

Improving Zonal Fairness While Maintaining Efficiency in Rideshare Matching

Ashwin Kumar, Yevgeniy Vorobeychik, William Yeoh ¹

¹ Washington University in St. Louis, St. Louis, MO, USA 63130

Abstract

Order dispatching algorithms, which match passenger requests with vehicles (agents) in ridesharing systems, are able to achieve high service rates (percentage of requests served) using deep reinforcement learning techniques to estimate the relative values of the different combinations of passenger-vehicle matches. While the goal of such algorithms is to maximize the service rate, this may lead to unintended fairness issues (e.g., high disparity between the service rates of geographic zones in a city). To remedy this limitation, researchers have recently proposed deep reinforcement learning based techniques that incorporate fairness components in the value function approximated. However, this approach suffers from the need to retrain should one wish to tune the degree of fairness or optimize for a different fairness function, which can be computationally expensive. Towards this end, we propose a simpler online approach that uses state-of-art deep reinforcement learning techniques and augments their value functions with fairness components during the matching optimization step. As no additional training is needed, this approach can be adapted to use any existing value function approximator and benefits from improved flexibility in evaluating different fairness objectives efficiently. In this paper, we describe several fairness functions that can be used by this approach and evaluate them against existing state-of-the-art deep RL based fairness techniques on standard ridesharing benchmarks. Our experiments show that our fairness functions outperform existing fairness techniques (i.e., it finds matching solutions that result in higher service rates *and* lower service rate disparity across zones), demonstrating the practical promise of this approach.

Keywords

Ridesharing, Fairness, Matching, Transportation

Introduction

On-demand ridesharing has been gaining traction over the past few years as a solution to the growing need for urban mobility. Providing low cost rides to passengers at the expense of small detours, higher earnings for drivers, and a way to reduce the number of vehicles on the streets, it is a solution that benefits everyone. Consequently, there has been work on finding efficient matches of passengers and vehicles to minimize delays and maximize system efficiency for such ecosystems. Recent approaches have used concepts from dynamic programming and deep reinforcement learning to learn policies for matching pools of passenger *requests* to available drivers, achieving real-time performance capabilities. Some approaches, such as that by Alonso-Mora et al. [1], have also looked at this problem through the lens of managing fleets of high-capacity autonomous vehicles. These advances have led to significant improvements in the performance of these algorithms vis-a-vis the *service rate* (i.e., the fraction of passenger

ATT'22: Workshop Agents in Traffic and Transportation, July 25, 2022, Vienna, Austria



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

requests served out of all requests made) as well as the total delay passengers have to experience [2, 3, 4].

Optimizing for such metrics, however, can have unintended effects on the fairness of such systems. The issue of fairness in ridesharing has been discussed in various contexts, ranging from passenger-side fairness in terms of delay and partner choice, driver-side fairness in terms of equitable earning, and even sub-group level fairness [5, 6] for drivers and passengers. Sühr et al. [7] outlines the three stakeholders in the ridesharing ecosystem: (i) the taxis/drivers, (ii) the customers, and (iii) the platform (Uber, Lyft, etc.). The platform takes on the job of mediating between customers and drivers, a problem that is often cast as an Integer-Linear Program (ILP) maximizing overall driver preferences over incoming requests. This central role situates ridesharing platforms to enforce fairness via balancing it with efficiency. Most fairness research in ridesharing aims to quantify this tradeoff in one way or the other as part of the objective function, by intervening in the matching process through a central mechanism.

In our approach, we view the problem in the context of a fleet of autonomous vehicles controlled by the platform. The optimization is formulated as a reinforcement learning problem, drawing from state-of-art techniques [2, 5] to combine the best performing algorithms with fairness approaches. Using zonal fairness as an example of sub-group fairness for passengers, we propose an online approach to incorporate sub-group fairness into ILP-based matching algorithms, without needing to retrain the pre-learned function approximators. Further, we propose designating only a small fraction of the vehicles or requests as “*fairness-aware*” and postulate that this solution is better at trading off efficiency and fairness compared to applying the same modifications to all vehicles. We perform experiments to demonstrate the efficacy of the approach, and perform a grid search over hyperparameters to qualitatively analyze the fairness-efficiency tradeoff. Our experimental results show that our methods outperforms existing techniques (i.e., it finds matching solutions that result in higher service rates *and* lower service rate disparity across zones), demonstrating the practical promise of this approach.

Specifically, our contributions are as follows:

- We develop an online framework that uses off-the-shelf value function approximators and user-defined fairness objectives to trade off efficiency for fairness using state-of-art matching algorithms.
- We introduce new metrics for zonal fairness that consider fairness across source-destination zone pairs.
- We provide real-time tuneable parameters for changing between different types and degrees of fairness, allowing systems to adapt on the fly, and compare the results to existing benchmarks.

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 [8, 9], reducing the pickup and detour delays [1, 10], and increasing drivers’ earnings [11]. 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 [9, 12] or assigning all active requests together in a batch for a small capacity [13, 14]. The sequential solution is faster to compute but the solution quality is typically poor [15]. Alonso-Mora et al. [1] proposes integer optimization approaches for assigning all active requests together for high-capacity ridesharing. Shah et al. [2] and Lowalekar et al. [8] 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 [16], using game-theoretic approaches to fairness [17], and benefit sharing by ensuring non-increasing disutility [18]. Driver-side fairness has also been explored from the economic perspective, by using a max-min approach to fairness to balance efficiency and fairness [11], and by looking at fairness over longer periods of time by equalizing driver income proportional to the number of hours spent on the platform [7]. 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 [5] and drivers [6].

A work that is closest to ours is by Raman et al. [19], 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.

Matching using Reinforcement Learning

A *Ridesharing Matching Problem* is a tuple $RMP = \langle \mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{C} \rangle$ consisting of the set of requests \mathcal{R} , vehicles \mathcal{V} , road network \mathcal{G} , and constraints \mathcal{C} . The solution to this problem is an assignment \mathcal{A} that satisfies the constraints and provides a matching between vehicles and requests.

Specifically, a request $r_i = \langle s_i, g_i, t_i \rangle$ contains the pickup location s_i , dropoff location g_i and the arrival time of the request t_i . The vehicle state $v_i = \langle l_i, p_i, c_i, r_i \rangle$ includes its location l_i , current path p_i , capacity c_i and the requests it is currently serving r_i . 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. Intuitively, it corresponds to the time needed by a taxi to traverse that edge on the graph. The matching algorithm then needs to match requests to vehicles given some time and capacity constraints \mathcal{C} , while optimizing some metric (e.g., maximizing the service rate).

We now describe how NeurADP [2], a state-of-the-art deep reinforcement learning based method, solves this problem. (Our approach is based on NeurADP.) NeurADP learns a *Value Function Approximator* (VFA) for a vehicle's state, which approximates the expected future value of being in that particular state, using a deep neural network. For each vehicle, it generates a set of feasible trips [1] and score them using the VFA. Then, it uses an *Integer Linear Program* (ILP) to maximize the cumulative score over all vehicles.

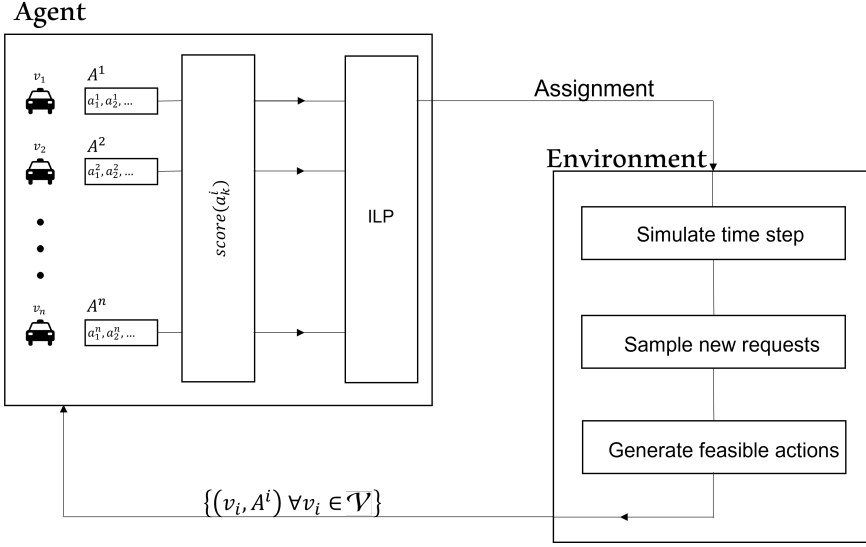


Figure 1: The RMP Pipeline

In typical reinforcement learning fashion, there is an agent and an environment. The environment simulates the motion of vehicles and the arrival of requests with the passage of time. Each decision epoch, the agent takes the current state of vehicles and pending requests as input and solves the RMP, matching requests to vehicles. The agent in our setting involves the combination of a VFA for each vehicle, combined with the ILP that finds the optimal assignment based on the values. Using the VFA in this fashion to predict expected returns allows it to make non-myopic decisions that eventually improve the performance of the system.

The VFA is learned based on TD-learning [20] using experience replay, techniques popularly used in DQNs [21]. We observe transitions from the environment, which are stored in a buffer and later sampled in mini-batches to learn from. All vehicles share the same value function, allowing for efficient reuse of experiences. The training process and network architecture used for the VFA is identical to the one used by Shah et al. [2], and we refer the reader to that paper for more details.

We define an *action* as the matching of a set of requests to a vehicle. If vehicle v_i services request r_j , we denote the reward for the vehicle as $R(v_i, r_j)$. If the set of available (feasible) actions for vehicle v_i is A^i , then, for each action $a_k^i \in A^i$, the score is the corresponding immediate rewards obtained for servicing those requests plus the expected future value of the new vehicle state after being assigned those requests:

$$\text{score}(a_k^i) = V(s') + R(a_k^i) \quad (1)$$

where $V(\cdot)$ is the VFA and s' is the new state of the vehicle v_i after accepting all requests $r_j \in a_k^i$. As a shorthand, overloading some notation, we can write the reward for an action as:

$$R(a_k^i) = \sum_{r_j \in a_k^i} R(v_i, r_j) \quad (2)$$

Let o_k^i be an indicator variable for whether action a_k^i was selected for vehicle v_i . Then, we can write the objective function of the ILP as follows:

$$\max \sum_{i \in |\mathcal{V}|} \sum_{a_k^i \in A^i} o_k^i \times \text{score}(a_k^i) \quad (3)$$

subject to the constraints:

$$\sum_{a_k^i \in A^i} o_k^i = 1, \forall i \in |\mathcal{V}| \quad (4)$$

$$\sum_{i \in |\mathcal{V}|} \sum_{a_k^i \in A^i \text{ s.t. } r_j \in a_k^i} o_k^i \leq 1, \forall r_j \in \mathcal{R} \quad (5)$$

$$o_k^i \in \{0, 1\}, \forall i \quad (6)$$

Intuitively, 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 at a given decision epoch.

The objective function in NeurADP [2] is to maximize the number of requests served. As the reward for each request is uniform and the VFA’s predictions are a discounted estimate of the number of requests the vehicle will pick up in the future, the efficiency can be measured using the *service rate*, which is the fraction of requests assigned out of all requests that were made over a given duration.

Zonal Fairness

As discussed in earlier sections, there has been interest in preserving fairness across sub-groups of the passenger and driver populations. In our approach, we focus on the passengers, looking at service rate fairness across *zone pairs* for pickup and dropoff locations. The motivation to select this metric is two-fold: (i) It is an intuitive metric and is consistent with other fairness objectives that aims for equity in efficiency across sub-groups; and (ii) It is difficult to get information about race, ethnicity, and other cultural factors from publicly-available ridesharing datasets.

Zonal fairness becomes a concern with algorithms like NeurADP [2], which aim to maximize the number of passengers served. In such cases, if most of the demand is concentrated within a certain area, the model will learn to prefer requests to/from that area. Areas with lesser demand are ignored in favor of the overall maximization objective, resulting in the algorithm sending a large proportion of the vehicle population towards regions of high demand. As we discuss in the following sections, small steps towards fairness can go a long way, because serving only a small number of requests in the less-served areas can improve the inequity, possibly at only a marginal cost to the overall efficiency of the system.

We define m zones as disjoint subsets of the locations \mathcal{L} in the city graph. Our sub-groups of interest are people moving between pairs of zones. Recent work [19] aims to maximize the minimum service rate by including variance in service rates across different zones in the

optimization objective. Specifically, they look at the service rate by source zone, which we denote as z_i (i.e., the service rate for all requests originating in zone i).

However, this metric does not take into account the destination of the requests, and there may be a disparity between fairness based on source zones and fairness based on zone-pairs, where some source-destination pairs may be severely under-served. To address this limitation, we define zone-pair service rate z_{ij} as the service rate for requests originating in zone i and ending in zone j . This gives us a grid \mathbf{Z}_p of zone-pair service rates of size $m \times m$, in addition to a vector \mathbf{Z}_s of source-zone service rates of size m .

Given these two sub-group statistics, we can compute fairness metrics of interest, namely the minimum service rate (by source zone or zone-pair) and the Gini coefficient for the service rates (also by source zone or zone-pair). We treat each zone/zone-pair as an individual and fairness is defined by how high the minimum service rate is [11] and by how low the Gini coefficient is. We would like to note that these metrics and definitions of fairness are just examples, guided by recent literature, and we do not claim that these are the only metrics that matter. Instead, fairness is nuanced and subjective, and we aim to show how one can improve the fairness for some particular definitions.

Optimizing Zonal Fairness

The key idea of our approach is that the score function in Equation 1 can be modified to encourage particular request-vehicle matches by adding a bonus to particular vehicles servicing requests that would improve fairness. This bonus can be tuned to suit the need for fairness, and the original score function can be used otherwise. This gives us the flexibility of using the best value functions for general operation to maintain efficiency, and using the bonus to guide the system towards fairer decisions. It is important to note that we do not retrain the value function, only using predictions from an existing one.

Specifically, we define two fairness scores for source-zone (f_s) and zone-pair (f_p) fairness.

$$f_s(r) = \frac{\|\mathbf{Z}_s\|_1}{m} - z_j \quad (7)$$

$$f_p(r) = \frac{\|\mathbf{Z}_p\|_1}{m^2} - z_{jk} \quad (8)$$

Here, r is the request of interest, and j and k are the source and destination zones of the request. Intuitively, each score captures how much worse the request's source zone (zone-pair) is, compared to the average score. A positive score means it is worse. For ease of representation, we also overload this function when used for an action a as:

$$f_s(a) = \sum_{r \in a} f_s(r) \quad (9)$$

$$f_p(a) = \sum_{r \in a} f_p(r) \quad (10)$$

We propose a few methods that use the fairness score in conjunction with the ILP in different ways to improve fairness. Each approach modifies the score function (Eq. 1), and can be used with either f_p or f_s , but we find that f_p is empirically stronger.

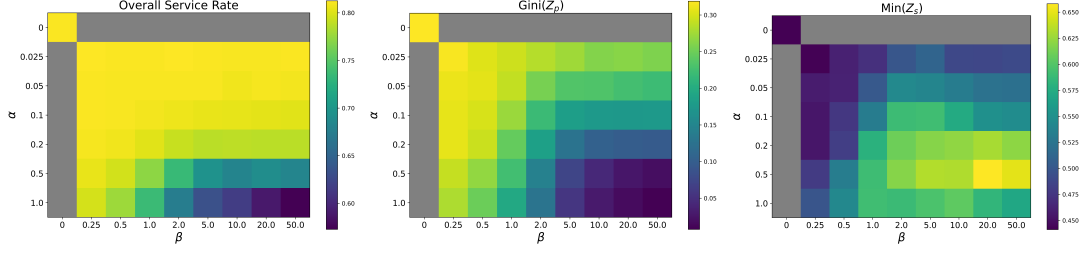


Figure 2: Trends for change in service rate, zone-pair Gini and minimum source-zone service rate with changing α and β for α -veh. The top left ($\alpha = 0, \beta = 0$) indicates the value for unmodified NeurADP.

Fairness-aware Vehicles:

In this method, we designate a random fraction α of the vehicles as “fairness-aware,” and these vehicles receive the fairness bonus. The identity of the fairness-aware vehicles does not change over time. Intuitively, this lets some dedicated vehicles improve fairness while most vehicles continue to improve the efficiency objective. Let \mathcal{V}_α denote the set of fairness-aware vehicles. We then have the following two variants:

- **α -fair vehicles (α -Veh):** The fairness bonus is added to the existing score function for \mathcal{V}_α with weight β . In other words, Equation 1 is replaced with:

$$score(a_k^i) = \begin{cases} V(s') + R(a_k^i) + \beta f.(a_k^i), & \text{if } v_i \in \mathcal{V}_\alpha \\ V(s') + R(a_k^i), & \text{otherwise} \end{cases} \quad (11)$$

- **α -exclusive-fair vehicles ($X\alpha$ -Veh):** \mathcal{V}_α use only the reward and the fairness score, but not the estimated future value from the neural network. Thus, the fairness-aware vehicles are not concerned with maximizing the long-term value, just the fairness.

$$score(a_k^i) = \begin{cases} R(a_k^i) + \beta f.(a_k^i), & \text{if } v_i \in \mathcal{V}_\alpha \\ V(s') + R(a_k^i), & \text{otherwise} \end{cases} \quad (12)$$

Request-based Fairness:

While having designated vehicles trying to enforce fairness is one solution, it is possible that those vehicles aren’t in the right place at the right time and are thus unable to help improve the inequity. To combat this, we can instead look at fairness on a per-request basis, adding the fairness bonus to the rewards of certain requests. Thus, vehicles will be incentivized to pick up requests that have the fairness bonus, even if the value function gives it a lower score. For this approach, we modify the reward $R(a_k^i)$. Let \mathcal{R}_f denote the set of requests with the fairness bonus. We then have the following new reward function to replace Equation 2:

$$R(a_k^i) = \begin{cases} \sum_{r_j \in a_k^i} (R(v_i, r_j) + \beta f.(r_j)), & r_j \in \mathcal{R}_f \\ \sum_{r_j \in a_k^i} R(v_i, r_j), & \text{otherwise} \end{cases} \quad (13)$$

To select which requests get the fairness bonus, we propose two methods:

- **α -subset of requests (α -Req):** Using a pair of hyperparameters (α, β) similar to the two used for fairness-aware vehicles, this approach selects requests by ranking all requests by decreasing $f.(r)$ and selecting the top α fraction of requests.
- **Positive fairness score requests (+Req):** In this approach, after ranking requests (as in α -Req), we select all the requests that have a positive $f.(r)$. It amounts to assigning a bonus to all requests going between zones whose service rate is worse than the average service rate.

We can vary α and β to change the degree of fairness we expect to see in the system. Each method may have a different response to the exact values, but in general, we expect larger values of α and β to improve the fairness objective, while causing a decrease in the overall service rate.

Experimental Setup

To evaluate the efficacy of the different approaches, we evaluate the performance and fairness metrics after running the matching algorithm over a 24-hour period on the island of Manhattan using demand data from the NY Yellow Taxi dataset [22]. The locations in the road network correspond to street intersections, with edges as roads connecting them. We define zones as all intersections falling within neighborhoods on Manhattan island.¹

We use pre-trained NeurADP models [2] as the base value function. We use the method proposed by Raman et al. [19] as a baseline for fairness, using their passenger-side fairness implementation, which we call FairNN. FairNN uses one hyper-parameter (λ) to trade off profit and fairness. In the original paper, experiments were performed for 50 and 200 vehicles with a capacity of 4, for $\lambda \in \{10^8, 10^9, 10^{10}\}$. We use the same parameter settings for our experiments for a fair comparison to the baseline.

However, even with 200 vehicles, the demand from requests saturates the fleet’s capacity. To simulate a more realistic scenario, we run the experiments with 1000 vehicles as well, similar to the work by [2]. For this setting, we also run FairNN with a wider range of λ ($10^3 - 10^{11}$), to more clearly see the trend. Note that each setting of λ requires retraining the value function for FairNN, which is costly.

For the methods discussed in this paper (α -Veh, $X\alpha$ -Veh, α -Req, +Req), we perform a logarithmic grid-search over the hyperparameters α (0-1) and β (0-50). We run this search using both f_p and f_s as the fairness scores, independently. For each setting, we find the overall service rate, the Gini coefficient of the service rate by source zone ($\text{Gini}(\mathbf{Z}_s)$) and by zone-pair ($\text{Gini}(\mathbf{Z}_P)$), and the minimum service rate for any source zone ($\min(\mathbf{Z}_s)$) or zone-pair ($\min(\mathbf{Z}_P)$).

Experimental Results

In this section, we go over the key results from our experiments. The full set of experiments had over 900 runs;² we thus present selected results that illustrate the overall trend. We discuss

¹<https://github.com/erikgregorywebb/nyc-housing/blob/master/Data/nyc-zip-codes.csv>

²The complete results can be found at <https://github.com/ashwinkwashu/ATT22ZonalFairness>

Table 1

Comparison of different methods, selecting the hyperparameter value that maximizes $\min(Z_p)$. The values in bold are the best in their respective columns.

Matching Algorithm	Service Rate	Min(Z_s)	Gini(Z_s)	Min(Z_p)	Gini(Z_p)
NeurADP	0.8119	0.4595	0.0928	0.0370	0.3102
FairNN ($\lambda=10^7$)	0.6855	0.6421	0.0690	0.5355	0.0872
α -Veh ($\alpha=0.5, \beta=20$)	0.6900	0.6580	0.0124	0.5833	0.0259
$X\alpha$ -Veh ($\alpha=0.5, \beta=0.5$)	0.7782	0.7122	0.0227	0.3290	0.1020
α -Req ($\alpha=0.2, \beta=50$)	0.7712	0.7323	0.0184	0.5555	0.0534
+Req ($\beta=15$)	0.7841	0.7313	0.0209	0.5416	0.0544
+Req ($\beta=2$)	0.8146	0.6536	0.0514	0.3414	0.1520

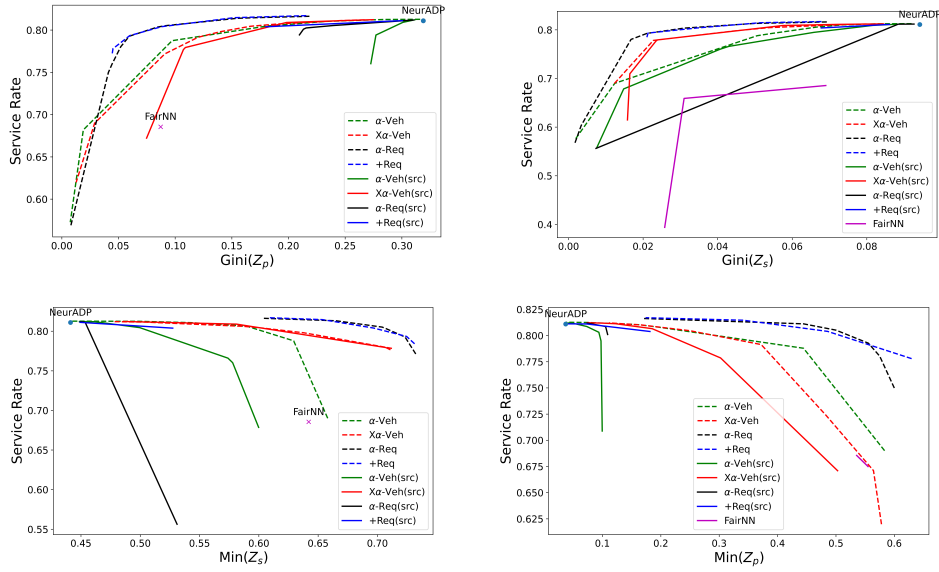


Figure 3: Pareto frontiers for all methods. We use solid lines for runs optimizing source zones fairness, and dotted lines for zone-pair fairness. For Gini, the optimal point is to the top-left (low gini, high service rate). For Min, the optimal point is to the top-right (high min, high service rate).

the results from the analysis of the 1000 vehicle case here, though the trends hold for 200 and 50 vehicles as well.

Pickup-only vs. pickup/dropoff-pair fairness: Figure 3 shows the Pareto frontiers that represent the overall service rate and fairness tradeoff.

Solid lines are associated with our approaches in which groups are defined with respect to *pickup-only* locations (f_s), while dashed lines correspond to approaches that define groups based on *pickup-dropoff* location pairs (f_p).

Our first observation is that in nearly every instance the dashed lines (our approaches using zone-pairs) Pareto dominate solid lines (our approaches using source zones only), *even when we use source zones (Z_p) for evaluation*. This observation demonstrates the usefulness

of the additional observation provided by the latter in improving geographic fairness, and is particularly noteworthy given that the prior FairNN approach defined zones using only pickup locations. Due to this observation, the remaining analysis focuses on pickup-dropoff variants of our approaches.

Baseline comparisons: Figure 3 compares the proposed approaches (consider only the dashed lines henceforth) with two baselines: NeurADP and FairNN. For FairNN, we construct the Pareto frontier by considering many values of λ and omitting any that were Pareto dominated (in some cases, there was only a single Pareto dominant result for FairNN, in which we plotted just this point). First, observe that our approaches Pareto dominate FairNN by a considerable margin. Thus, in addition to computational benefits (we do not have to retrain the value function), the proposed methods can achieve *both* higher service rate *and* greater fairness than FairNN. We also observe that the proposed methods can yield considerable fairness improvement over NeurADP, with relatively minor degradation in service rate. This is further illustrated in Table 1. Consider comparison between NeurADP, FairNN($\lambda = 10^7$), the best parameter value we identified, and +Req($\beta = 15$) in this table. +Req yields nearly the same overall service rate as NeurADP, but with a dramatic increase in the minimum service rate and a dramatic reduction in the Gini coefficient. It also outperforms FairNN on all of these metrics, in several cases by a large margin.

Fairness-efficiency tradeoff: Figure 2 shows how the various metrics change with changing hyperparameters for α -Veh, using the zone-pair fairness score f_p . β controls the weight of the fairness score and α is the fraction of fairness-aware vehicles. We see that as α and β increase (stronger fairness), overall service rate decreases and the fairness objective used in the fairness score (here, $\text{Gini}(\mathbf{Z}_p)$) improves. Other fairness metrics like the $\min(\mathbf{Z}_s)$ also improve, but the change is not monotonic, as can be seen in the third heatmap. This trend is seen across all different methods and both fairness scores. This suggests that there is a direct tradeoff between efficiency and the selected fairness metric, and it is possible to use the other metrics to select a suitable value for the hyperparameters.

Further, this shows that there is merit in applying the fairness score to a subset of the vehicle population, as the change in service rate along the β axis is slower, while still giving gains for fairness. This is also shown by the fact that $\min(\mathbf{Z}_s)$ is maximized when half the vehicles are fair. Specifically, it does not lie at $\alpha = 1$. We also observe that there are diminishing returns with respect to the fairness utility as α increases. As shown in Figure 2, doubling the number of fair vehicles does not double the fairness gain.

Relative performance of our methods: We can observe from Figure 3 that request-based methods (α -Req and +Req) tend to outperform vehicle-based methods (α -Veh and $X\alpha$ -Veh) across the board, with the only exception coming for very low Gini coefficients with respect to pickup-dropoff-based zones (leftmost plot). This is likely because vehicle-based methods require the fairness-aware vehicles to be in the right place in order to have an impact on fairness (e.g., in order for the requests coming from underserved zones to be feasible for these). Request-based methods have comparable performance, but having the additional parameter in α -Req helps it achieve lower Gini coefficients than +Req.

Driver fairness: While fairness in the distribution of driver profits is not our primary focus, it is an important additional consideration in ridesharing. We thus consider how the proposed methods compare to NeurADP and FairNN in terms of the distribution of number of requests

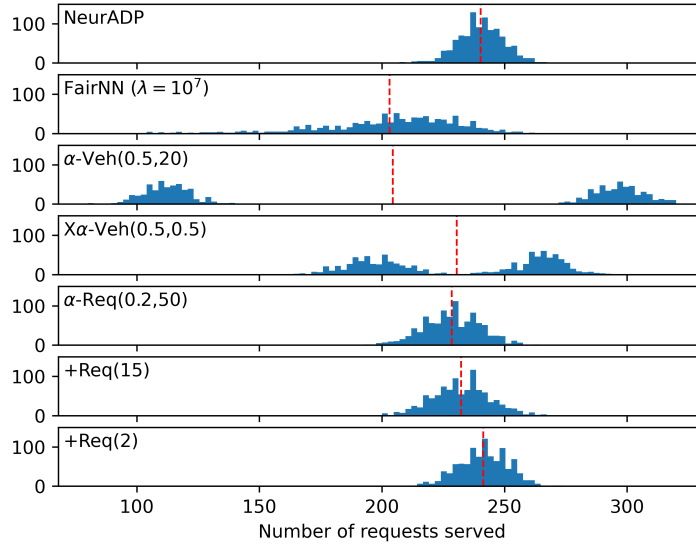


Figure 4: Histograms showing the distribution of trips per driver for each case in Table 1, with the y-axis showing the number of drivers in each bin. The red lines represent the average driver trips.

served (equivalent to income in our setting, as we assume identical costs per trip for all trips) by the drivers for the scenarios in Table 1.

Figure 4 shows the distribution of the trips each driver served in one day. Interestingly, NeurADP exhibits a relatively low variance in the number of requests served, particularly compared to FairNN that uses fairness of *requests* in the objective, perhaps suggesting a significant tradeoff between achieving these two types of equity. What is particularly noteworthy is that our vehicle-based methods yield considerable unfairness in requests served among drivers – not only is variance high, but the distribution itself becomes bimodal. In contrast, our request-based methods exhibit variance comparable to NeurADP, suggesting that it may be possible to achieve high overall service rate, and *both* a high level of rider and driver fairness.

Conclusions

As the demand for cutting edge algorithms for urban mobility increases, their effect on the underlying fairness of these systems needs to be studied, and measures taken to ensure that our algorithms do not inherit the implicit biases that result from pure optimization. In this work, we discussed the issue of zonal fairness in ridesharing systems, introducing new fairness metrics for zone-pair fairness. Through four simple methods that build on existing solutions for order dispatching, we showed how simple techniques can be used to trade off efficiency for fairness. Our methods outperform existing approaches to zonal fairness via an online modification to state-of-art techniques in ridesharing, with the added advantage of not needing any extra training and the ability to be used with any VFA. The tradeoffs proposed in this work can be

dynamically adjusted to have increased fairness at minimal cost to efficiency. Our experiments showed that it is better to apply fairness incentives to a subset of the driver or request population for maximum results, as opposed to a blanket approach to fairness.

Our work leaves some interesting avenues open for future work:

- The generalizability of our approach needs to be evaluated. While it can theoretically be applied with any VFA, experiments are needed to verify that. Similarly, this approach needs to be tested with different sub-group fairness measures apart from zonal fairness.
- Our approach is limited by the need to find the right hyperparameter setting for a desired fairness level, but this problem is shared across other solutions for fairness. A potential solution to this problem may involve learning to adapt the hyperparameters to the current state of the world. Such an approach would automatically be able to tune the fairness based on the changes in demand and supply.
- It is of interest to prove theoretical bounds on the expected gains such an approach can provide, given a particular method to improve fairness. We provide general intuition on why we expect our approaches to work, and to explain the results we see. Future work may find mathematical guarantees that confirm our empirical results.

References

- [1] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, D. Rus, On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment, *Proceedings of the National Academy of Sciences* 114 (2017) 462–467. URL: <http://dx.doi.org/10.1073/pnas.1611675114>. doi:10.1073/pnas.1611675114.
- [2] S. Shah, M. Lowalekar, P. Varakantham, Neural approximate dynamic programming for on-demand ride-pooling, *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020) 507–515. URL: <http://dx.doi.org/10.1609/aaai.v34i01.5388>. doi:10.1609/aaai.v34i01.5388.
- [3] M. Lowalekar, P. Varakantham, P. Jaillet, Zac: A zone path construction approach for effective real-time ridesharing, in: *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, 2019, pp. 528–538. URL: <https://aaai.org/ojs/index.php/ICAPS/article/view/3519>.
- [4] M. Li, Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, J. Ye, Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning, in: *The World Wide Web Conference, WWW '19*, Association for Computing Machinery, New York, NY, USA, 2019, p. 983–994.
- [5] V. Nanda, P. Xu, K. A. Sankararaman, J. Dickerson, A. Srinivasan, Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours, *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020) 2210–2217. URL: <http://dx.doi.org/10.1609/aaai.v34i02.5597>. doi:10.1609/aaai.v34i02.5597.
- [6] Y. Xu, P. Xu, Trade the system efficiency for the income equality of drivers in rideshare, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization*, 2020. URL: <http://dx.doi.org/10.24963/ijcai.2020/580>. doi:10.24963/ijcai.2020/580.

- [7] T. Sühr, A. J. Biega, M. Zehlike, K. P. Gummadi, A. Chakraborty, Two-sided fairness for repeated matchings in two-sided markets, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2019. URL: <http://dx.doi.org/10.1145/3292500.3330793>. doi:10.1145/3292500.3330793.
- [8] M. Lowalekar, P. Varakantham, P. Jaillet, Zone path construction (zac) based approaches for effective real-time ridesharing, arXiv:2009.06051 (2020). URL: <https://arxiv.org/abs/2009.06051>.
- [9] S. Ma, Y. Zheng, O. Wolfson, Real-time city-scale taxi ridesharing, IEEE Transactions on Knowledge and Data Engineering 27 (2015) 1782–1795. URL: <http://dx.doi.org/10.1109/TKDE.2014.2334313>. doi:10.1109/tkde.2014.2334313.
- [10] Y. Huang, F. Bastani, R. Jin, X. S. Wang, Large scale real-time ridesharing with service guarantee on road networks, Proceedings of the VLDB Endowment 7 (2014) 2017–2028. URL: <http://dx.doi.org/10.14778/2733085.2733106>. doi:10.14778/2733085.2733106.
- [11] N. S. Lesmana, X. Zhang, X. Bei, Balancing efficiency and fairness in on-demand ridesourcing, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 5309–5319. URL: <http://papers.nips.cc/paper/8772-balancing-efficiency-and-fairness-in-on-demand-ridesourcing.pdf>.
- [12] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, K. Xu, A unified approach to route planning for shared mobility, Proceedings of the VLDB Endowment 11 (2018) 1633–1646.
- [13] X. Yu, S. Shen, An integrated decomposition and approximate dynamic programming approach for on-demand ride pooling, IEEE Transactions on Intelligent Transportation Systems (2019).
- [14] L. Zheng, L. Chen, J. Ye, Order dispatch in price-aware ridesharing, Proceedings of the VLDB Endowment 11 (2018) 853–865.
- [15] Uber, Uber matching solution, <https://marketplace.uber.com/matching>, 2018.
- [16] O. Wolfson, J. Lin, Fairness versus optimality in ridesharing, in: 2017 18th IEEE International Conference on Mobile Data Management (MDM), IEEE, 2017. URL: <http://dx.doi.org/10.1109/MDM.2017.25>. doi:10.1109/mdm.2017.25.
- [17] L. Foti, J. Lin, O. Wolfson, Optimum versus nash-equilibrium in taxi ridesharing, GeoInformatica (2019). URL: <http://dx.doi.org/10.1007/s10707-019-00379-6>. doi:10.1007/s10707-019-00379-6.
- [18] R. Gopalakrishnan, K. Mukherjee, T. Tulabandhula, The costs and benefits of ridesharing: Sequential individual rationality and sequential fairness, CoRR abs/1607.07306 (2016). URL: <http://arxiv.org/abs/1607.07306>. arXiv:1607.07306.
- [19] N. Raman, S. Shah, J. Dickerson, Data-driven methods for balancing fairness and efficiency in ride-pooling, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, 2021.
- [20] R. S. Sutton, Learning to predict by the methods of temporal differences, Machine learning 3 (1988) 9–44.
- [21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).
- [22] NYC Taxi & Limousine Commission, Tlc triprecord data - tlc, 2020. URL: <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.