000 vehicles, with a maximum request waiting time of 300 seconds. In all approaches, any request not assigned in the current assignment (one minute window) is dropped. All experiments were run on a Ryzen 3700x CPU, RTX2080 Super GPU, and 32GB RAM.¹

We consider two standard measures of equity: The *Gini* coefficient $Gini(\mathbf{Z})$ and the minimum metric value $\min(\mathbf{Z})^2$ for a set \mathbf{Z} of service rates for passenger-side fairness or income for driver-side fairness. For ease of comprehension, we

²For drivers, we plot the minimum income (unscaled), to prevent bias because of scaling by the max.

plot $F_{Gini}(\mathbf{Z}) = 1 - Gini(\mathbf{Z})$ so that our goal is to maximize each of these metrics. Our efficiency measure is the overall service rate, defined as the fraction of all passenger requests served.

Generality: Evaluation on Benchmark Algorithms

We tested the efficacy of our framework on a variety of natural and state-of-the-art ridesharing algorithms:

- **Greedy:** A baseline algorithm that considers only current rewards $R(i, a)$ as the score in the ILP optimization.
- **R+D:** A pioneering approach proposed by Alonso-Mora et al. (2017) that estimates future values by using delays for passengers, as part of the score function in the ILP optimization.
- **NeurADP:** A state-of-the-art approach by Shah, Lowalekar, and Varakantham (2020) that uses deep reinforcement learning to approximate the value function, as part of the score function in the ILP optimization. We use a pre-trained model trained with 1000 vehicles.
- **AsyncNeural:** An asynchronous and distributed baseline algorithm that uses the approximated value function from NeurADP, but each vehicle greedily chooses its action and ties in vehicle order are broken randomly. (No ILP.)

Figure 2 show the impact of SI on our four ridesharing algorithms. The origin of each arrow corresponds to the performance of an algorithm without SI and the arrowhead corresponds to the best performance with SI while restricting the service rate to be above 95% of the service rate without SI. Figures 2(a) and 2(b) show the results for passenger-side fairness for the F_{Gini} and min metrics, respectively; while Figures 2(c) and 2(d) show the results for driver-side fairness for those two metrics as well.

We observed that our methods can provide significant improvement in fairness at a marginal impact to service rates. Even with varying degrees of initial fairness, consistent improvement for both passengers and drivers can be seen. The one outlier is that SI failed to improve driver fairness for AsyncNeural. The reason is that, for each driver, their fairness incentive scaling term in $F_d(\cdot)$ for all actions are identical. Therefore, the incentive does not affect their preference ordering of requests to serve, and the outcome of the algorithm remains unchanged with and without the fairness incentive term. On the other hand, with the other centralized approaches, the ILP is able to arbitrate between different vehicles and prioritize low-income drivers.

Competitiveness: Comparisons with FairNN

As our fairness baseline, we use *FairNN*, a recent fairness extension of NeurADP (Raman, Shah, and Dickerson 2021). Like our SI framework, FairNN also considers geographic fairness for passengers *or* income fairness for drivers, but unlike SI, it cannot consider both together. It does this by minimizing the variance in service rates or incomes, learning a neural network-based VFA to do so. As it follows a similar ILP formulation to solve the optimization problem, it is well suited for comparison with SI. It also has a hyperparameter λ that controls the scale of the variance term in the objective, similar to β or δ in our formulation.

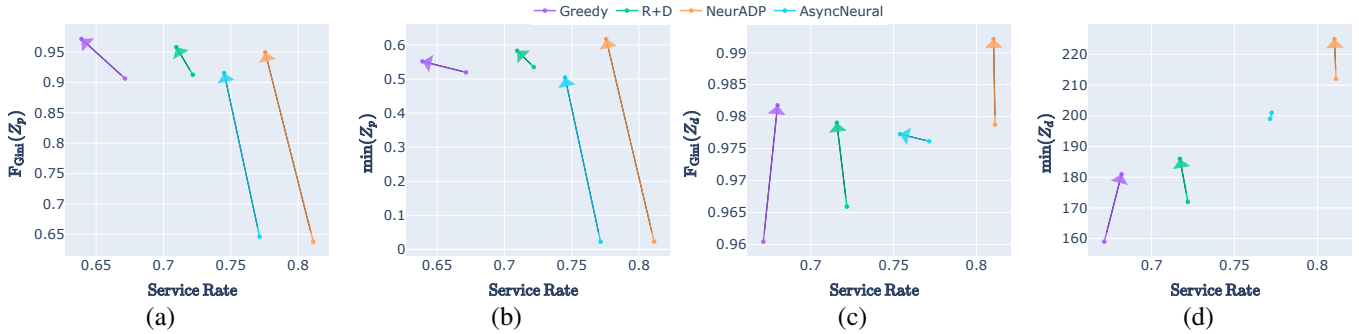


Figure 2: Change in efficiency and passenger-side fairness (with (a) F_{Gini} and (b) \min) as well as driver-side fairness (with (c) F_{Gini} and (d) \min) when Simple Incentives are used. The arrows show the best improvement for each algorithm metric while limiting SR to remain above 95% of the initial value. Optimal point is to the top-right (high fairness & high service rate).

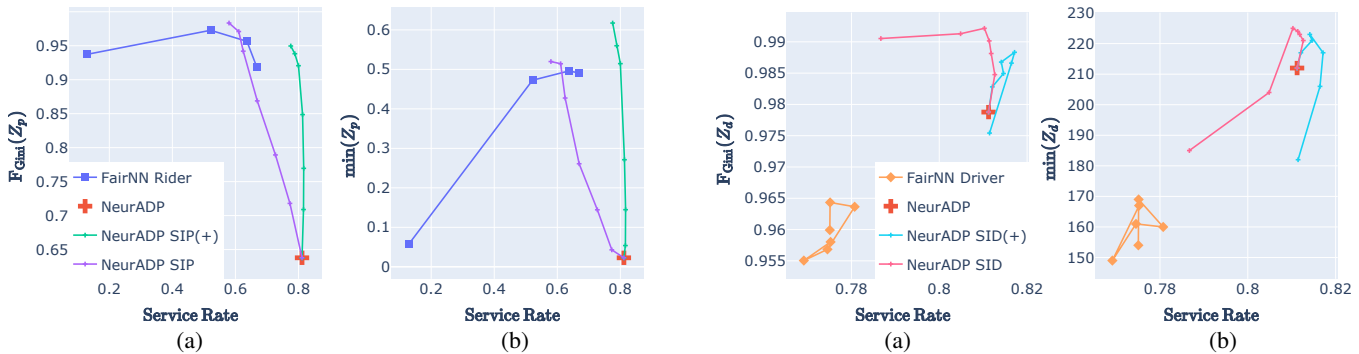


Figure 3: Comparison of SIP & SIP(+) against NeurADP & FairNN (line plotted in the order of hyperparameter values) for passenger-side fairness with (a) F_{Gini} and (b) \min . Optimal point is to the top-right (high fairness & service rate).

We note below some key differences between our approach and FairNN: (1) FairNN directly includes variance in the optimization, but as history size increases, the change in variance per action diminishes, reducing the impact of the fairness term if a constant weight is used. SI computes the scores based on historical inequalities rather than marginal contribution, which, combined with metric scaling, allows our approach to work well even with large histories. (2) FairNN applies the fairness uniformly across all actions, and we can improve *both* overall efficiency and fairness by adding further flexibility to the objective (as shown below with SI(+)). (3) Possibly most important of all, this approach requires full retraining of VFA for even a small change in the tradeoff weight, or a change in any other problem parameters (such as the particular measure of fairness used), making it difficult to scale in practice. Our approach is completely online, and can be used with any pre-existing value function, and with any hyperparameter value.

We trained FairNN using their provided code, suggested parameter values, and suggested hyperparameter values λ from $10^7 - 10^{10}$ for passengers and $\frac{1}{6} - \frac{6}{6}$ for drivers, each of which requires costly retraining. For SI, we use the pre-trained NeurADP value function described in the previous

Figure 4: Comparison of SID & SID(+) against NeurADP & FairNN (line plotted in the order of hyperparameter values) for driver-side fairness with (a) F_{Gini} and (b) \min . Optimal point is to the top-right (high fairness & service rate).

section. NeurADP (without fairness) also acts as our efficiency baseline.

Baseline Comparisons for Passenger-side Fairness: Figures 3(a) and 3(b) compare SIP and SIP(+), with different β values, against both baselines. FairNN outperforms SIP when the service rate is small (~ 0.65), but SIP(+) significantly outperforms FairNN by achieving similar fairness with much higher service rates. Consistent with results in Figure 2, SIP and SIP(+) improves the fairness of NeurADP as β increases.

Baseline Comparisons for Driver-side Fairness: Figures 4(a) and 4(b) compare SID and SID(+), with different δ values, against NeurADP. FairNN for drivers had poor efficiency compared to NeurADP, which is consistent with observations by the authors (Raman, Shah, and Dickerson 2021). Performing better than FairNN, SID and SID(+) improve the fairness of NeurADP as δ increases. Analogous to the passenger-side variants, SID(+) outperforms SID.

Relative Performance of our Variants: For both drivers and passengers, we observed the SI(+) variants provided better tradeoffs between fairness and efficiency. The reason is that these variants improve fairness without signifi-

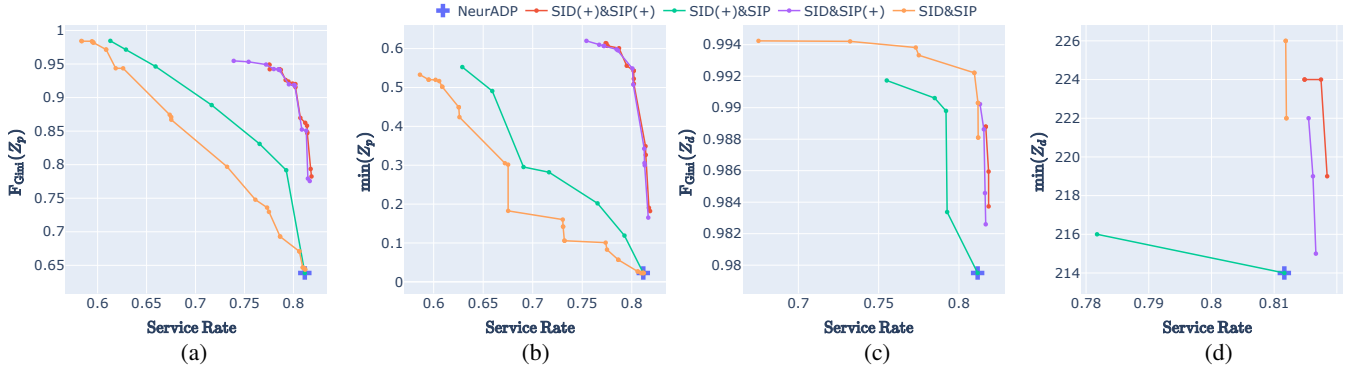


Figure 5: Comparison of Pareto frontiers of our combined variants on efficiency vs passenger-side fairness (with (a) F_{Gini} and (b) min) and driver-side fairness (with (c) F_{Gini} and (d) min). Optimal point is to the top-right (high fairness & service rate)

cantly sacrificing efficiency since high-efficiency groups are not penalized. The base variants achieved much better fairness in the extreme, albeit at a higher cost to service rate. The reason is that if extreme fairness is required, then there may be no choice but to penalize high-efficiency groups.

Ablation Experiments: Two-Sided Fairness

Finally, we evaluate SI with NeurADP’s value function on two-sided fairness with both passenger- and driver-side fairness approaches combined. Specifically, we ran ablation experiments, running all combinations of β and δ values in a logarithmic grid search for the score function:

$$s_{\beta, \delta}(i, a) = s(i, a) + \beta F_p(i, a) + \delta F_d(i, a) \quad (26)$$

Figure 5 shows the Pareto frontier for each passenger- and driver-side fairness combination. In general, we observe the trend that using SIP leads to much lower service rates as compared to SIP(+). SID and SID(+) with SIP(+) are almost equivalent, Pareto dominating the other approaches, with the exception of extremely high fairness regions. SIP and SID combined resulted in the fairest algorithm, albeit at a high cost to service rate.

Typically, one would expect a tradeoff between fairness and service rate. However, out of 175 combinations of β and δ hyperparameter values as well as SIP/SIP(+) and SID/SID(+) pairs, 15 combinations ($\sim 8.5\%$) outperformed NeurADP on *all* five metrics (four fairness metrics and service rate). A commonality across all 15 combinations is that they all have small hyperparameter values with $\beta \leq 2$ and $\delta \leq 2$. Further, 41 out of 175 combinations ($\sim 23.5\%$) outperformed NeurADP on four out of five metrics, with $\beta \leq 2$ and $\delta \leq 10$. This observation reinforces the idea that *fairness does not have to be a trade off* and, in many cases, improving fairness and efficiency can go hand-in-hand.

Discussion and Conclusions

As the demand for cutting-edge algorithms for urban mobility increases, their effects on the underlying fairness of these systems need to be studied, and measures taken to ensure that the algorithms do not inherit implicit biases that result from pure optimization. In this work, we focused on

the issue of fairness in ridesharing systems, specifically geographic fairness for passengers and income fairness for drivers. We proposed Simple Incentives (SI), a general fairness framework that can be adapted to a class of existing ridesharing algorithms that use an integer linear program to find matches. At a high level, SI applies fairness incentives to groups of passenger requests and drivers based on their service rate and income disparity. Under mild assumptions, SI provably improves the service rate of passengers from the group with the worst service rate as well as the income of driver with the lowest income. Our experimental results demonstrated its generality by showing that it improves the fairness for several ridesharing benchmarks and its competitiveness by outperforming existing state of the art in terms of both efficiency (measured through overall service rates) and fairness (measured through the Gini coefficient and the minimum service rates and driver income) despite requiring no retraining. Our experiments showed that it is better to apply fairness incentives only to requests from passenger groups with below-average service rates and to drivers with below-average income, as opposed to a blanket approach that applies fairness incentives to all passenger requests and drivers. Finally, interestingly, our results showed that it is possible to improve both efficiency and fairness in some cases.

Limitations: We do not prescribe “best” hyperparameters as that decision is subjective and will depend on the use-case. Instead, we provide an easy way to tune the importance of fairness using the hyperparameters. We selected geographic fairness because of the lack of other protected demographic information of passengers in public datasets. However, our formulation is general enough to allow any group-based division of passengers (e.g., by race and/or gender). Our income fairness does not take into account the number of hours worked by drivers as we assume that all drivers work all 24 hours, consistent with the literature. Finally, SI is a myopic approach to fairness. We may expect to see a non-myopic approach perform better when system dynamics allow for making suboptimal actions in the present that could lead to better fairness returns in the future. However, we expect these situations to be infrequent in practice.

Ethics Statement

This work has the potential to affect livelihoods of taxi drivers in real ridesharing systems. We do not intend this work to be directly translated to application without sufficient testing to ensure that the effects on drivers' and other stakeholders' livelihood are not adverse.

Acknowledgements

This research is partially supported by the National Science Foundation under awards 1812619, 1905558, 1939677, 2020289, 2118201, and 2214141; by ARO grant W911NF1810208; and by Amazon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the United States government.

References

- Agarwal, A.; Dudík, M.; and Wu, Z. S. 2019. Fair Regression: Quantitative Definitions and Reduction-based Algorithms. In *International Conference on Machine Learning*, 120–129.
- Alonso-Mora, J.; Samaranayake, S.; Wallar, A.; Frazzoli, E.; and Rus, D. 2017. On-demand High-capacity Ride-sharing via Dynamic Trip-vehicle Assignment. *Proceedings of the National Academy of Sciences*, 114: 462–467.
- Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and Removing Disparate Impact. In *ACM International Conference on Knowledge Discovery & Data Mining*, 259–268.
- Foti, L.; Lin, J.; and Wolfson, O. 2021. Optimum versus Nash-equilibrium in Taxi Ridesharing. *GeoInformatica*, 25: 423–451.
- Gopalakrishnan, R.; Mukherjee, K.; and Tulabandhula, T. 2016. The Costs and Benefits of Ridesharing: Sequential Individual Rationality and Sequential Fairness. *arXiv:1607.07306*.
- Huang, Y.; Bastani, F.; Jin, R.; and Wang, X. S. 2014. Large Scale Real-time Ridesharing with Service Guarantee on Road Networks. *Proceedings of the VLDB Endowment*, 7: 2017–2028.
- Lesmana, N. S.; Zhang, X.; and Bei, X. 2019. Balancing Efficiency and Fairness in On-Demand Ridesourcing. In *Advances in Neural Information Processing Systems*, 5309–5319.
- Li, M.; Qin, Z.; Jiao, Y.; Yang, Y.; Wang, J.; Wang, C.; Wu, G.; and Ye, J. 2019. Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning. In *The World Wide Web Conference*, 983–994.
- Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2019. ZAC: A Zone Path Construction Approach for Effective Real-time Ridesharing. In *International Conference on Automated Planning and Scheduling*, 528–538.
- Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2021. Zone pAth Construction (ZAC) Based Approaches for Effective Real-Time Ridesharing. *Journal of Artificial Intelligence Research*, 70: 119–167.
- Ma, S.; Zheng, Y.; and Wolfson, O. 2015. Real-Time City-Scale Taxi Ridesharing. In *IEEE Transactions on Knowledge and Data Engineering*, 1782–1795.
- Nanda, V.; Xu, P.; Sankararaman, K. A.; Dickerson, J.; and Srinivasan, A. 2020. Balancing the Tradeoff between Profit and Fairness in Rideshare Platforms during High-Demand Hours. In *AAAI Conference on Artificial Intelligence*, 2210–2217.
- NYC Taxi & Limousine Commission. 2020. TLC TripRecord Data - TLC. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. Accessed: 2022-03-15.
- Raman, N.; Shah, S.; and Dickerson, J. 2021. Data-Driven Methods for Balancing Fairness and Efficiency in Ride-Pooling. In *International Joint Conference on Artificial Intelligence*, 363–369.
- Shah, S.; Lowalekar, M.; and Varakantham, P. 2020. Neural Approximate Dynamic Programming for On-Demand Ride-Pooling. In *AAAI Conference on Artificial Intelligence*, 507–515.
- Sühr, T.; Biega, A. J.; Zehlike, M.; Gummadi, K. P.; and Chakraborty, A. 2019. Two-Sided Fairness for Repeated Matchings in Two-Sided Markets. In *ACM International Conference on Knowledge Discovery & Data Mining*, 3082–3092.
- Tong, Y.; Zeng, Y.; Zhou, Z.; Chen, L.; Ye, J.; and Xu, K. 2018. A Unified Approach to Route Planning for Shared Mobility. *Proceedings of the VLDB Endowment*, 11: 1633–1646.
- Uber. 2018. Uber Matching Solution. <https://marketplace.uber.com/matching>. Accessed: 2022-03-20.
- Wolfson, O.; and Lin, J. 2017. Fairness versus Optimality in Ridesharing. In *IEEE International Conference on Mobile Data Management*, 363–369.
- Xu, Y.; and Xu, P. 2020. Trade the System Efficiency for the Income Equality of Drivers in Rideshare. In *International Joint Conference on Artificial Intelligence*, 4199–4205.
- Yu, X.; and Shen, S. 2019. An Integrated Decomposition and Approximate Dynamic Programming Approach for On-Demand Ride Pooling. In *IEEE Transactions on Intelligent Transportation Systems*, 3811–3820.
- Zheng, L.; Chen, L.; and Ye, J. 2018. Order Dispatch in Price-aware Ridesharing. *Proceedings of the VLDB Endowment*, 11: 853–865.