Comparative assessment of fairness in on-demand fleet management algorithms

Tarek Chouaki*[1] and Sebastian Hörl[1]

[1]IRT SystemX, France

*Corresponding author

Short summary

On-demand mobility systems in which a fleet of shared vehicles are increasingly tested and deployed. Their efficiency gains are partly due to central algorithms that control the movements and actions of vehicles and drivers. Existing assessments of the performance of such algorithms in large-scale simulation environments assume homogeneous users and vastly ignore special needs of vulnerable users. In this paper, we perform an assessment of two frequently used fleet management algorithms and compare their behaviour when working with heterogeneous customer demand. We show that requests for which higher interaction times at pick-up are anticipated are rejected with higher probability, propose measures to increase the fairness of these algorithms, and propose pathways for future research.

Keywords: fairness, fleet management, on-demand mobility, simulation

1 Introduction

On-demand mobility systems have been an activate subject of research over the past years. In such systems, a fleet of vehicles is managed centrally or in a decentralized way such that a stream of incoming travel requests by customers is served. One field of research focuses on the performance of such systems in realistic agent-based simulations (Jing et al., 2020). These assessments are invaluable because on-demand mobility systems are increasingly emerging in the real world and, for instance, their performance compared to conventional transit services becomes a focus of research (Leffler et al., 2021). Furthermore, while current on-demand mobility systems are managed by drivers, the large potential lies in the automation of such services (Narayanan et al., 2020).

In such environments, fleet management algorithms with distinct objectives, constraints, and other operational requirements can be tested and compared (Hörl et al., 2019). To date, most simulation-based analyses of on-demand mobility systems focus on homogeneous users and homogeneous fleets. This means that specific user groups such as the elderly or mobility-impaired persons are not considered in a particular way. This, however, is necessary to provide human-centered future mobility solutions Gall et al. (2021).

While research on the Heterogeneous Vehicle Routing Problem (HVRP) has been ongoing for many tears (Koç et al., 2016), simulation-based analyses, in which thousands of customers and vehicles are simulated at once, often require heuristic algorithms that are sufficiently correct, but more performant than classic VRPs. Furthermore, the field of HVRP frequently looks at vehicles of varying characteristics and transport *goods*, but rarely looks at individual people with individual needs. Some examples exist, such as Beirigo et al. (2022) who investigate a Dial-a-Ride Problem (DARP) with a business class, Miyaoka et al. (2018) who propose a DARP with the objective to consider a generic measure of inconvenience for the users. Similarly, Aleksandrov (2021) compares different DARP objective formulations that, for instance, explore the minimization of the sum of individual wait times with the minimization of the maximum wait time observed in the system. However, the whole customer demand is known upfront, while this is not the case in simulation-based assessments in which requests arrive dynamically throughout the day.

On a larger scope, these considerations are strongly linked to the emerging field of algorithmic fairness (Mitchell et al., 2021) that aims assessing in how far individuals are disproportionately favoured or discriminated by algorithmic decisions.

In the present work, we perform a first assessment and comparison of two on-demand mobility fleet management algorithms with respect to fairness in a simulation-based environment using an agent-based transport simulation. For comparison, we select the DRT algorithm (Bischoff et al., 2019) that is implemented in the agent-based transport simulation framework MATSim (Horni et al., 2016). Its analysis is relevant as, recently, much research has emerged that uses this platform and its default algorithm to assess the performance of on-demand mobility systems. We compare this algorithm to the one proposed by (Alonso-Mora et al., 2017), one of the most cited and applied fleet management algorithms in place that we denote as the HCRS (High-capacity Ride-sharing) algorithm in the following.

2 Methodology

Simulation environment

To perform our assessments, we make use of the agent- and activity-based transport simulation framework MATSim Horni et al. (2016). While the framework provides the functionality to simulate the daily decision-making of a synthetic population, we opt for a simpler configuration.
In our simulation set-up, we make use of a large-scale MATSim implementation for the Île-de-France region around Paris. From the outputs of that simulation, we extract the car trips of a peri-urban town. The trips, which are realistically distributed with respect to departure time and origin and destination coordinates, establish the demand for our on-demand mobility system. In total, about 8,400 trips have been extracted.
Each trip is assigned whether it belongs to a *vulnerable user* according to a probability P_V which we vary in our experiments. While non-vulnerable users require a duration of 60s to enter a fleet vehicle, vulnerable users need a configurable *interaction time* T_I which is higher than 60s and varied in our experiments. Vulnerable users, hence, take longer to interact with the fleet vehicles and the simulations are configured such that the fleet management algorithms know the interaction time. This corresponds to the case in which an operator may guess the expected interaction time or where it is communicated directly by the user, because, for instance, a request with the need for wheelchair-accessibility has been sent.
The local road network of the city has been extracted as a basis for the simulation. The initial locations of the on-demand vehicle fleet, for which we vary the size, are sampled randomly from all available road links. All vehicles are the same and have a passenger capacity of four persons.
During the MATSim simulation, all vehicle movements are simulated on a second-by-second basis. Customer requests are communicated to the dispatching algorithms at the departure time of each trip. The dispatcher reacts to these requests either immediately or with a certain delay, and may accept and reject requests at any time (including rejects after initial acceptance). The dispatcher, furthermore, controls the movements of the vehicles in the road network at every time step.

Fleet management algorithms

The fleet management algorithms that are tested in this research are MATSim's DRT algorithm (Bischoff et al., 2019) and, HCRS, the one proposed by Alonso-Mora et al. (2017).

DRT: The DRT algorithm is an insertion-based algorithm that directly responds to incoming requests. Within one decision-step (every second) the requests are processed in the order in which they have arrived in the previous time step. For each request, the algorithm will try to insert new pick-up and drop-off activities for the new request into the schedules of the fleet vehicles. Those contain the pick-up and drop-off locations of already assigned requests. An insertion is a combination of a pick-up and a drop-off index along the sequence of existing actions of the vehicle. For each insertion point, it is checked whether the insertion is feasible. This is the case if, by inserting the new action, neither the pick-up time nor the drop-off time of any already assigned request would be shifted beyond a promised threshold.
The threshold for the pick-up time is the latest pick-up time T_p defined as

$$T_p = t_d + \Delta T_w \tag{1}$$

with t_d indicating the desired departure time, ΔT_w the maximum accepted wait time. In our experiments, we fix the maximum wait time for all requests to *ten* minutes.

Furthermore, drop-off times of already assigned requests are not allowed to be shifted beyond the latest arrival time T_a defined as

$$T_a = t_d + \alpha \cdot T_{tt} + \beta \tag{2}$$

with T_{tt} indicating the *direct* travel time between the request's origin and destination in the road network that is scaled by a positive factor α and modulated by the offset β. In our experiments, we fix $\alpha = 1.5$, i.e., we allow a travel time that is 1.5 times longer than the direct trip, plus an offset β of *five* minutes.
If various insertion points across the vehicle fleet are found that fulfil these conditions for all assigned requests and the new request, the candidate is chosen that causes the least additional drive time for the vehicle fleet. If no insertion point is found, the request is rejected.
Note that the DRT algorithm only performs insertions in existing schedules that are extended with every new request. Already assigned requests can not be rejected, and can also not be rescheduled between vehicles or along the stop sequence of the assigned vehicle. Once a request has been accepted, it will be served.

HCRS: The HCRS algorithm applies the same constraint structure as DRT, making sure that requests are picked up before the latest pick-up time T_p and dropped off before the latest arrival time T_a. However, the algorithm is more dynamic than the DRT algorithm.
At every decision step (every 30 seconds), the algorithm reconstructs new vehicle stop sequences from scratch, given all active (not yet picked up) requests. This means that already assigned requests may be assigned to different vehicles, and they may be shifted more flexibly along the schedule of one vehicle. In particular, inserting a new request may cause other requests that have not been picked up yet to be rejected. The problem of which request to assign to which vehicle (or any at all) has been formulated as a Mixed Integer Linear Program (MILP).
The objective of the MILP is to minimize the *total travel delay* δ which corresponds to the difference between the departure time t_d and the expected arrival time, summed over all assigned requests:

$$\min_{\mathcal{A},\mathcal{R}} \quad \sum_{k \in \mathcal{O}} \delta_k + \sum_{k \in \mathcal{A}} \delta_k + \sum_{k \in \mathcal{R}} Q \tag{3}$$

Here, k reference the individual requests and $\mathcal{O}$ the set of requests that are already on board and, hence, can not be rejected any more. The set $\mathcal{A}$ indicates all requests that will be assigned to a vehicle in a particular feasible solution, and $\mathcal{R}$ the set of rejected requests. Each rejected request is considered with a large, constant penalty Q in the objective.
In our experiments, we impose a penalty of *24 hours* for new requests and *1000 times 24 hours* for already assigned requests, thus, avoiding that requests are rejected that have already been assigned before.

3 Results and discussion

In our experiments, the generated requests are served by a vehicle fleet of varying size that is either controlled by either of the two algorithms.
For the fleet sizes, we test values from 25 to 300 vehicles to serve the 8,400 requests. Furthermore, we vary the interaction time of vulnerable users T_I with two and four minutes, respectively, two and four times longer than for non-vulnerable requests. Furthermore, we vary the share of vulnerable users from low values (10%) to 100% in order to test the response of the two algorithms.
While our main interest lies on the rejection rates of vulnerable and non-vulnerable users, we also examine the induced wait times for both groups.

Fleet sizing

Figure 1 shows the rejection rates observed for both user types for DRT (left) and HCRS (right) with a share of vulnerable users that has been fixed to 50%. For all examined fleet sizes, the rejection rate is higher for vulnerable users, because it is more difficult for the algorithms to insert them in the vehicle schedule. This is the even the case above 150 vehicles, which lead to rather relaxed systems states. While almost all requests are accepted for non-vulnerable users at these fleet sizes, rejections can still be observed for vulnerable users. Moreover, we see that the HCRS dispatcher is especially sensitive to the interaction time, rejecting many more vulnerable users when

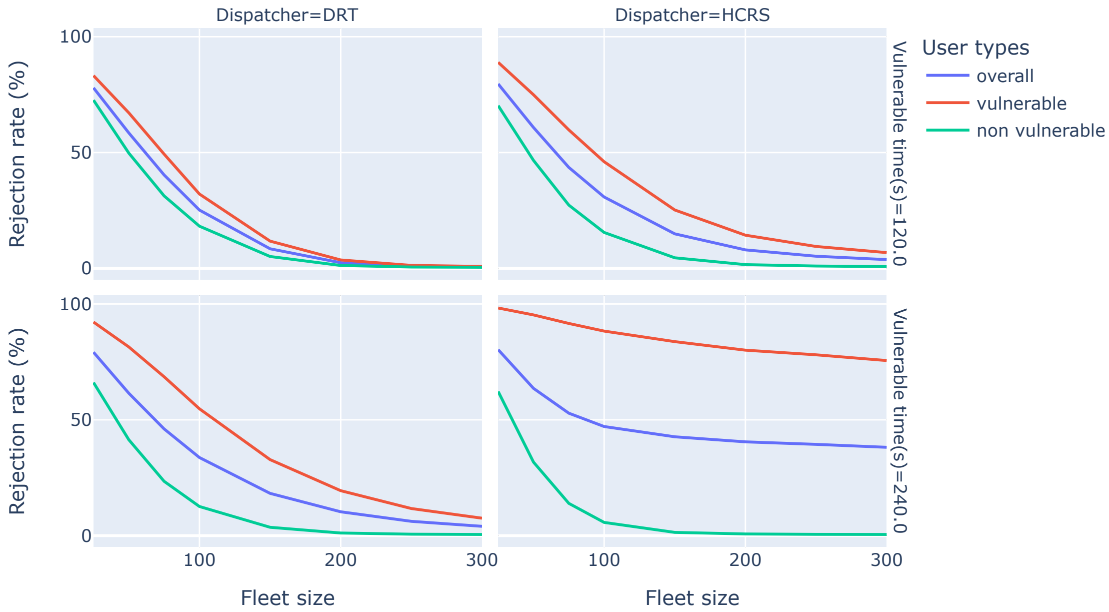

Figure 1: Rejection rates per type of users according to the fleet size

the interaction time is 120s (top) compared to 240s (bottom). This is a shortcoming in terms of fairness of the algorithm, in absolute and relative terms, compared to DRT.

Share of vulnerable users

The fleet sizing analysis allowed to identify 100 vehicles as a good middle ground between very a constrained and rather relaxed service configurations. In the following, we focus our analysis on this fleet size while varying the share of vulnerable users.
Figure 2 focuses on the rejection rates dependent on the share of vulnerable users. At the limit of 100%, the overall rejection rate is equal to the rejection rate for vulnerable users, and it is equal to the one observed by non-vulnerable users in the opposite case. The two dispatchers are shown by column while different interaction times are shown in the rows of Figure 2.
As observed above, vulnerable users have higher rejection rates than non-vulnerable ones in all configurations. As the share of vulnerable users increases, their rejection rate slightly decreases. However, the rejection rate of non-vulnerable users decreases at the same time. This means that the remaining non-vulnerable users benefit disproportionately, instead of distributing the gained performance margin fairly among all users to reach a comparable level of rejections.
Moreover, we can assess the sensitivity of the algorithms towards the passenger interaction time by observing the vertical extent of the resulting Z-shape. Apparently, the identified fairness issues are more severe in HCRS, where the difference between vulnerable and non-vulnerable users is already higher than for DRT at an interaction of time of 120s (top), but increases even further at 240s (bottom).
The phenomenon is further explored in Figure 3, where we propose a first indicator that may quantify the level of fairness of fleet management algorithms. We define the *rejection factor* as the quotient between the rejection of vulnerable users and the rejection rate of the non-vulnerable users. It, hence, indicates, that vulnerable users are rejected n-times more often than the others. This gives a direct comparison of the two algorithms, with HCRS showing a higher rejection factor than DRT.
Figure 4 shows the observed wait times for both algorithms and at the two interaction times. The results are comparable to our analysis of the rejection rates. However, the HCRS algorithm shows a particular behaviour, as the share of vulnerable users increases, the waiting time for non-vulnerable ones decreases. This is exaggerated when the interaction time of vulnerable users is 240 seconds (bottom right), increasing the share of vulnerable users up to 70% allows the non-vulnerable users to lose so much average wait time that it allows for the overall wait time of the service to decrease.

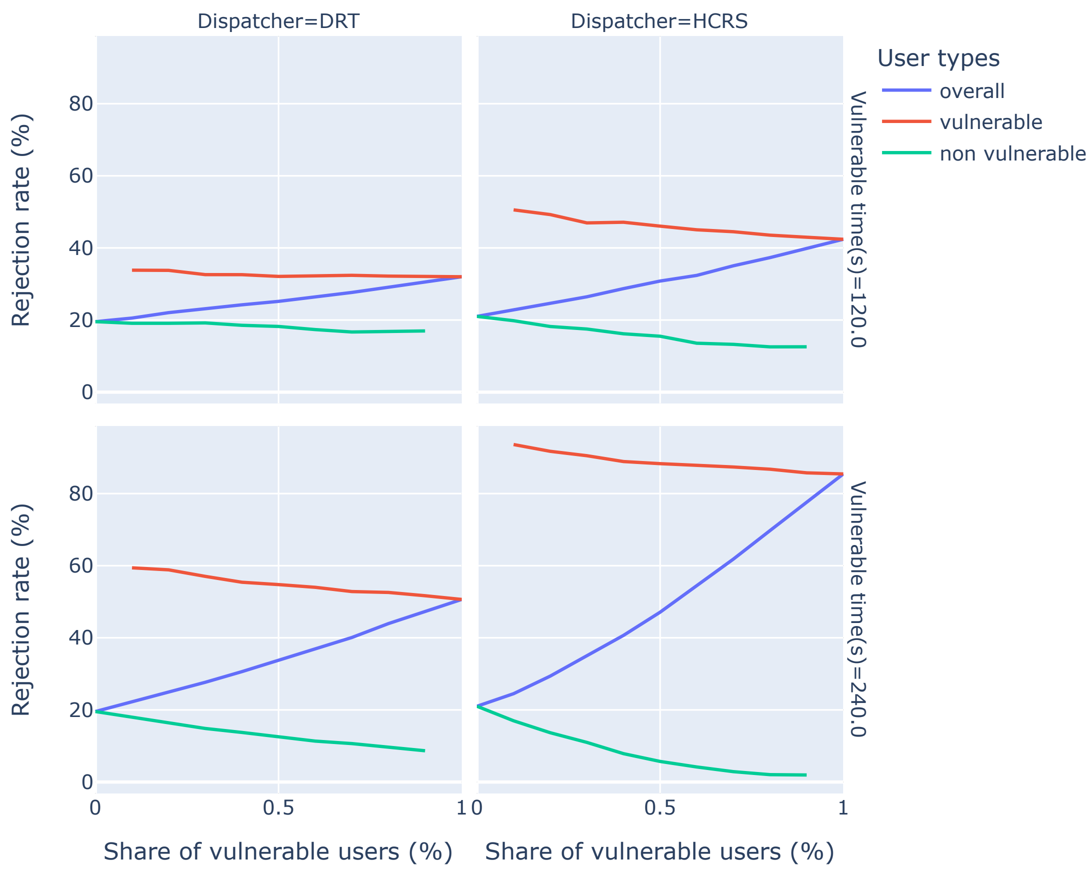

Figure 2: Rejection rates per type of users according to the share of vulnerable users.

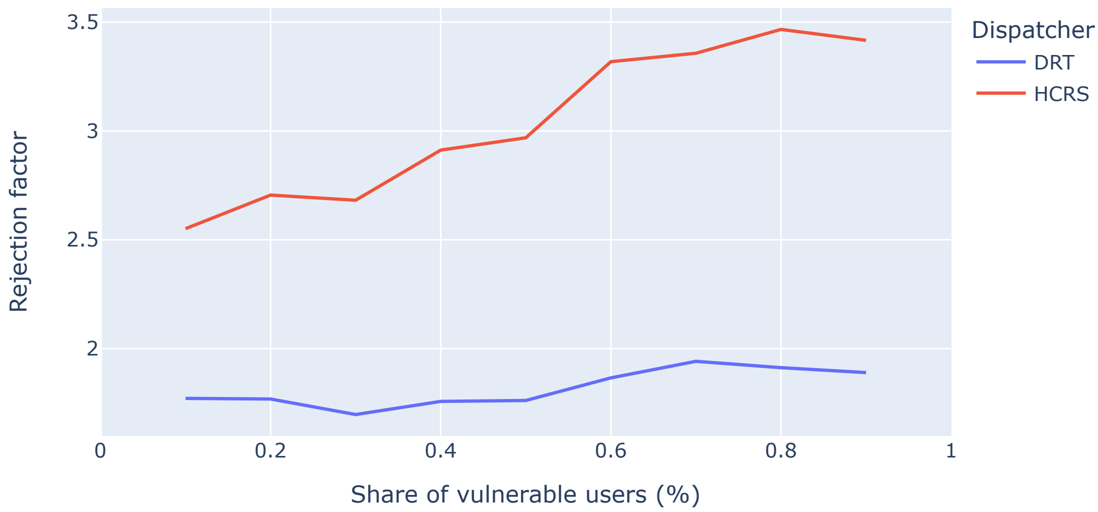

Figure 3: Rejection factor rates per type of users according to the share of vulnerable users.

Figure 4: Average waiting times per type of users according to the share of vulnerable users.

Finally, another main indicator of the performance of a shared on-demand mobility system is the detour factor, which is the ratio between the distance travelled by the passenger in the pooled on-demand vehicle (potentially shared with other travellers) and the shortest distance that they would have travelled if the vehicle was not shared. The observed detour factors are presented in Figure 5. In contrast to the previously presented metrics, the HCRS algorithm is less sensitive to the proportion of vulnerable users and their interaction time. This is in line with its major objective, which is the minimization of overall travel time of the passengers, as outlined above.

Mitigation

Our results show that the algorithms react differently to heterogeneous demand. Based on the anecdotal evidence from our simulation, we can state that the DRT algorithm is more robust against unequal treatment of fleet customers, although effects can clearly be perceived.

When thinking about mitigation measures, we first need to clarify what is the objective of any mitigation procedure: Is it to reduce the rejection rates for vulnerable users to the level of non-vulnerable users? Then a reconfiguration of the offer in terms of fleet size may be necessary, that can be supported by algorithmic changes. If the question is to shift rejection rates of vulnerable and non-vulnerable users to some level that represents a middle-ground, a purely algorithmic change may be feasible. In the following, we present two simple algorithmic mitigation strategies, based on the description of the algorithms further above and which may be tested in future research.

For the DRT algorithm, the objective definition has only minor effects on the dispatching procedure, as requests are inserted or not in exactly the order in which they are passed to the algorithm. However, this fact may lead to a potential mitigation measure: One could introduce a batching mechanism similar to the HCRS algorithm where requests are collected during a certain period (for instance, one minute) and then treated as a batch. Within each batch, one may give a priority value to each request and treat them with decreasing priority. This way, it would be possible to prioritize vulnerable requests or introduce a more complex scoring scheme.

For the HCRS algorithm, the objective function plays a crucial role as requests compete each other in every decision-step. Requests with high insertion times, which are not only more difficult to integrate in existing itineraries, but also cause shifting of assigned requests, are systematically penalized. One option would, therefore, be to adjust the contribution of each request to the objective. Technically, this could mean giving a bonus (negative contribution) per vulnerable request. Note that this bonus may be substantial as it not only needs to compensate for the longer travel delay for the vulnerable request itself, but also for all delay that are caused on others.

In summary, these two examples already show that there may be not one solution that fits all algorithms, but that mitigation measures must be specifically designed for existing fleet management approaches. Furthermore, as outlined above, it must be clearly defined which inequality is tackled by a mitigation measure and how the effect can be quantified.

Note that our analysis only considers immediate requests that expect the operator to pick them within a fixed waiting time right after submission. This creates a paradox in terms of mitigation measures. One's goal may be to provide equal rejection rates for everybody. Following Figure 2 this may mean deteriorating the service level for non-vulnerable users to provide enough of the slack to properly serve vulnerable users. In the limit, we may encounter the situation where there are one or two vulnerable users on some days, and none on others. The system would hence run voluntarily under optimal efficiency most of the time. This paradox may be solved through anticipation. A system in which requests, especially for vulnerable users, can be sent in advance may provide a good trade-off between access to mobility and allocating the required capacity during fleet operation. Prebooking should, therefore, be part of future analyses and the required delays may be investigated.

4 Conclusions

In this paper, we have applied ready-to-use simulation tools to perform an assessment on how heterogeneous customers are processed by on-demand mobility fleet management algorithms. Our analyses show that off-the-shelf algorithms reject requests with high anticipated interaction times with higher probability. This is a potential problem for equal access to future mobility systems. Since on-demand mobility solutions are increasingly tested and deployed world-wide, this problem is of high importance.

Therefore, we recommend pursuing further research on fairness and potential discriminatory effects

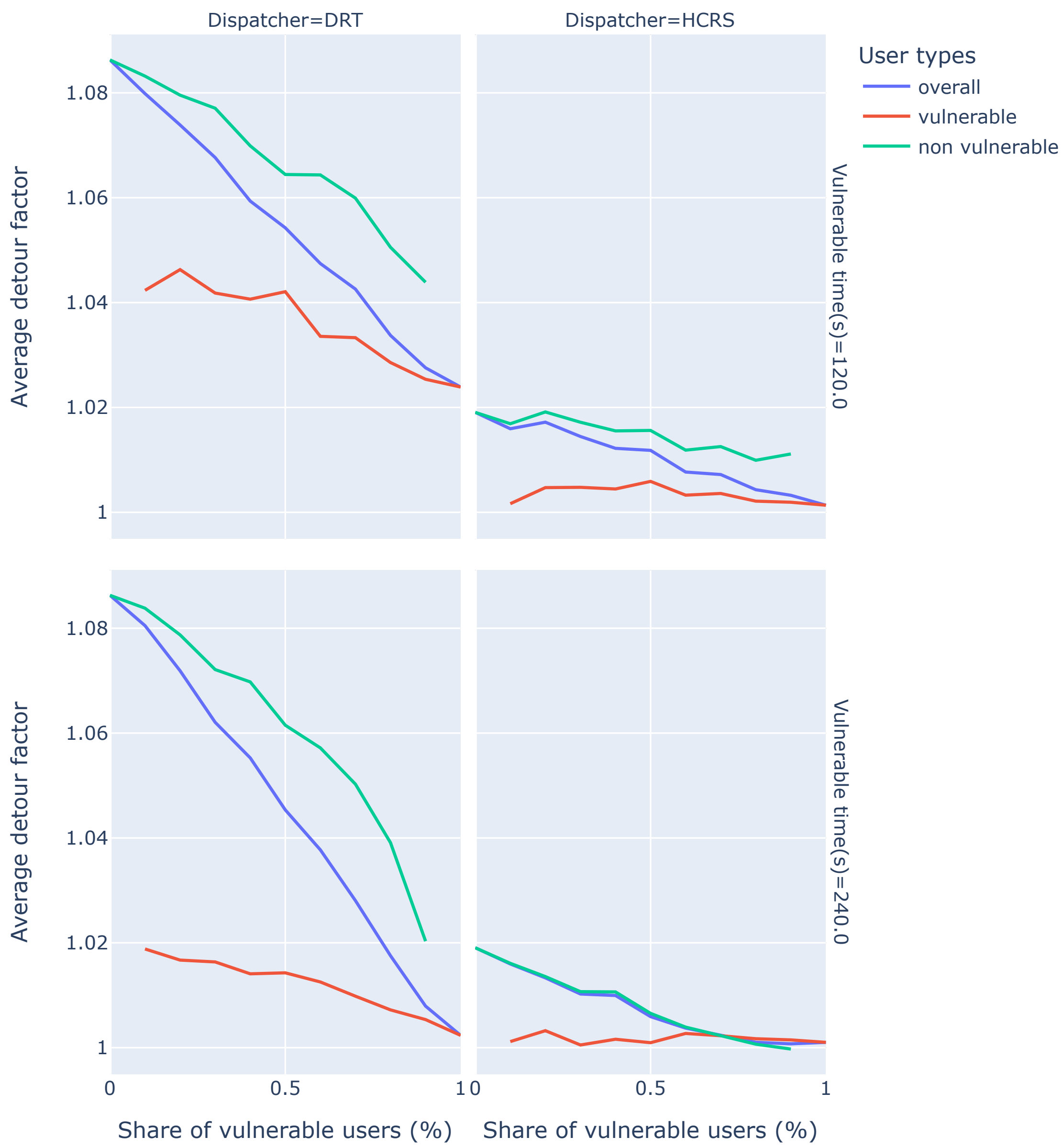

Figure 5: Average detour factors per type of users according to the share of vulnerable users.

of fleet management algorithms. Our goal is to perform a systematic benchmark of existing fleet management algorithms, to quantify their response to heterogeneous demand and propose mitigation measures in future research. For that, it is of utmost importance to develop indicators that are able to quantify unequal treatment and fairness in this context. The *rejection factor* introduced in this paper is a first step in that direction.

Acknowledgements

This paper presents work developed in the scope of the project SINFONICA. The project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101064988. The content of this paper does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in this paper lies entirely with the authors.

References

Aleksandrov, M. D. (2021, July). Fleet Fairness and Fleet Efficiency in Capacitated Pickup and Delivery Problems. In *2021 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1156–1161). Nagoya, Japan: IEEE.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017, January). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, *114*(3), 462–467.

Beirigo, B. A., Negenborn, R. R., Alonso-Mora, J., & Schulte, F. (2022, March). A business class for autonomous mobility-on-demand: Modeling service quality contracts in dynamic ridesharing systems. *Transportation Research Part C: Emerging Technologies*, *136*, 103520.

Bischoff, J., Führer, K., & Maciejewski, M. (2019, January). Impact assessment of autonomous DRT systems. *Transportation Research Procedia*, *41*, 440–446.

Gall, T., Vallet, F., Douzou, S., & Yannou, B. (2021, August). Re-defining the system boundaries of human-centered design. *Proceedings of the Design Society*, *1*, 2521–2530. (Publisher: Cambridge University Press)

Horni, A., Nagel, K., & Axhausen, K. W. (Eds.). (2016). *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press. doi: 10.5334/baw

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., & Axhausen, K. (2019, May). Fleet operational policies for automated mobility: A simulation assessment for Zurich. *Transportation Research Part C: Emerging Technologies*, *102*, 20–31.

Jing, P., Hu, H., Zhan, F., Chen, Y., & Shi, Y. (2020). Agent-Based Simulation of Autonomous Vehicles: A Systematic Literature Review. *IEEE Access*, *8*, 79089–79103.

Koç, , Bektaş, T., Jabali, O., & Laporte, G. (2016, February). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, *249*(1), 1–21.

Leffler, D., Burghout, W., Jenelius, E., & Cats, O. (2021, November). Simulation of fixed versus on-demand station-based feeder operations. *Transportation Research Part C: Emerging Technologies*, *132*, 103401.

Mitchell, S., Potash, E., Barocas, S., D'Amour, A., & Lum, K. (2021, March). Algorithmic Fairness: Choices, Assumptions, and Definitions. *Annual Review of Statistics and Its Application*, *8*(1), 141–163.

Miyaoka, M., Sukegawa, N., & Asano, T. (2018). The dial-a-ride problem with fairness. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *12*(3), JAMDSM0067–JAMDSM0067.

Narayanan, S., Chaniotakis, E., & Antoniou, C. (2020, February). Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, *111*, 255–293.