Ref. Ares(2024)1835863 - 09/03/2024

SINFONICA

Deliverable 2.3

Generic simulation framework for CCAM solutions

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101064988.

Version number:	1.0
Main authors:	Tarek Chouaki – IRTX, Sebastian Hörl - IRTX
Dissemination level:	PU
Internal Reviewers:	TUD, ICCS
Lead contractor:	IRTX
Due date:	29/02/2024
Delivery date:	09/03/2024

Version history			
Version	**Date**	**Main author**	**Summary of changes**
0.1	29/02/2024	Tarek Chouaki, Sebastian Hörl (IRT SystemX)	Initial version
0.2	01/03/2024	Madlen Ringhand (TUD)	Peer review: Minor changes and comments
0.3	04/03/2024	Lazaros Giannakos (ICCS)	Peer review: Minor changes and comments
0.4	06/03/2024	Tarek Chouaki, Sebastian Hörl (IRT SystemX)	Reworking the peer review comments and some improvements
0.5	08/03/2024	Andrew Winder (ERTICO)	Final quality check
1.0	09/03/2024	Giulia Renzi (UNIMORE)	Final submission

Legal disclaimer

Funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

Connected and automated mobility holds great promise for improving the transportation of so-called People with Mobility Challenges (PMC), such as the elderly or mobility-impaired persons. By making specific areas more accessible and extending urban and suburban services in time and space, CCAMs make it possible to offer targeted services to improve social integration.

This report sets out the technical challenges specific to CCAM mobility from the perspective of inclusiveness. The operational performance of on-demand fleet management systems has always focused on optimising the circulation and availability of vehicles to carry out as many journeys as possible and minimising wait times. It seems that another perspective could be adopted to ensure that human factors are considered and prioritised within the mathematical models themselves.

A concise framework for benchmarking and testing CCAM operational around the agent-based transport simulation framework MATSim is elaborated. Furthermore, new extensions that have been developed during the SINFONICA project and which allow researchers to study aspects of fairness and inclusiveness in fleet dispatching are documented.

The results of the analyses produced in this report show that current state-of-the-art fleet management algorithms discriminate against certain categories of users, in particular People with Mobility Challenges (PMC), which includes diverse users who may be disadvantaged or potentially vulnerable due to age, gender, income, disability, ethnicity or a range of other factors. By prioritising operational efficiency to the detriment of users requiring longer interaction times with the vehicles (boarding and alighting), these algorithms reject proportionally more requests from PMC users when they are integrated into a heterogeneous population. These analyses were carried out on two separate algorithms and led to the same conclusions.

Finally, this document proposes concrete approaches and solutions to mitigate the segregation of users when orchestrating journeys. The developments that will be carried out on this basis will make it possible to design service offers that are best suited to the operational contexts of CCAM projects and to all the populations to be transported, regardless of their individual characteristics.

Contents

Tables

Figures

Abbreviations

Abbreviation	Meaning
CCAM	Connected Cooperative and Automated Mobility
PMC	People with Mobility Challenges
DARP	Dial-a-Ride-Problem
DRT	Demand-Responsive Transport
HCRS	High-Capacity Ride Sharing
HVRP	Heterogeneous Vehicle Routing Problem
MATSim	Multi-agent transport simulation
MoD	Mobility-on-demand
VRP	Vehicle Routing Problem

1 Introduction

SINFONICA aims to explore how CCAM services can be designed and deployed in an inclusive and accessible way that provides benefits in mobility to the entire population. The main focus of the project is a co-creational approach in which challenges in terms of inclusivity and accessibility for CCAM, especially for People with Mobility Challenges (PMC), and where recommendations and best practices for deployment will be developed. One line of research in the project aims at informing these considerations by simulation-based experiments in which CCAM services are simulated under different service configurations and in varying environments. These investigations are interesting for both cities/public authorities and transport/CCAM operators. For the former, they would help to make sure that all user groups have equal access to the service. For the latter, they would allow to find suitable trade-offs between inclusivity and overall service performance and cost implications.

The activities around simulation-based assessments in SINFONICA are covered mainly by the two tasks *T2.4: Creation of simulation models for upscaling of measures* and *T3.7: Scaled up impacts on mobility*. The present document represents the final deliverable for task T2.4. We describe how a simulation platform has been set up, which is the technical foundation for the simulation experiments that are to be performed in T3.7 for the specific SINFONICA research sites of Hamburg, Noord-Brabant, West Midlands, and Trikala.

While CCAM simulation toolkits exist to date, they often lack aspects that allow researchers to specifically study the impact of operational decisions on PMC users. Their major assumption is that all users and their requests are uniform and do not differ from each other, for instance in the required time to enter a vehicle. In consequence, fleet management algorithms, which are already used today for non-automatised MoD systems, do not consider questions of fairness and inclusivity. The goal of the simulation-based assessments in SINFONICA, and the technical developments in this report, is to provide a simulation testbed that explicitly allows to assess CCAM and MoD systems with heterogeneous user and vehicle characteristics.

Furthermore, the present report assesses how two frequently used fleet management algorithms operate in the presence of user heterogeneity and identify issues in terms of fairness towards PMCs and explore pathways on mitigating the problem. This way, a solid technical foundation is set for the experiments to be conducted in T3.7.

1.1 Purpose and structure of the document

The present document represents deliverable D2.3 of the SINFONICA project that describes a CCAM fleet simulation toolbox. The work presented here has been implemented in task T2.4. In the following, we introduce both the description of T2.4 and D2.3 and **highlight** the most important aspects that have guided the content of this deliverable.

Task description T2.4: Creation of simulation models for upscaling of measures

An important part of assessing inclusive CCAM solutions is to estimate how such services need to be designed and how specific design decision affect the economic model of operators. Especially, specific user requirements, and thus, dispatching and management of vehicles with special abilities has rarely been studied in the scope of operational and strategic cooperative fleet management.

SINFONICA will **set up a simulation toolkit** for on-demand mobility services given a generic **network** environment, fleet and **vehicle characteristics**, and **request information**. To that end, components from the open-source multi-agent fleet simulations tools **MATSim** (Horni et al., 2016)and **AMoDeus**(Ruch et al., 2018) will be **extended**, customised and thoroughly tested to provide a generic testbed for fleet operational policies.

To allow for the impact analysis of specific needs as proposed in SINFONICA, (1) flexible interfaces to define the **operating area (network, requests)** will be defined and standardised, (2) functionality to define detailed **per user and per request requirements** for the fleet simulations will be implemented, (3) existing algorithmic approaches around the state-of-the-art **High Capacity Dispatching** fleet operational strategy by(Alonso-Mora et al., 2017) will be extended with relevant constraints and solution mechanisms, (4) the developed algorithms will be tested thoroughly based on standard use cases and in comparison to other **baseline algorithms** (e.g., (Hörl and Balac, 2021)). The developed software components will be published as **open source and packaged** in a reusable way.

Deliverable description D2.3: Generic simulation framework for CCAM solutions

A report will present the **added capabilities** to the AMoDeus fleet dispatching framework by providing analyses of standard test cases in literature.

In the following, we give an overview of the individual chapters of this report and how they relate to the task and deliverable description.

Chapter 1 introduces the Task 2.4 of the SINFONICA project and gives a general overview of the task implementation activities.

Chapter 2 provides a thorough analysis of the components that constitute a fleet dispatching benchmarking framework. We draw insights from literature reviews and own experience in working with CCAM dispatching algorithms and detail the functioning of the individual elements of the relevant **simulation toolkits** from passenger-vehicle assignment to charging the vehicle fleet. We furthermore identify which components impact notions of inclusivity and fairness in fleet dispatching. Finally, we compare the identified requirements to benchmark algorithms in terms of fairness with existing capabilities of the **MATSim** simulation platform. We identify necessary **extensions** that needed to be added to the framework.

Chapter 3 takes a closer look at those extensions. We describe how these extensions have been implemented in the MATSim framework. A major component is the introduction of specific **per-request** attributes of individual users and requests. Furthermore, we cover how data can be prepared to run simulations using the framework in terms of **network definition**, **vehicle fleet**

description and **individual user requests**. We finish by describing how the new components have been contributed to the **open-source** community to ensure maintenance and reuse of the task outcomes.

Chapter 4 performs a detailed analysis of how different fleet dispatching decisions impact the service quality for PMC users with the specific example of increased passenger-vehicle interaction times of certain user groups. We identify an issue of systematic discrimination of existing algorithms, most notably in the highly cited **High-Capacity Dispatching** approach by(Alonso-Mora et al., 2017) and quantify the problem. The algorithm is compared to a classic insertion-based dispatching **baseline algorithm** that is frequently used in MATSim-based studies. We, hence, show that the simulation framework can study the impact of operational decisions in fleet dispatching that will be explored in detail in task T3.7 of SINFONICA.

Going beyond the initial scope of the task description, **Chapter 5** proposes first mitigation measures for the specific case of discrimination identified in Chapter 4. We explore individual strategies for the two analysed algorithms an provide general insights.

Finally, we conclude with **Chapter 6** where we summarize the outcomes of this task and how the implemented tools will be used in T3.7 to assess the individual study areas of SINFONICA. Furthermore, as the topic of discriminatory behaviour in fleet dispatching is new, we discuss our simulation results and give recommendations for future research in that field.

1.2 Methodology

Tasks T2.4 and T3.7 make use of simulation-based assessments to estimate the impact of a new CCAM system on a specific territory and under specific operational decisions. These decisions include the size of the fleet, the operational mode (free-floating, stop-based, schedule-based) and service criteria such as the promised maximum wait and travel times. This type of simulation shall be used in T3.7 to assess specific configurations for the SINFONICA research sites of Noord Brabant, Hamburg, Trikala and West Midlands.

In terms of methodology, the present deliverable represents the theoretic foundation to set up such simulations and provides documentation on the technical developments that have been performed to provide the simulation platform itself. The focus of T3.7 will be building simulation scenarios of the SINFONICA research sites and user-friendly visualisation tools to allow non-experts to browse various CCAM settings and investigate their performance and inclusivity implications.

To show the relevance of the approach, we follow a methodology in four steps:

- First, a general assessment of components that make up a CCAM fleet simulation platform are identified. While several of such platforms exist, they usually do not consider special needs for PMC users and often treat all requests equally. We advance the state of the art by identifying how the individual system components impact and are affected by heterogeneous users.

- Second, we assess which components of the agent-based transport simulation platform MATSim need to be extended to allow for such an assessment of inclusivity and general accessibility to a wide range of user request types.
- Third, we implement the necessary components and contribute the developed elements to the open-source community to ensure continuous maintenance. The components are therefore ready to be used in the use-case-specific analyses in T3.7.
- Fourth, we demonstrate the use of the framework for a specific case of discrimination against PMC user groups that is based on different expectations of user interaction times with the vehicles. This assessment has helped to debug and stabilize the simulation platform, while providing valuable scientific insights.

1.3 Intended audience

The intended audience of this report is transport researchers and modellers that are familiar with agent-based transport modelling tools and methodologies. Their interest in our findings is twofold.

On the research level, we provide first insights into how current fleet management algorithms discriminate against specific user groups, even if only small hints of a degraded performance are given to those algorithms. Our findings, hence, provide a technical basis and set a foundation for further research on fairness in fleet dispatching, which is a topic that is barely present in literature.

From the technical perspective, we describe new extensions for the MATSim simulation framework, which is used by many researchers world-wide. Since those extensions have been contributed as open source to the framework, other researchers and modellers will be able to make use of those components and implement the research needed to fill the gaps identified by this report.

1.4 Interrelations

This deliverable is linked D1.1 *Mobility needs and requirements of European citizens* as well as the work performed in T1.4 in the report of which relevant user groups for SINFONICA have been identified.

This present deliverable D2.3 links directly to deliverable D3.4 in which the technical components elaborated here will be used to assess the specific SINFONICA research sites for Hamburg, West Midlands, Noord Brabant and Trikala of the SINFONICA project.

The results of the present deliverable and D3.4 will feed into *D5.5 Guidance and recommendations for demonstrating and implementing user-centric CCAM* to give best practices on the operational aspects that should be fostered and enforced when implementing CCAM services in the long term.

2 Simulation requirements

The simulation-based assessments in SINFONICA make use of agent-based transport simulation to explore how different operational and service design decisions affect the performance and effectiveness of a CCAM transport service. The following sections detail the components that are necessary to perform such a simulation. Furthermore, we explore how an existing agent-based transport simulation framework can be used to perform such assessments and how it has been extended in the framework of SINFONICA. The goal of these extensions is to not only assess performance with respect to a uniform group of users (as is the state of the art), but to consider individual user groups with individual requirements and needs.

2.1 System components

Several simulation frameworks exist that allow the simulation of on-demand mobility services (Jing et al., 2020; Narayanan et al., 2020) in a realistic manner. In general, they ingest the following set of data:

- Usually, a **demand data** set is provided that describes the individual trips with their origin and destination in the study area. Additionally, the frameworks work on departure time data for these requests to know when a person is requesting a ride from the service. Such data can be collected in many ways, very often through representative region-level household travel surveys. In some cases, demand data is not exclusive to CCAM, but describes the overall pool of travels from which trips using a service can be extracted or modeled.
- Second, the **fleet of vehicles** is defined. In most cases, all vehicles of a fleet have the same capacity and start at a configurable depot location.
- Third, the frameworks require a description of the study area where representations reach from simple Manhattan grid-based environments to detailed **network**-based assessments where network data, for instance, from OpenStreetMap, are converted to be used in the simulation.

Such input data can be processed by algorithms from the area of operations research, from the field of Vehicle Routing Problems (VRPs) and, in particular, Dial-a-ride-Problems (DARPs) (Molenbruch et al., 2017; Mourad et al., 2019). These approaches take the ensemble of trips (demand) for one day and aim at finding optimal routes of all vehicles picking up and dropping off one request after another by minimising a global objective such as the total wait time for the requests or the driven distance of the fleet. These approaches require a substantial computational effort and are merely used to find the best possible options to solve a particular demand situation.

In contrast, there are dynamic agent-based transport simulations that rather aim at replicating reality by simulating in detail the movements of travellers and fleet vehicles. By performing a simulation step-by-step in time (usually second by second) the fleet operator also only discovers new requests at the time when they are submitted to the system indicated by the demand data. The operator, hence, needs to react adaptively to the customer demand like an operator in reality would need to. Commonly used frameworks that allow for this kind of analysis are MATSim (Horni et al., 2016), SUMO (Lopez et al., 2018), and POLARIS (Auld et al., 2016).

Technically, most of these simulation frameworks are separated in several individual components that will be described in the following.

Simulation of travellers: The travellers are simulated in a time-step based manner. For each request in the demand data set, the departure time is known. The traveller simulation, hence, needs to make sure that the traveller requests appear during the day at the requested time. They are then sent to the fleet management component (see below) to be treated by the on-demand mobility service.

Simulation of vehicle activities: Furthermore, the activities of all vehicles need to be simulated. In most simulations, vehicles can be described by multiple states including "vehicle is idling at some point in the study area" when it is waiting for new instructions, "vehicle is driving" when it is currently moving from one point in the area to another, and "stopping to let people enter or leave the vehicle", also known as performing passenger pick-ups and drop-offs. The vehicle simulation component specially makes sure that vehicles are moved while driving, meaning that time step by time step the vehicle location is advanced in the network, dependent on the permissible speeds on the roads that are taken by the vehicles. Additionally, the fleet simulation component simulates the interactions between passengers and vehicles. For instance, a vehicle may arrive at a stop location where a passenger is waiting for that vehicle. The fleet simulation component would then move the passenger into the vehicle such that the person will be served by the operator. Likewise, whenever a vehicle stops at a location, the fleet simulation component should make sure that passengers exit the vehicle (drop-off) that have the stop location as their destination.

Fleet management simulation: The *fleet management* component (also known as *dispatching component*) controls the movements of the vehicles. In regular intervals, this component is called to make decisions on the activities of the vehicles. Those activities (or *tasks*) are contained in a schedule for each vehicle, which describes the sequence of actions to be performed by the vehicle. While all vehicles start in "idle" mode, the fleet management component may construct arbitrarily complex schedules that instruct the vehicle to drive to a pick-up location, then drive to pick up another request and after visit the destinations of the respective passengers. To do so, the fleet management component receives the current information on the system described by a list of requests with their current state (waiting or on-board) and a list of all vehicles with their current activity and planned schedules. The aim of the fleet management component is then to call a *dispatching* or *fleet management* algorithm to find an updated schedule configuration that will serve all requests as efficiently as possible. The notion of "efficiency" depends strongly on the fleet management algorithm that is used. Further below, we will introduce two commonly used dispatching algorithms.

While traveller simulation, passenger/request simulation and fleet simulation are the general building blocks of almost all fleet management simulation platforms, there are additional components, dependent on the use case, that are frequently used. Those include the tracking of emissions and energy consumption of the fleet, as well as the technical infrastructure to plan in advance when and where vehicles need to charge. Other questions can be raised with respect to where and when vehicles park in the study area when they don't carry customers. However, for the scope of our experiments in SINFONICA, the latter components are of limited relevance.

2.2 The MATSim framework

MATSim[1] (Horni et al., 2016) is a multi-agent transport simulation framework which has been used for a wide range of transport simulation and forecasting applications. Those range from the analysis of road-pricing to the deployment of automated vehicles, to the measurement of noise emissions and the analysis of mitigation measures for train perturbations. It is an open-source software which is maintained by a large community of researchers and practitioners around the world. This way, it is constantly updated with new functionality, contributed by some of its users.

The main characteristic of MATSim is that it combines simulation on the demand and supply side. On the demand side, the decision-making of travellers is simulated, which are represented by individual entities with their individual plans of what to do during one day and how to move between their activities. The resulting demand in travels is then served in a detailed mobility simulation which is able to represent in detail the movements of private cars, public transport vehicles or on-demand services, among others. Since each vehicle is also modelled as an individual entity, MATSim allows to study and design the heavily dynamic interplay between customers and services in time and space. It is therefore highly adapted to perform analysis and planning tasks for emerging (automated) on-demand mobility services from an operational perspective. On top, it allows for a detailed sociodemographic and economic assessment of such services in a wide range of deployment and adaption scenarios.

Note that the agent-based approach of MATSim provides a high temporal and spatial detail which allows a user to track individual agents (be it vehicles as part of a mobility service or the individual travellers) throughout the day. To realistically model the behaviour and the interactions of highly dynamic mobility services such as CCAM, this is a necessity. Agent-based transport simulations, in general, are used where such complex interplays and detail and time and space are required, contrary to more classic, macroscopic, approaches that focus on large flows of travellers between more aggregated zones and the representation of system-wide car or public transit flows.

The main functionality of MATSim that is relevant for SINFONICA is the simulation of on-demand mobility services in MATSim and the control of the vehicle fleet using a fleet management algorithm. In general, the mobility simulation, in which the on-demand simulation is embedded, is time-step based. This means that the simulation advances by a configurable time-step, usually the simulations are performed on a resolution of one second. For the simulation of on-demand mobility this means that the simulator checks every second in simulation time which agents want to travel with the on-demand service. Accordingly, a request is created, which holds information on the submission time, the origin and destination of the trip in combination with potential additional information such as a user requirement of a maximum wait time or a latest arrival time at the destination. At this point, the traveller agent is removed from the general simulation (which also includes car traffic, public transport, etc.) and is assigned to the on-demand simulator, which decides in the following what

[1] https://matsim.org/

happens to the agent and the associated request. In the worst case, the request and the agent could just be ignored, which means that the agent would be stuck at the point of departure until the end of the simulated day.

Therefore, the on-demand simulator sends the request to the DVRP (Dynamic Vehicle Routing Problem) component (Maciejewski et al., 2017). This is the main implementation of the algorithm that decides how to move the on-demand vehicle fleet. To do so, the dispatcher receives all the requests that have been generated in the current time step and information on all on-demand vehicles that are available in the fleet with their current position and schedule. A vehicle schedule consists of tasks, which can be either “Stay”, “Stop” or “Drive” tasks. “Stay” means that a vehicle is resting at a specific location for a certain time, while at a “Stop” task a vehicle remains for a short period to pick up or drop off passengers. To make the vehicle move and transport passengers, those tasks are connected by “Drive” tasks in which the vehicle follows a certain route in the network from an origin to a destination location. The task of the dispatching component is then to reconstruct all vehicle schedules while taking into account all customer requests.

An example dispatching step is shown in Figure 1. In this scenario, the two vehicles A and B are available in the town. Their schedules are shown on the left and the right side of the figure. Currently, vehicle A is driving towards a “Stop” to pick up the passenger for request R1. Afterwards, it has a drive task to move to the next “Stop” task in which request R1 will be dropped off at the train station. Afterwards, the vehicle will stay there. The current schedule of vehicle B is slightly more complicated: First, the vehicle will pick up request R2, then it will drive to pick up request R3. The first request (R2) will then be dropped off along the route while passenger R3 will be brought further to the train station. Finally, also vehicle B will stay at the train station. While these are the current schedules of the vehicles, they can now be changed by the dispatcher. Specifically, the dispatcher has received one new request RN in the current time step, though, generally, there could be more than one request arriving in each time step. The task of the dispatcher is now to reconstruct the schedules of all vehicles to provide the desired service. This means that RN could be inserted into the schedule of vehicle B, but this seems unlikely.

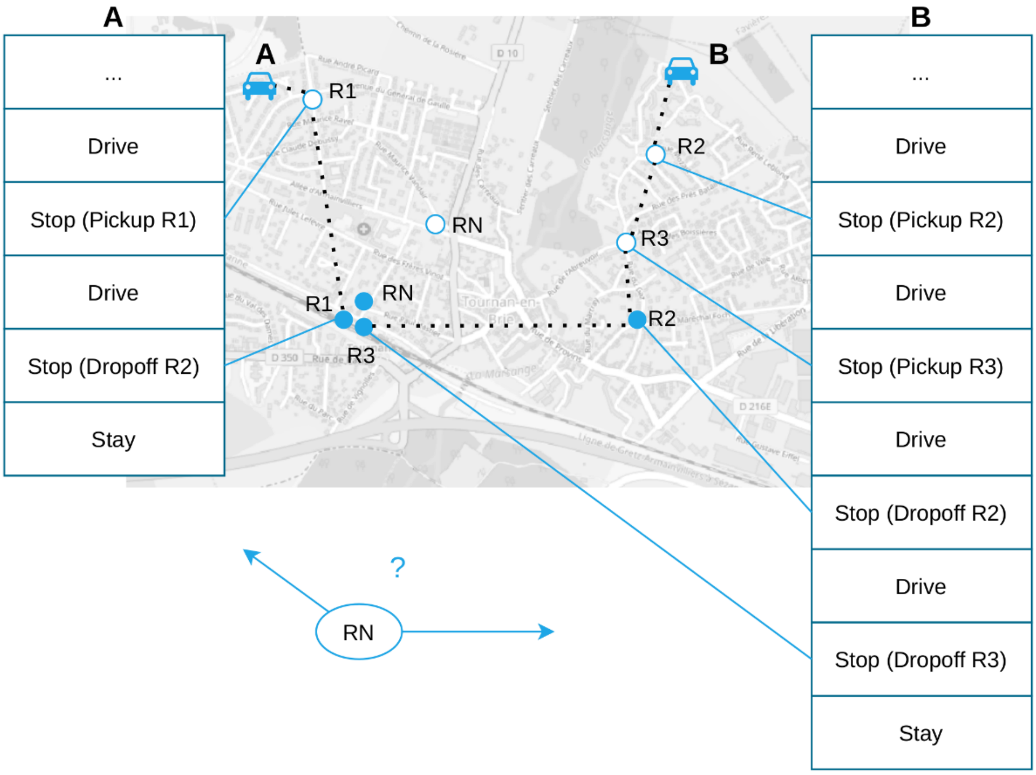

Figure 1: Example for fleet management in MATSim

For vehicle A, the pick-up of request RN could be inserted after the pick-up of R1. However, maybe the vehicle is already late for picking up R1 and to provide a customer-friendly service level, the dispatcher could decide to first bring R1 to the train station and only after pick up RN. In case the customer RN is willing to wait, the dispatcher could also temporarily save RN internally and only consider matching it with a vehicle in the next time step, when, maybe, a more favourable situation arises. Finally, the dispatcher could also decide to directly reject the request, in which case it will be removed from the simulation. The traveller logic then decides if the agent uses a fallback mode, if a new request is sent after some time, or if the agent simply aborts the daily plan. It is important to note that the dispatching component is called in every time step, also if no new requests have arrived. This makes it possible for the dispatching component to react dynamically to congestion and other unforeseen circumstances.

How exactly requests are matched to vehicles and how the vehicle schedules are constructed is the task of the dispatching component and many potential implementations are possible. Currently, a couple of approaches are implemented in different ways. The main point of entry is the DRT (Demand Responsive Transport) extension of MATSim (Bischoff et al., 2019). To ease the development of dispatching components, the AMoDeus (Ruch et al., 2018) framework has been developed which offers a streamlined interface for control engineers. Unfortunately, this additional layer on MATSim has not been actively maintained for a while so that for the SINFONICA project we make use of MATSim directly, which is backed by a large community of researchers.

2.3 Requirements for representing inclusiveness

The components of the MATSim framework have been tested with respect to the requirements that have been developed for the fleet simulations to be performed in SINFONICA:

- Realistic simulation of a CCAM service with vehicle movements and realistic representation of fleet management
- Different deployment modes (free-floating, stop-based, line-based)
- Ability to vary the simulation context (study area) and define operational characteristics (operating area, fleet size, ...)
- Immediate and prebooked service configurations in which requests can be send well in advance, especially for PMC persons
- Individual service requirements per passenger, especially with respect to interaction with the fleet vehicles (interaction time)
- Individual characteristics per vehicle
- Ability to benchmark different fleet management algorithms and ability to modify them

The MATSim framework corresponds to the basic needs of the SINFONICA project in that it provides the functionality to simulate agents (the fleet vehicles) in a realistic way in a detailed road network for a study area. As a drawback, MATSim uses very specific data formats that are not commonly available and used by researchers. Therefore, a need has been identified to simplify the generation of such road networks using data from generic data sources such as OpenStreetMap.

With respect to different deployment modes, MATSim already offers the possibility to configure a service in a free-floating, stop-based or line-based manner. However, the configuration of such a service is embedded in a larger, heavy-weight configuration file of the whole simulation environment. For that purpose, a need has been identified to provide a simplified way of configuring a fleet simulation with a small number of necessary parameters.

The requirement of performing simulation of requests that are either sent immediately to the fleet and well in advance has only been added recently to the MATSim framework. Therefore, a thorough testing and maturation of this functionality needed to be performed in SINFONICA.

In terms of user and vehicle attributes, MATSim, as most existing fleet simulation platforms assumed homogeneous fleets and passenger demands, meaning that individual passengers did not differ in terms of maximum allow wait times, maximum detour requirements and similar constraints. For that reason, the framework needed to be extended to represent a heterogeneous demand data set.

Finally, MATSim had a general lack of detail in terms of simulating the interaction of passenger and vehicles during the stop. In particular, requests were always dropped off as soon as the vehicle arrived a stop and picked up after a fixed time that was uniform for all vehicles and requests. This process needed to be made more flexible to represent different durations of people interaction with vehicles depending on their user profile (average user, mobility-impaired persons, elderly, ...).

The existing functionality of MATSim and the proposed extensions should allow us to consider the mobility needs identified in Deliverable 1.1 *Mobility needs and requirements of European citizens* as well as represent realistically inclusivity challenges of various user groups that were identified in the scope of T1.4. Various of those challenges will be treated in T3.7 where a holistic perspective on the design of CCAM services will be taken, including the additional costs of providing staff, support, and infrastructure mitigating challenges related to *gender, cognitive disabilities* and *digital access* as well as challenges in affordability. Furthermore, varying and specific demand patterns such as for *young* and *migrant* travellers will be covered T3.7 and also questions of system design respecting other participants in traffic such as *cyclists*. The specific user groups and challenges that can be targeted with the present dispatching-related components are the following:

- **Varying times to enter and exit** the vehicle correspond to challenges of *elderly* and people with *physical disabilities*. Issues of higher rejection rates can likely be solved by providing interesting and well-designed pre-booking schemes.
- Travels in the *rural space* are characterised by sparse mobility supply and are expected to be discriminated by dispatching algorithms, especially when operated as a connecting mode of transport to the urban centres. Since our simulations provide a spatially disaggregated representation of the demand, we can observe and mitigate such effects.
- Travellers with *physical disabilities* may need require specific on-board facilities such as wheelchair accessibility, which are represented by providing individual **requirements** to travellers and **skills** to vehicles in our simulations.
- User groups such as single parents which require bookings for two persons when travelling will be considered further in T3.7 with respect to the cost of performing such bookings. Regarding dispatching**, individual attributes** can be attached to their requests which indicate the need for multiple seats, which, in turn, allows us to explore algorithmic discrimination with respect to group sizes and vehicle capacity constraints.

The following of this deliverable is dedicated to detailing our developments with the goal of supporting dispatching-related implications of various user groups (Chapter 3). Later in Chapter 4, we use the developed components to present a first case-study on the behaviour of existing dispatching algorithms under the presence of PMC users and uncover differentiated treatments towards user-groups. We propose and assess mitigation measures for each of the studied algorithms in Chapter 5. We finally conclude in Chapter 6 by summarising the results and outlining the next steps.

3 Framework extensions

The previous Chapter identifies individual components of the MATSim framework that needed to be extended and added to allow for a realistic simulation of PMC users and requests with special requirements. The following sections describe the technical implementation of those components.

3.1 Stop simulation

Initial situation: The fleet simulations in MATSim assumed a fixed duration per *vehicle stop*. Drop-offs (persons exiting the vehicle) were performed at the beginning of this stop duration, while pick-ups (persons entering the vehicle) were performed at the end of the stop duration. In consequence, every agent has the same time of interacting with the vehicle, starting from the time at which the vehicle arrived at the stop location, which are zero seconds for leaving the vehicle and `stopDuration` configurable seconds when entering the vehicle.

Requirement: In our simulations for SINFONICA the goal was to simulate individual interaction times with vehicles, as we expected this metric to be a driving factor in determining whether requests of specific users are discriminated against or not by existing fleet management algorithms. Therefore, the simulation functionality of MATSim and its DVRP extension needed to be extended to individually simulate the entering and existing process of passenger agents.

Implementation: In our implementation, which has been contributed to the open-source MATSim code base, we now allow the framework users to define individual pick-up and drop-off interaction times per request. This means that once a vehicle reaches the stop location, each person on board will need to wait its associated `pickupDuration` until the person actually exits the vehicle, and each person that is waiting at the stop waits `dropoffDuration` until it actually enters the vehicle. Technically, this behaviour is realised by starting a simple queue simulation at the beginning of a vehicle stop: Initially, the times of pick-up and drop-off of each request are calculated based on their individual durations summed up with the stop arrival time of the vehicle. After, in every simulated second, the vehicle logic queries this list to see whether one of the requests/persons is supposed to enter or leave the vehicle. If this is the case, the respective action is triggered.

This modification has implications on the analysis outputs of the simulation as well. Initially, agents waited to enter a vehicle (to be picked up) (`stopDuration + X`) seconds, with X representing the time for the assigned vehicle to arrive. With variable stops durations, the wait time of a request is more flexible dependent on the pick-up interaction time. Likewise, the additional time for exiting the vehicle (previously 0) increases the experienced travel time of the agents as they spent an additional short amount of time “inside” the vehicle. These changes have been considered in updated analysis components for MATSim.

Finally, the changes needed to be considered in the dispatching algorithms themselves. Initially, when finding new ways of inserting a pick-up and drop-off of a traveller into a vehicle’s schedule, only a fixed duration for each stop along the vehicle itinerary needed to be considered. With the addition of flexible interaction times, this logic has been considerably complexified. The corresponding changes have been implemented into MATSim.

Result: Based on our developments it is now possible to assign individual interaction times of travellers with the vehicles to MATSim. This allows us to model in better detail mobility-impaired persons for which an operator may expect additional time to enter or exit the vehicles. In turn, this addition allows us to explore algorithmic discrimination against such requests as will be shown further below.

The standard behaviour of MATSim (fixed pick-up time and zero drop-off time) is a special case in our new flexible component that can be achieved by configuring the parameters accordingly. Hence, we have ensured compatibility with previous simulations performed by us and other MATSim users in the research community.

3.2 Prebooking

Initial situation: The standard implementation of fleet simulations MATSim was based on immediate requests. This means that traveller agents sent requests as soon as they wanted to go on a ride with the service. The framework was, hence, only able to simulate services that are highly responsive to the travel demand. However, our analysis of requirements for inclusive CCAM has yielded the need for also being able to simulate requests that are sent in advance. On the one hand, such services have been identified as a requirement because they will likely be implemented in reality, especially when also rural and low-density areas should be served, for which it will not be able to have such a high service density that vehicles will be sufficiently close to any potential trip origin at any time during the day. Second, such services may mitigate problems of fairness in current fleet dispatching algorithms as we will explain further below in this report.

Implementation: A first draft of the prebooking functionality has been proposed in a previous project (Hörl et al., 2024), while it has been thoroughly tested in SINFONICA. The main idea is that MATSim knows in advance the planned movements of the agents during the simulation. For instance, if an agent intends to depart at 9am, this is initially encoded in the agent plan. While the agent was only submitting a request for the fleet service at that time, i.e. upon departure, we now define additional per-trip attributes that indicate when the request is sent. This information is collected at the beginning of the daily simulation and noted down. Then, a queue-based simulation checks every simulated second whether the requests for certain future departures should be sent. If this is the case, the respective request is sent to the fleet management algorithm.

This basic logic has fundamental impacts on the fleet management algorithms that are currently used in MATSim, so they had to be adapted to the new behaviour. In particular, previously, the assumption held that an agent would already be at the pick-up point when the vehicle would arrive there. This is no longer the case when prebooked requests are taken into account as a request for a departure at 9am may be sent at 8am so the fleet manager may already sent a vehicle in anticipation of the departure. In that case the vehicle would arrive before the agent. The difficulty in the existing implementation was, that a vehicle would then enter “stop” mode and wait for the departing agent. During this time, it was not possible to assign new requests to the vehicle, even there might have been enough of space to perform multiple additional trips serving other passengers. Hence, when scheduling requests into the vehicles’ schedules, new “stay” activities needed to be introduced which allow the fleet manager integrating new activities during that period for following dispatching steps of new incoming requests.

The contribution of SINFONICA in the implementation of the prebooking functionality is that during performing the experiments documented here various edge cases in the compatibility of the presented fleet management algorithms and the prebooking functionality have been discussed. These edge cases, for instance, when and where special prebooked requests can or can not be integrated into the vehicle schedules, have been fixed during the implementation of this deliverable.

Results: MATSim now features the functionality to simulate prebooked on-demand mobility systems. This functionality, together with the existing components, in particular, allows us to simulate all the operational modes of CCAM fleets that have been identified as relevant in our initial analysis. Furthermore, carefully designing who is allowed when and with which delay to prebook future rides provide another degree of freedom when benchmarking CCAM systems in terms of inclusivity and fairness.

3.3 Heterogeneous agents

Initial situation: As in most other fleet simulation systems, travellers and vehicles in MATSim were homogeneous, i.e., all requests and all vehicles were the same. All requests were characterised by a global maximum wait time that is accepted by the travellers and a maximum travel time based on the direct (door-to-door) travel time. Likewise, all vehicles were characterised by an individual number of seats and starting location, but otherwise no different was made between them. Such a setup does it make impossible to study (1) discrimination or specific requests based on their requirements and (2) compatibility questions between travellers and vehicles, for instance, if a mobility-impaired person needs a wheelchair-accessible vehicle.

Implementation: New functionality has been added to the MATSm framework that allows the framework users to define attributes such as the maximum wait time and the maximum detour factor, which were previously defined globally, on a per-person basis. This way, individual preferences and requirements can be represented. The definition of those per-person attributes includes also the pick-up and drop-off times that have been introduced previously. This way, individual agents may indicate that they need longer than the average user to enter a vehicle, for instance.

Similarly, the already existing infrastructure for defining per-vehicle attributes has been further extended. It is now possible to define, by setting per-vehicle attributes, flags such as whether a vehicle is wheelchair accessible or not. To make use of this information, further components have been developed that let fleet management algorithms check whether a request is compatible with a vehicle and its current state variables (such as the current occupancy and other information).

Results: In essence, these components introduce to the fleet simulation components of MATSim the concept of “skills” as it is commonly known in the research around VRPs in the operations research community. In brief, this functionality makes sure that requests that are tagged by the “wheelchair” requirement (for instance) are only allowed to be transported by a vehicle that carries the “wheelchair” skill.

These extensions allow us to describe in much more detail than before the capabilities and requirements of individual travellers and allow us to study effects such as the access to mobility by mobility-impaired persons under different diffusion rates of wheelchair-accessible vehicles in the overall vehicle fleet.

3.4 Data interface

A goal of our technical developments was to simplify the process of setting up a CCAM simulation based on MATSim. For that, we have defined simplified data interfaces that are described in the following.

The basic input formats for a MATSim simulation are the "population file" containing all traveller agents with their daily activities and respective departure times. This file has a relatively complicated structure in XML format, describing all activities of all agents during one day with their respective departure times. In SINFONICA, we are interested only in the fleet management aspect, without setting up full-scale multi-modal simulations of a territory. Hence, the only information that needs to be known are the requests that we assume during one day. These requests are represented by:

- Origin location in coordinates (longitude, latitude)
- Destination location in coordinates (longitude, latitude)
- Departure time

Furthermore, based on the per-agent characteristics described above, each trip has additional attributes such as:

- Maximum wait time
- Maximum detour factor
- Interaction time at pick-up/drop-off
- Other attributes such as whether a wheelchair-accessible vehicle is needed

The main input data ("demand data") is, hence, not a complex hierarchical structure as in the standard inputs of MATSim, but can be represented as a table in CSV format as shown in Table 1.

Table 1: Example of demand input data

request_id	origin_ lon	origin_ lat	destinatio_ lon	departure_ time	max_wait_ time	...
...	...	...	...	...	...	...

Likewise, vehicles are usually described in a XML format in MATSim that can be simplified substantial to a table-based format. In particular, for a vehicle, we require the following information as shown in Table 2:

- Depot location of the vehicle (longitude, latitude)
- Number of seats (capacity)

- Other attributes such as whether the vehicle is wheelchair-accessible

In both cases, we provide the code infrastructure to convert the simplified CSV files into the more complex MATSim format (generating one agent per trip in the case of the "population file") so that a simulation can directly run with the resulting files as input.

Table 2: Example of fleet input data

vehicle_id	depot_lon	depot_lat	number_of_seats
...	...	...	...

Finally, the third input needed for the MATSim simulations in the scope of SINFONICA is the network definition. Again, a network is usually encoded in the XML format for MATSim. A common source of network data is OpenStreetMap where network data is freely available around the world. Hence, we provide the code infrastructure to directly convert OpenStreetMap data that has been downloaded from common data sources to the right MATSim format. The resulting files, again, can be used as a direct input to a MATSim simulation.

Finally, given the demand data, the vehicles data, and the network data, it needs to be defined how the vehicle fleet is operated. We call such a configuration an *operational scenario*. For an operational scenario, the user is allowed to provide inputs in geographical format:

- The operating area in which the CCAM fleet will act: Vehicles will not drive beyond this area and, accordingly, requests outside of this area must be rejected. If no operating area is provided, vehicles will operate on the whole network that is provided by the user.
- A geographic file containing stop locations: If such a file is provided, MATSim will be configured to only let agents enter and exit vehicles at stop locations.

In the configuration file, the user is, additionally, allowed to define the name of a data column that is associated with average stop location in the stop file (if provided) that defines the order in which the stops are visited by the fleet vehicles. This allows to define line-based services that always follow the same trajectory. Finally, the user may define a frequency to also represent services in which the vehicles following the trajectory at fixed intervals, to simulate schedule-based services. An example of such a configuration file in YAML format is shown below:

```
# My service configuration
operating_area_path: /path/tp/operating_area.gpkg
stops_path: /path/to/stops.gpkg
stop_sequence: ~ # No sequence defined
line_frequency: ~ # Not frequency-based
```

The data inputs are documented and updated regularly in the code repository described in the following section.

3.5 Packaging

All the new components and extensions to MATSim that have been developed during T2.4 of the SINFONICA project have already been contributed to the MATSim framework and its open-source code base on Github[2] (GPLv2 license). All code that simplifies the setting up and running of CCAM simulations is also provided on Github[3] (GPLv2 license). While the current state of the code repository represents the technical state of this deliverable, it will be further extended during the developments in T3.7 which will require further customisations and standardisations of the code base. Part of these extensions in T3.7 are specific developments that consider the local context of the research sites Noord Brabant, Hamburg, West Midlands, and Trikala.

[2] https://github.com/matsim-org

[3] https://github.com/tkchouaki/inclusive-ccam

4 Impact on PMC users

In this section, we demonstrate the ability of our framework to investigate the inclusivity implications of dispatching algorithms. We look at how existing algorithms are able to balance PMC and non-PMC users and whether these algorithms show discriminating behaviours in regards to one user-group over the other. While other types of discrimination (such as serving low-density vs. high-density areas) could be analysed with our framework, we focus here on a parametric case study about requests that show an increased interaction time with the fleet vehicles. Such requests can be interpreted as being sent by elderly people that need help to enter the vehicles or mobility-impaired people that required wheelchair accessibility with the respective additional time during vehicle stops for preparing access ramps and ensuring a safe interaction with the vehicle.

The proposed study is highly relevant as, 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.

Large parts of this section were submitted as a conference article to the 12th Symposium of the European Association for Research in Transportation (hEART2024):

> Chouaki, T., Hörl, S. (2024) Comparative assessment of fairness in on-demand fleet management algorithms, extended abstract submitted for presentation at the 12th Symposium of the European Association for Research in Transportation (hEART 2024), Aalto, June 2024.

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 minimisation of the sum of individual wait times with the minimisation 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. This is the case in our more realistic simulation environment.

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.

The following sections describe how our test scenario with the respective study area have been set up, give insights on the algorithms that have been benchmarked, the experimental design that has been followed, as well as the simulation results.

4.1 Study area

For this study, we consider the geographical context of the city of Melun in the Ile-de-France region in France, where Paris is located (see Figure 2). We make use of an open-source pipeline that allows to generate realistic travel demand patterns using open data. This pipeline is applied to the Ile-de-France region with a sampling rate of 10%, resulting in 1.2 million generated travellers (agents) with trips in a 24 hours time-span. This synthetic population is then filtered to keep only the trips that occur (origin and destination) in the study area. This allows to consider not only the trips of residents but also those of external commuters. An optimistic assumption of CCAM adoption is then taken to obtain a set of trips that the service needs to satisfy. This process results in around 8200 trips for which the origins are distributed as shown in Figure 3.

Whereas we illustrate this first study on the inclusivity implications of fleet dispatching algorithms, our functionalities are generic and allow us to transfer the study to the SINFONICA research sites with minimal effort.

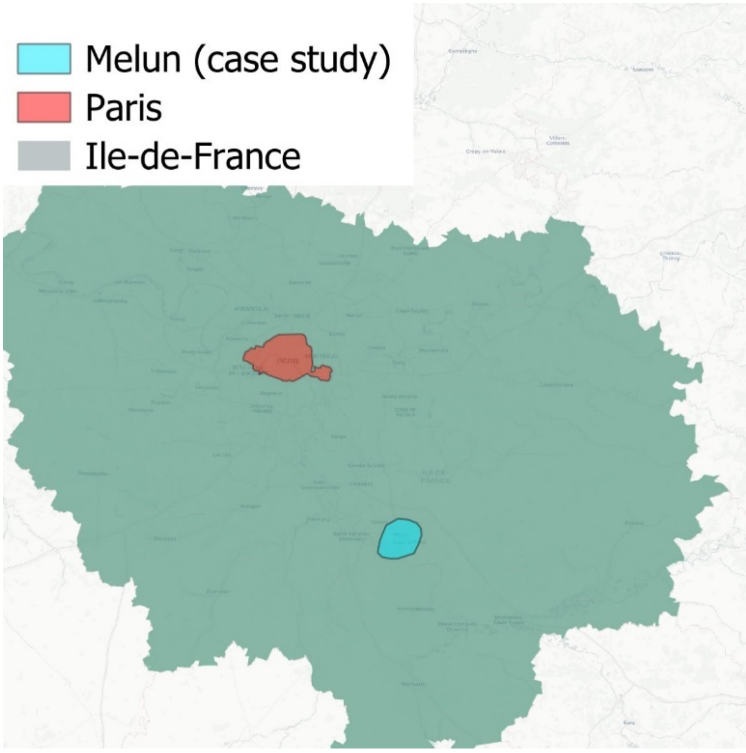

Figure 2: Geographical context of the study area (city of Melun)

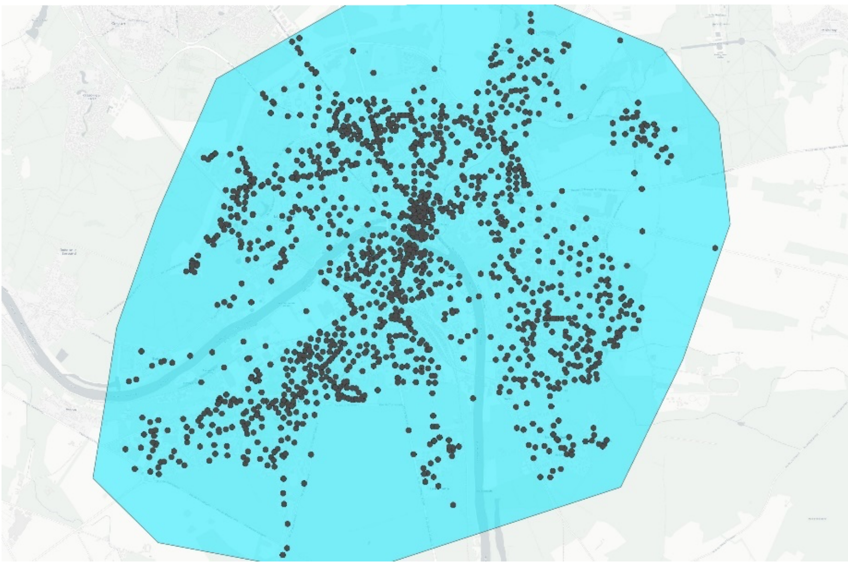

Figure 3: Distribution of the origins of demand trips for CCAM.

4.2 Algorithms

In this section, we present the algorithms that have been evaluated as part of this study. We chose to benchmark two algorithms that are already implemented in the MATSim framework and used in several studies in the literature. This choice benefits the reproducibility of this research by allowing to replicate the approach and extend it further by other researchers. These two algorithms are the MATSim's DRT algorithm (Bischoff et al., 2019) and, HCRS, the one proposed by (Alonso-Mora et al., 2017) which are detailed below. The former algorithm is the standard solution of the MATSim framework and has, therefore, been applied in numerous MATSim-based simulation studies in literature. The latter one is the most cited fleet management algorithm in literature that is used as a common benchmark for new developments. The two algorithms are therefore identified as some of the most influential and, hence, highly relevant for analysis and comparison in terms of fairness.

4.2.1 Demand-Responsive Transport (DRT) Algorithm

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:

$$T_p = t_d + \Delta T_w$$

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 *10* minutes as commonly practiced in the literature around the DRT algorithm.

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

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

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.

Recent developments of the DRT algorithm introduced a new criterion for selecting an insertion point amongst the possible ones. Instead of selecting the insertion that minimises the additional drive time for the vehicle, it is possible to select the insertion that minimises the delay for passengers. In this study, we investigate both variants of this algorithm. The first one is noted **DRT** whereas the second one is noted **DRT*** in the results section.

Note that the DRT algorithm only performs insertions in existing schedules that are extended with every new request. Already assigned requests cannot 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. Moreover, at each decision-making step, pending requests are processed in the order of arrival without considering the ones behind, making the first arriving requests have more priority than the latter ones.

4.2.2 High-Capacity Ride Sharing (HCRS) Algorithm

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, this 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 minimise 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:

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

Equation 1: Objective function of the HCRS dispatching algorithm.

Here, k reference the individual requests and $\mathcal{O}$ the set of requests that are already on board and, hence, cannot be rejected any more. The set $\mathcal{A}$ indicates all requests that will be assigned to a vehicle in a particular feasible solution. Two types of rejected requests are considered, on one hand, $\mathcal{R}$ is the set of previously unassigned requests that are rejected, for which a large constant penalty Q is considered in the objective. On the other hand, $\mathcal{R}'$ denotes the set of requests that are rejected after having been previously assigned. A penalty Q' is considered for these requests.

In our experiments, we impose a penalty $Q = 24\ hours$ for new requests and a penalty $Q' = 1000 \,.\, Q$ for already assigned requests, thus, avoiding that requests are rejected that have already been assigned before.

4.3 Experimental design

In our case study, we use the developed platform in order to evaluate the inclusivity aspects of the dispatch algorithms presented above. In order to do so, we make use of the developed features presented in Chapter 3 (mainly stop simulation and heterogeneous agents) to configure the simulations. The aim here is to explore how the algorithms react to the presence of PMC users that require more time to get in and out of the CCAM vehicles (we call this time PMC time).

A very common analysis performed in the literature is the service's fleet sizing, which consists in investigating various fleet sizes to identify the required number of vehicles to satisfy a certain demand or certain quality criteria. We perform these analyses to identify the operational implications of each algorithm under reference circumstances without PMC users. From the user perspective, we measure the rejection rate, the average wait time and average detour factor. The latter is the ratio between the recorded duration of the trip and the duration it would have had if the ride was unshared (the lower the better for the user). From the operator's perspective, we measure the total distance driven by the fleet, the driven distance during which the vehicles were empty, the ratio between the two distances in order to have a more direct view on the efficiency of the fleet.

We then build simulation scenarios with PMC users by jointly varying the following variables:

- The dispatch algorithm that is used.
- The fleet size, from 100 to 600 vehicles.
- The percentage of PMC users, from 10% to 100%.
- The PMC time, i.e. the time required by PMC users to get in and out of a CCAM vehicle (120 and 240 seconds).

In all the simulated scenarios, non-PMC users require 60 seconds for pick-up and drop-off. For this first assessment, a total of 693 scenarios are simulated and evaluated. We consider that this extensive parametric analysis is sufficient to observe the inclusivity implications of studied algorithms. As for the first part, we measure rejection rates, average wait times and average detour factor for user-perceived service quality. The operator perspective is summarised by the percentage of empty driven distance.

The following section details the obtained results and discusses the impact of introducing PMC users.

4.4 Simulation results

4.4.1 Fleet sizing without PMC users.

Figure 4 shows the impact of the fleet size on the rejection rate observed with each of the evaluated dispatch algorithms in a context with no PMC users. The best performance is reached starting from 200 vehicles for all the algorithms, with HCRS achieving slightly better performances with less than 200 vehicles. However, the average wait time keeps improving after 200 vehicles as shown in Figure 5, the DRT* algorithm consistently shows better wait times than HCRS which in turn performs better than DRT. In general, having more vehicles increases the chances of having a near and idle vehicle nearby when sending the request.

Figure 6 shows the evolution of the detour factor in function of the fleet size. We notice that the best values for the DRT* and HCRS algorithms are reached starting the same threshold fleet size as for the rejection rate (200 vehicles). Interestingly, the detour factors observed with the DRT algorithms do not improve when increasing the fleet size. This is resulting from the objective of the algorithm being to reduce the time during which vehicles are active, resulting in more sharing, hence more detours. This can be also noted when observing the vehicle distances in Figure 7, the DRT algorithm causes the vehicles to travel less distances than the DRT* and HCRS* algorithms. Moreover, for the three algorithms, the driven distances as well as the empty driven distances and their percentage (see Figure 8 and Figure 9) increase until peaking at 300 vehicles before decreasing back.

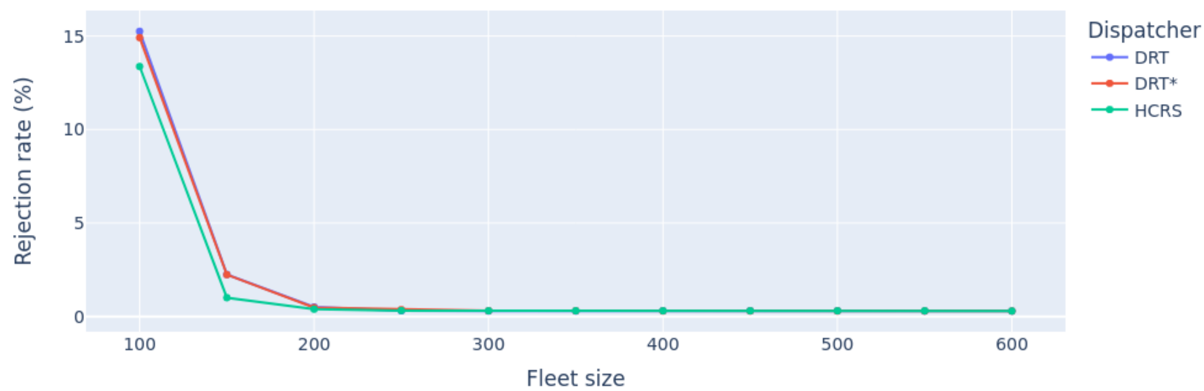

Figure 4: Rejection rates observed with various fleet sizes with each of the studied dispatch algorithms

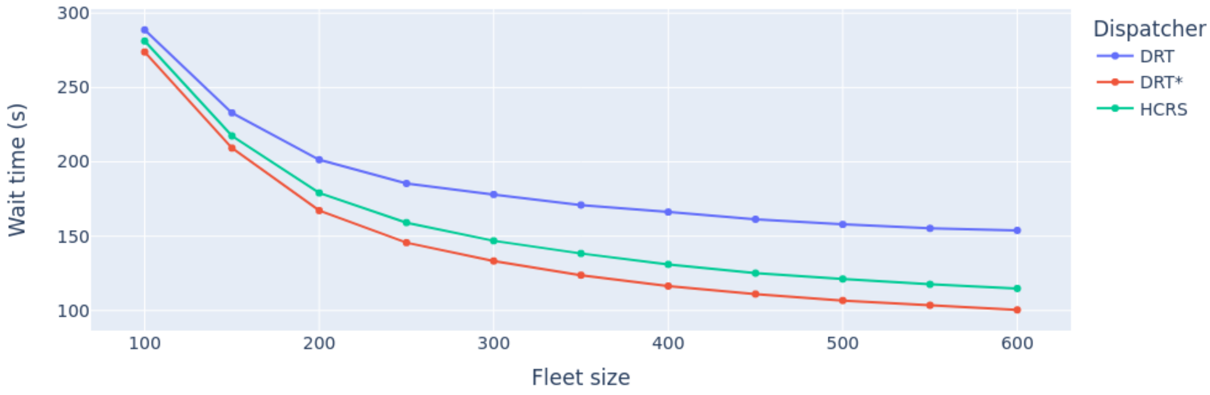

Figure 5: Wait times observed with various fleet sizes with each of the studied dispatch algorithms

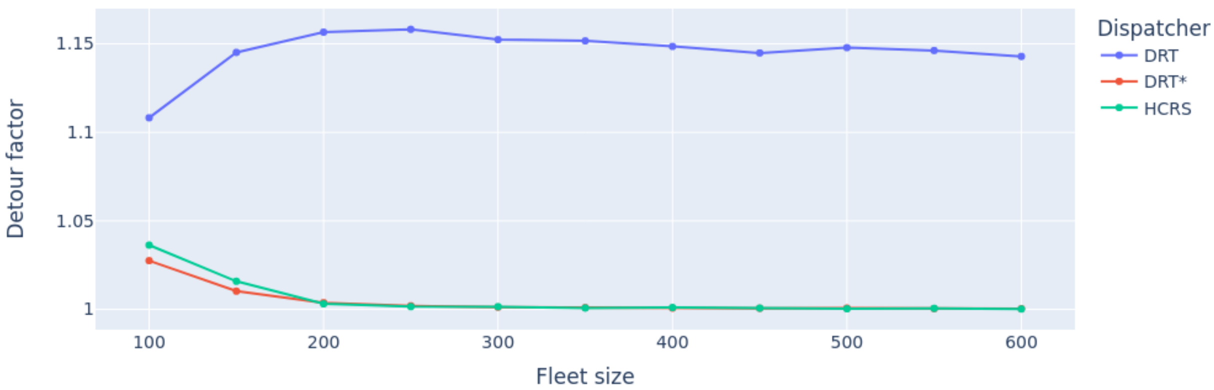

Figure 6: Detour factors observed with various fleet sizes with each of the studied algorithms

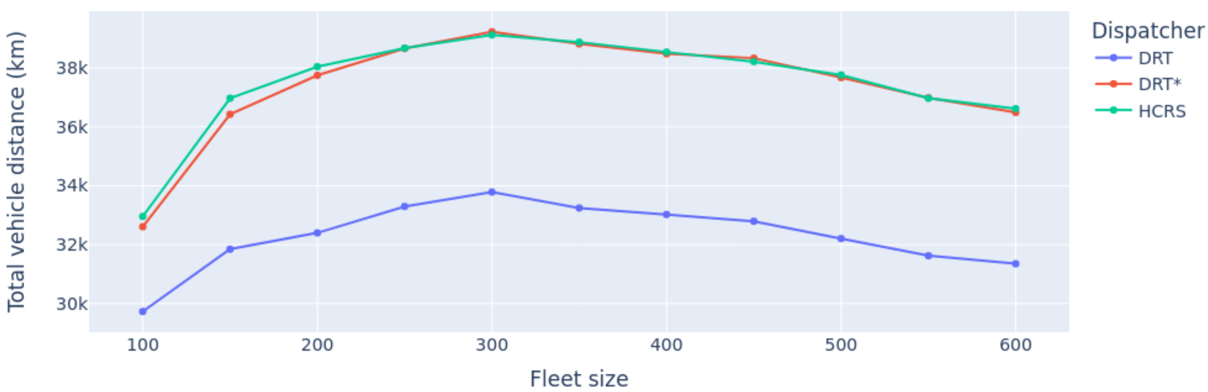

Figure 7: Total vehicle travelled distances with various fleet sizes with each of the studied algorithms

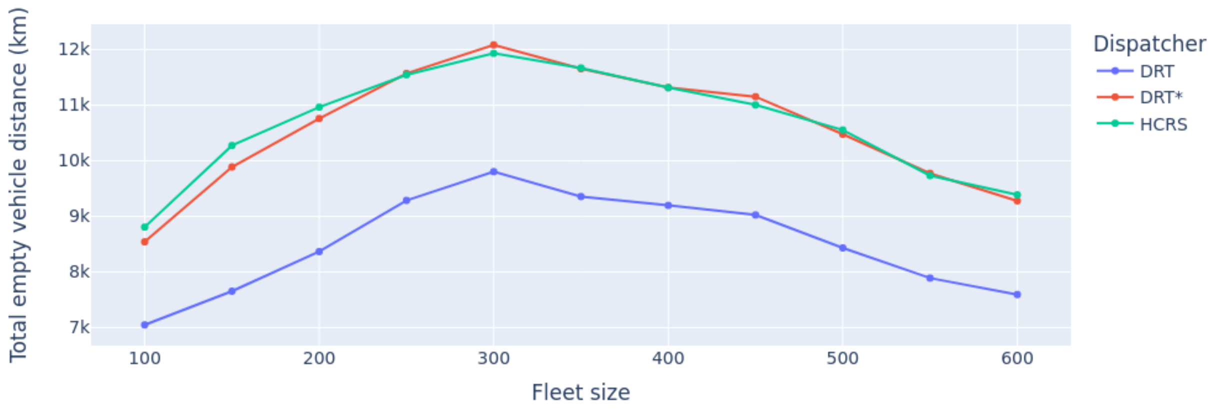

Figure 8 Total empty vehicle travelled distances with various fleet sizes with each of the studied algorithms

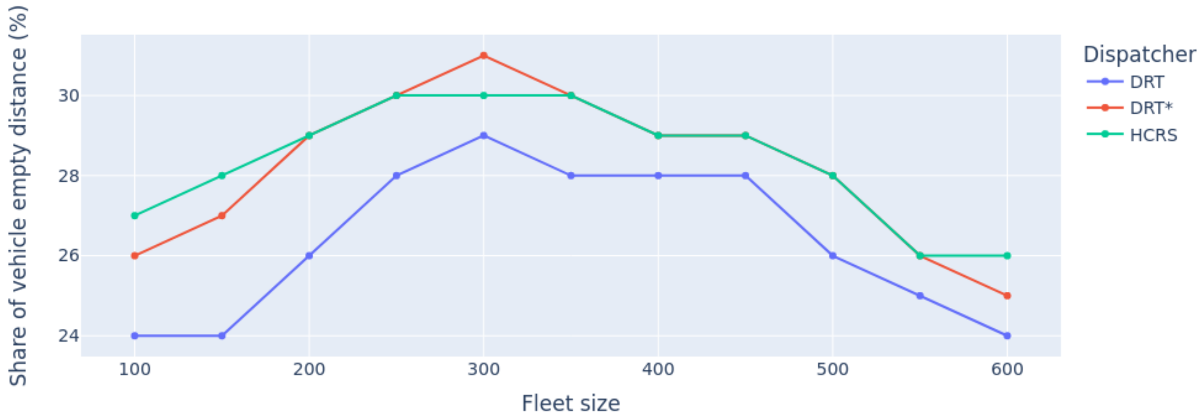

Figure 9: Shares of total empty vehicle distances with various fleet sizes with each of the studied algorithms

4.4.2 Introducing PMC users on a fixed fleet size

We present here the results obtained from simulations with a 100 vehicles fleet and varying the share of PMC users between 10% and 100%. The goal here is to study the behaviour of the system under an already constrained fleet size, hence 100 vehicles. This analysis is performed for the three algorithms with the two considered values of PMC times. For each measure that is taken, we consider two visualisations, one comparing between user types with respective curves on the same plot for separate algorithms in rows and another one with algorithm-related curves on the same plot for separate user-type in rows, in both visualisations, the results for both PMC times (120s and 240s) are shown in columns.

As shown in Figure 10, PMC users have higher rejection rates than non-PMC ones in all configurations. As the share of PMC users increases, their rejection rate slightly decreases. However, the rejection rate of non-PMC users decreases at the same time. This means that the remaining non-PMC 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. We can see that increasing the PMC time from 120s to 240s greatly impacts the fairness between PMC and non-PMC users for all algorithms. Figure 11 compares the rejection rates for each user type between the algorithms. The three algorithms produce roughly similar rejection rates with HCRS performing slightly better.

Figure 12 shows the observed wait times for the three algorithms and at the two interaction times. The results are comparable to our analysis of the rejection rates. In all settings, PMC users wait more on average for a CCAM vehicle than non-PMC users. Moreover, the disparities between user types do not decrease by increasing the share of PMC users. As shown in Figure 13, the relative wait time performance of the algorithms is not altered by the introduction of PMC users as DRT* still performs best as in Figure 5.

Detour factors are the only aspect on which PMC users are at least as well-served as non-PMC ones as shown in Figure 14, the DRT algorithm shows the largest differences in detour factor in the advantage of PMC users. For the DRT* and HCRS algorithms, the detour factors are only slightly better for PMC users with an interaction time of 120s and even more so with 240s. The better detour factors experienced by PMC users are a direct consequence of the longer time required by PMC users to interact with the CCAM vehicle. Due to all users having the same wait time and travel time constraints and the pick-ups and drop-offs being counted as part of the travel, it is harder to add more passengers to a vehicle with a PMC passenger without breaking the constraints. This reduces the level of sharing and detours experienced by PMC users. Comparing between algorithms in Figure 15, detour factors between algorithms show the same relative quality as what is observed in the reference simulation (Figure 6).

Figure 16 shows the shares of empty driven distances according the share of PMC users compared between the algorithms. Introducing PMC users does not alter the relative performance of the algorithms, with DRT always performing better than DRT*, which is in turn better than HCRS in this regard. The impact of the PMC time is focused upon in Figure 17, with a PMC time of 120s, the share of empty driven distances is not impacted by the share of PMC users. This contrasts with the setting with 240s of PMC time where the share of empty driven distances decreases as the share of PMC users increases, while maintaining the relative performances between the algorithms.

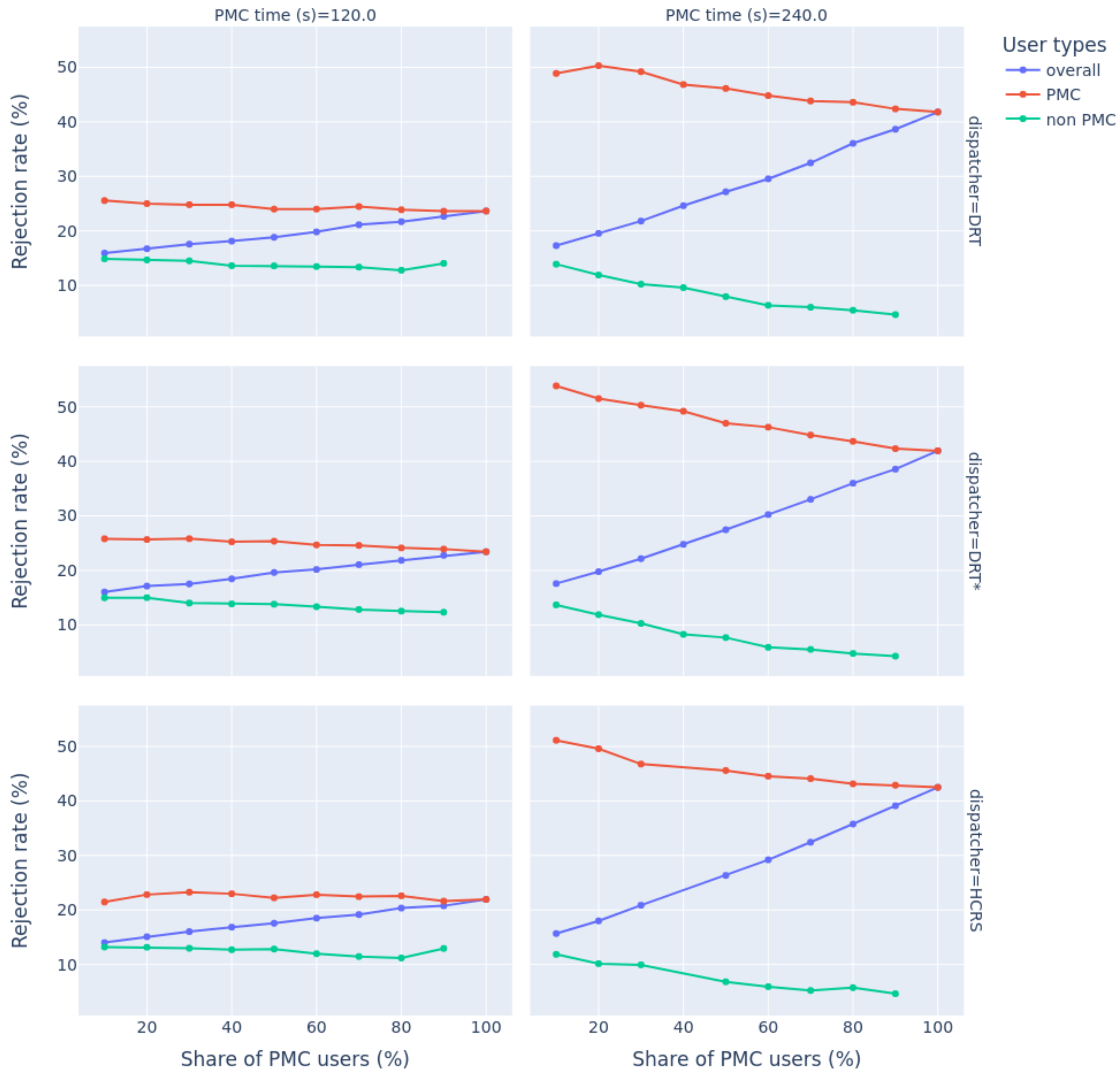

Figure 10: Observed rejection rates under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across user types.

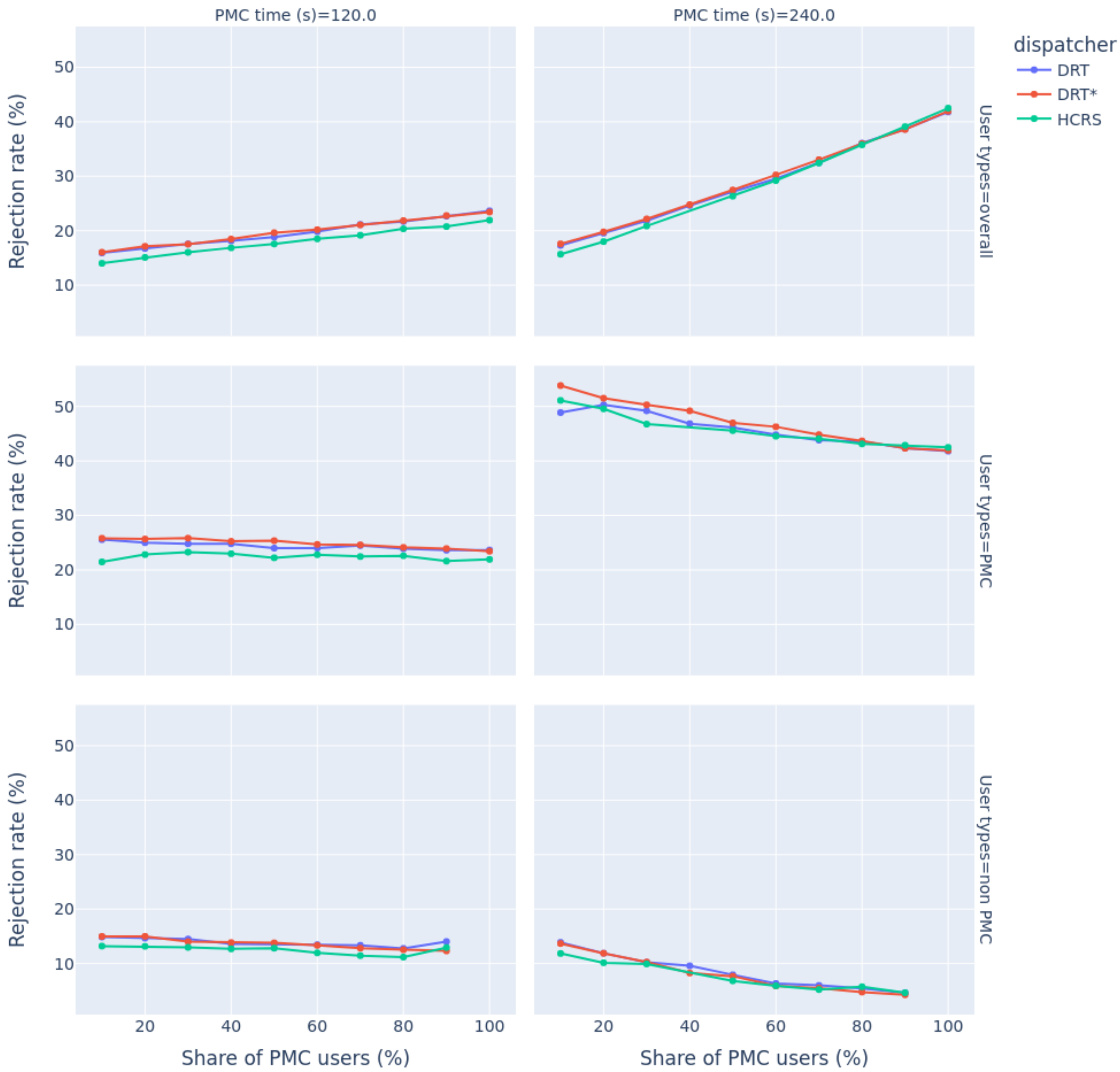

Figure 11: Observed rejection rates under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across algorithms.

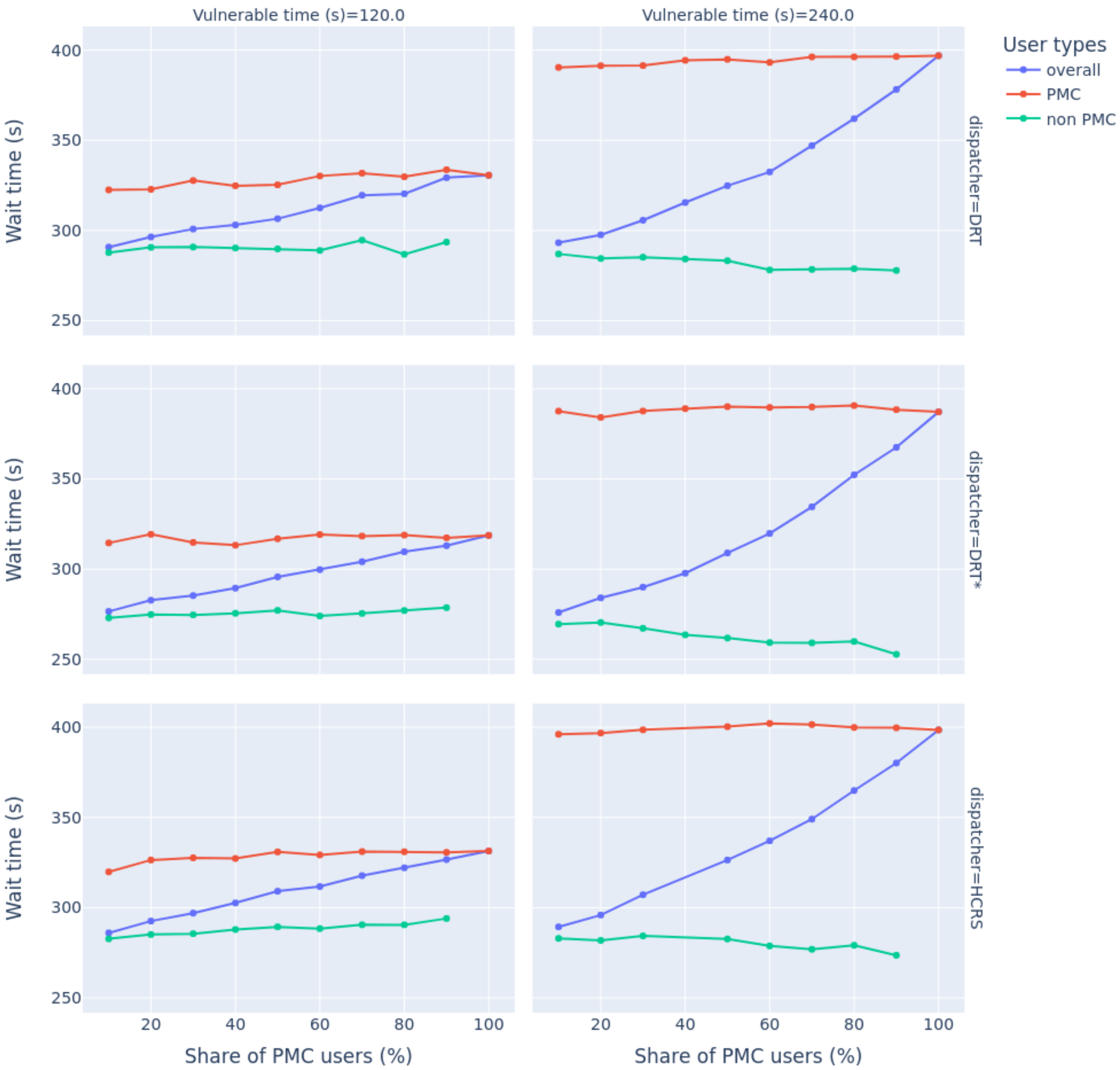

Figure 12: Observed wait times under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across user types.

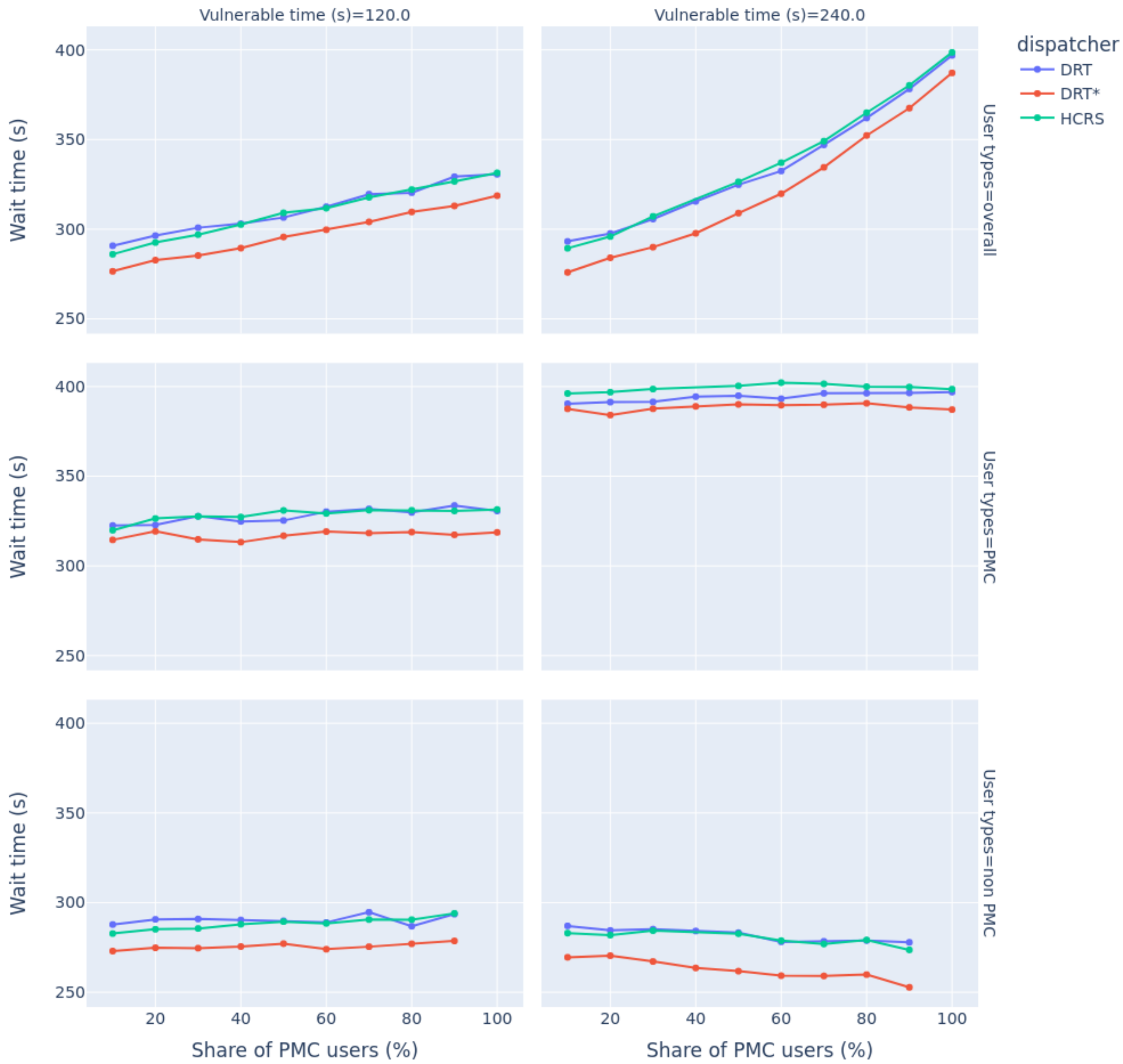

Figure 13: Observed wait times under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across algorithms.

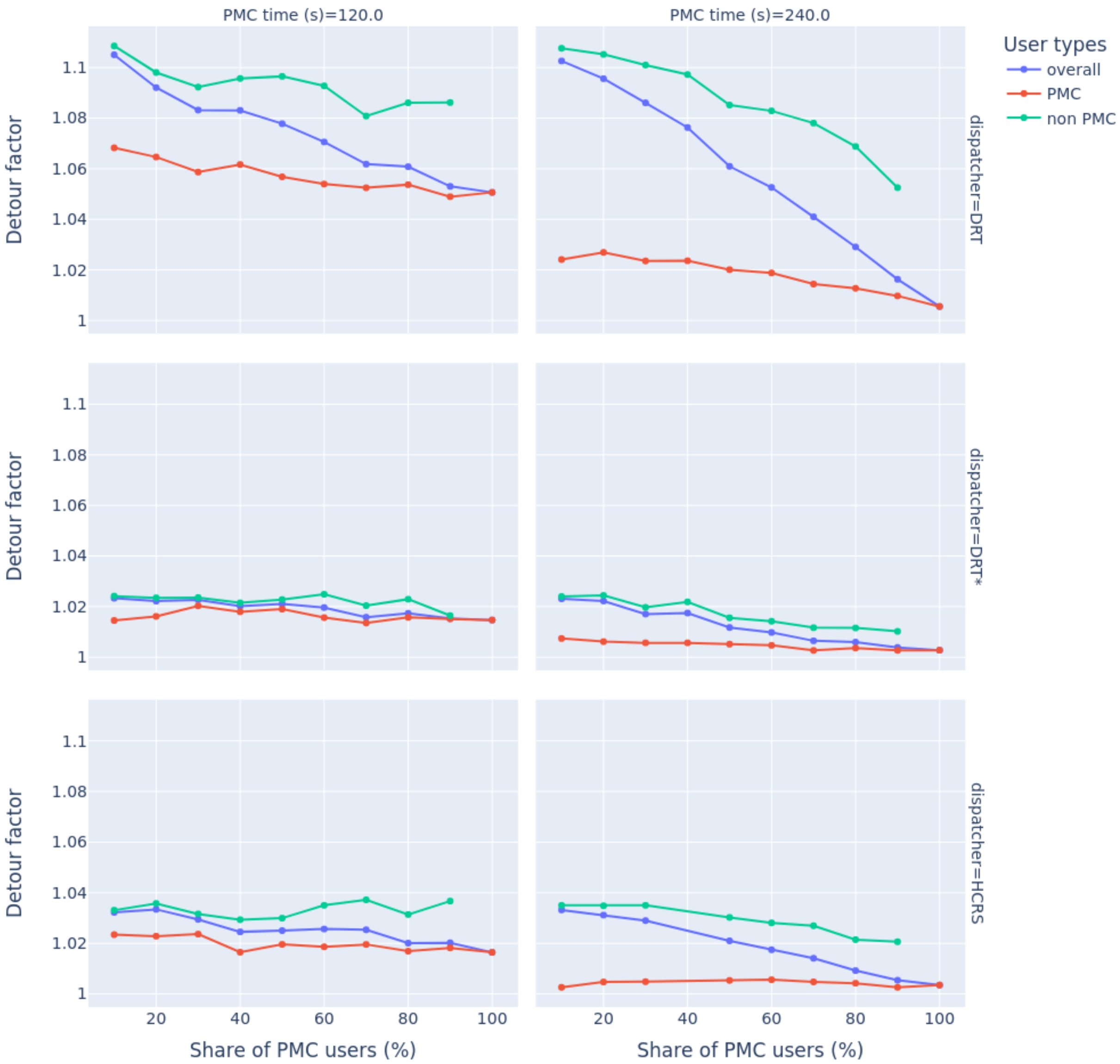

Figure 14: Observed detour factors under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across user types.

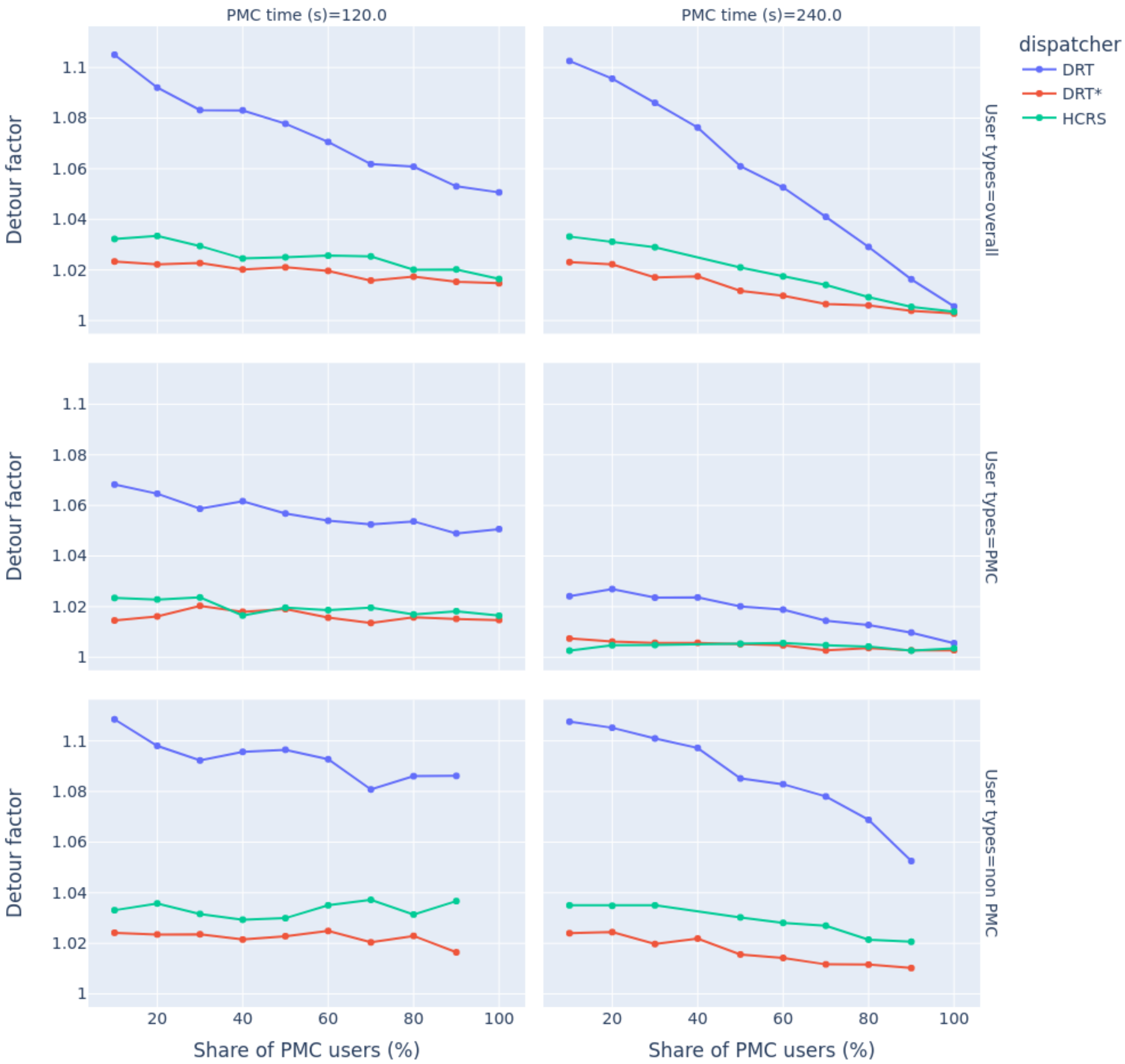

Figure 15: Observed detour factors under settings with various shares of PMC users with a fleet size of 100 vehicles, comparing across algorithms.

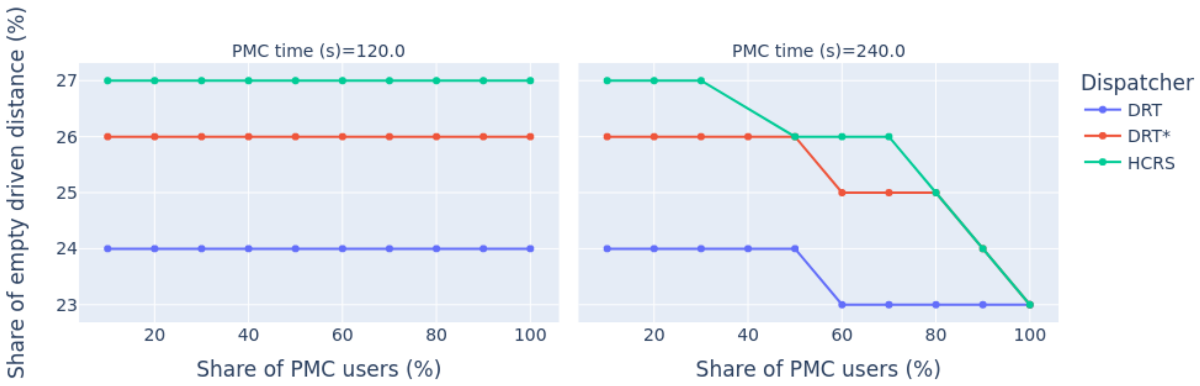

Figure 16: Observed shares of empty driven vehicle distances with a 100 vehicles fleet under various shares of PMC, comparing across algorithms.

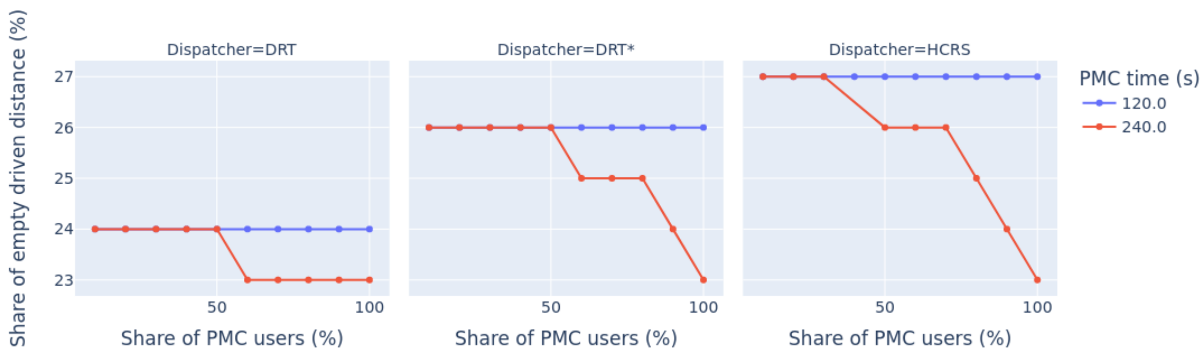

Figure 17: Observed shares of empty driven vehicle distances with a 100 vehicles fleet under various shares of PMC, comparing between PMC times.

4.4.3 Fleet sizing with a fixed share of PMC users

We now consider the same task of fleet sizing but with 50% of PMC users. This percentage is chosen arbitrarily to illustrate the impact of introducing PMC users under various fleet sizes. Analyses performed using other percentages were performed and led to the same conclusions as the ones presented below. The visualisations are similar to the previous subsection with two figures for each measure, one comparing directly between user types and the other between algorithms.

Figure 18 compares the rejection rates observed for each type of users and overall according to the fleet size. With a PMC time of 120 seconds, PMC users experience more rejections than non-PMC ones with small fleets but an optimal and almost-equal performance is reached starting from 200 vehicles. With a PMC time of 240 seconds however, the rejection rate for PMC users still does not match the one of non-PMC users. We conclude that the presence of PMC users strongly impacts the fleet sizing. Figure 19 shows the same measures compared directly between algorithms. These results suggest that the presence of PMC users does not impact the relative performance of the algorithms for each user type as similar rejection rates are observed across algorithms similarly to the results shown Figure 4.

Figure 20 compares the wait times for each user type according to the fleet size. In all evaluated settings, PMC users wait longer times in average for pick-up and increasing the fleet size does not bridge the gap in experienced quality. When the PMC time increases from 120 to 240 seconds, the difference between wait times for the users also increase. The comparison between algorithms shown in Figure 21 are generally consistent with the observations reported from the reference simulation shown in Figure 5. A special case is the measure of wait times experienced by PMC users with a PMC time of 240 seconds where DRT performs better than HCRS.

Detour factors are the only aspect on which PMC users are at least as well-served as non-PMC ones as shown in Figure 22. For the DRT* and HCRS algorithms, the detour factor for PMC users is better than for non-PMC ones and the two values progressively converge to each other when increasing the fleet size. Under the DRT algorithm, PMC users experience substantially less detours than non-PMC ones even with high fleet sizes. Figure 23 focuses on the relative performance between the algorithms in terms of detour factors within the fleet sizing. We notice that with a PMC time of 240s, the gap between the detour factors of DRT algorithms and the other algorithms is considerably smaller than with a PMC time of 120s.

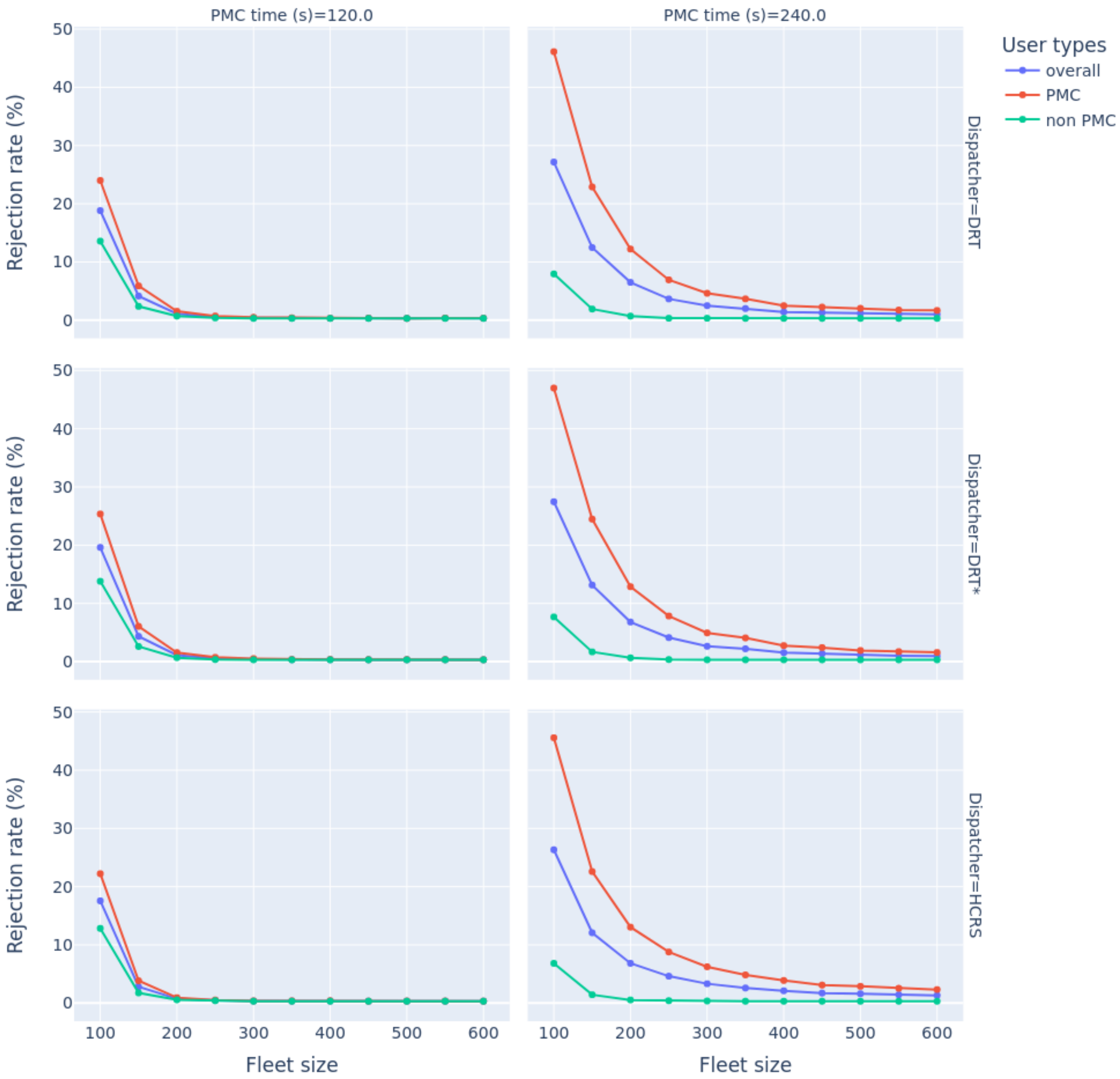

Figure 18: Rejection rates observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across user types.

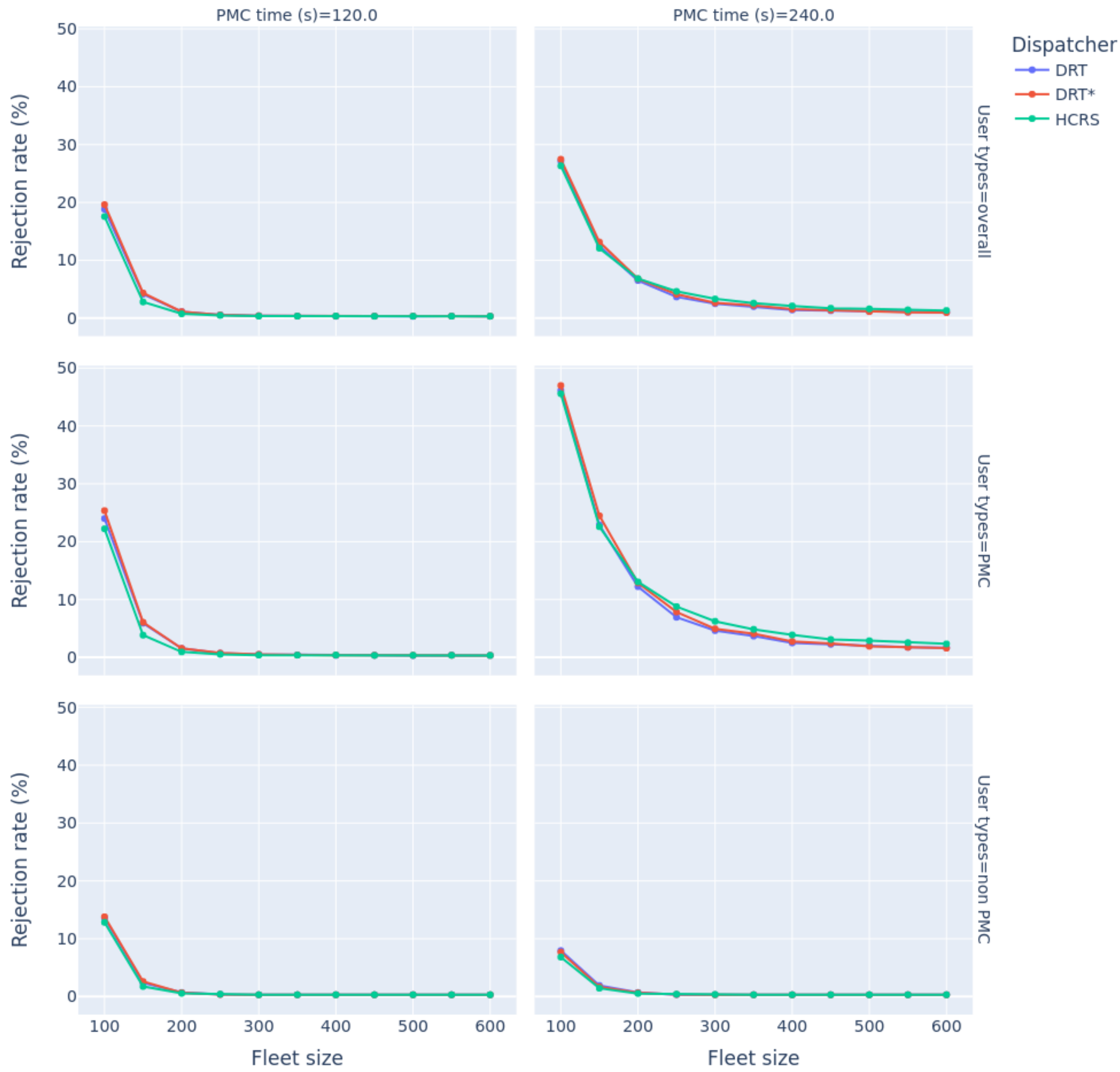

Figure 19: Rejection rates observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across algorithms.

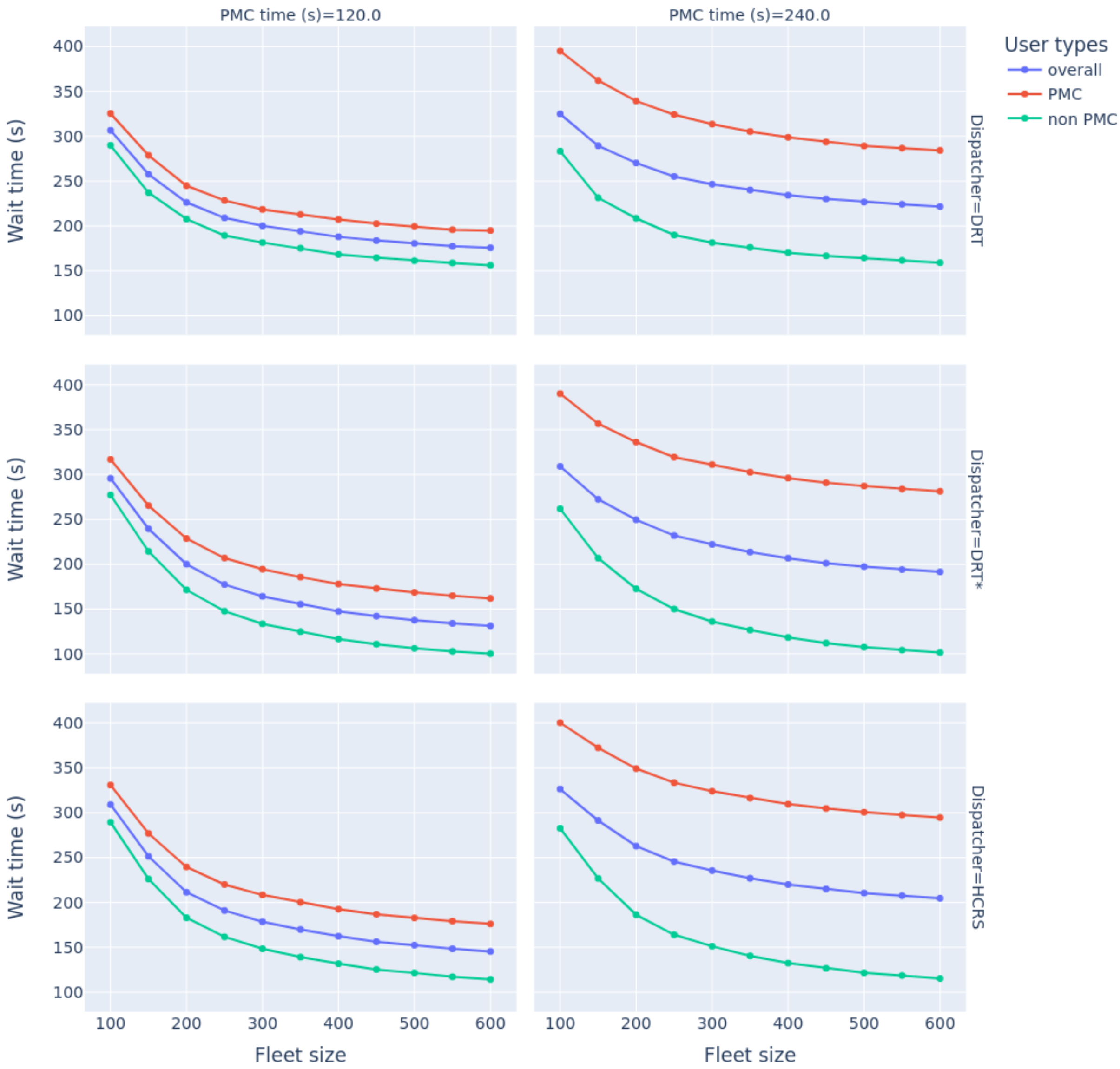

Figure 20: Wait times observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across user types.

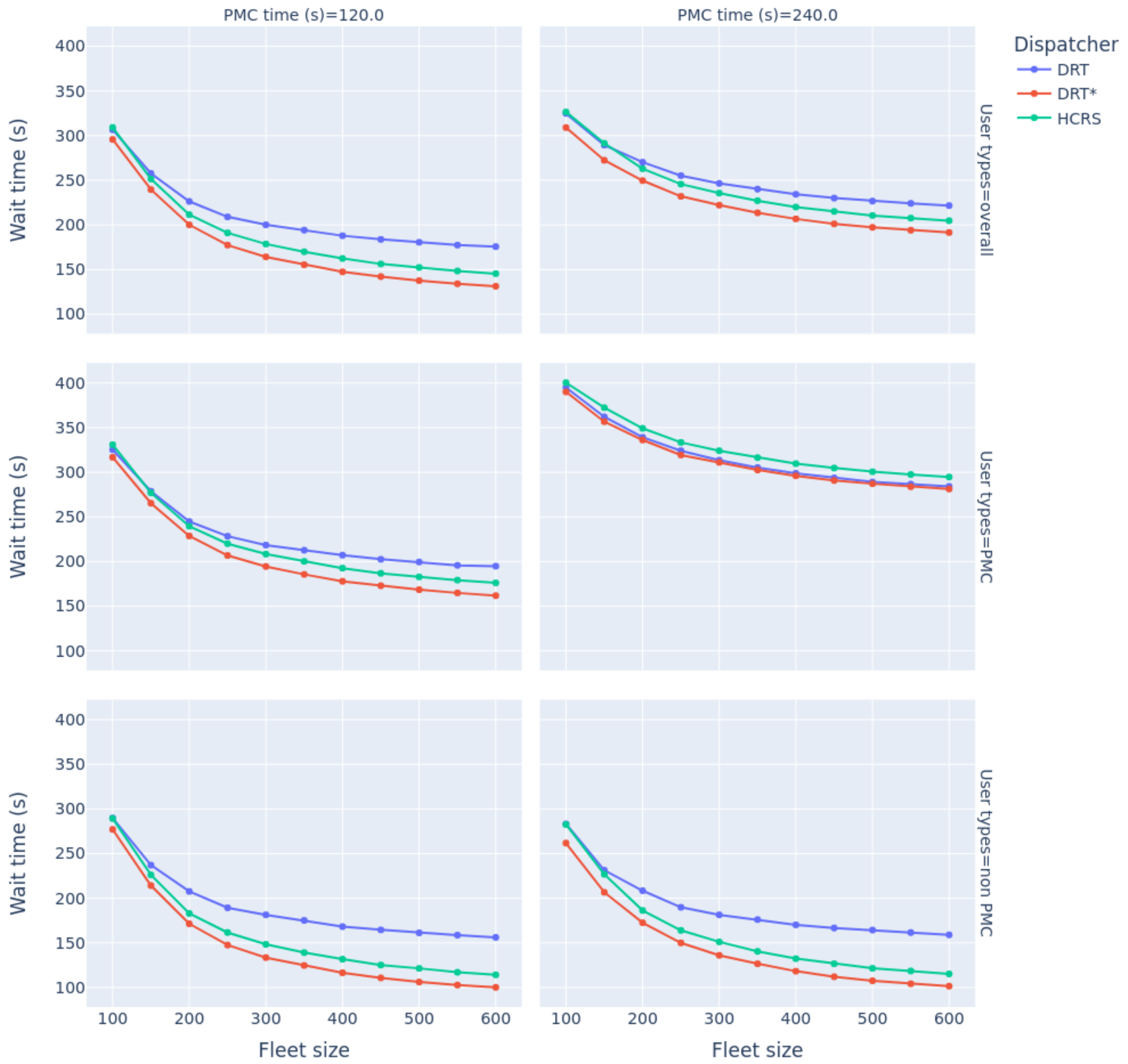

Figure 21: Wait times observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across user algorithms.

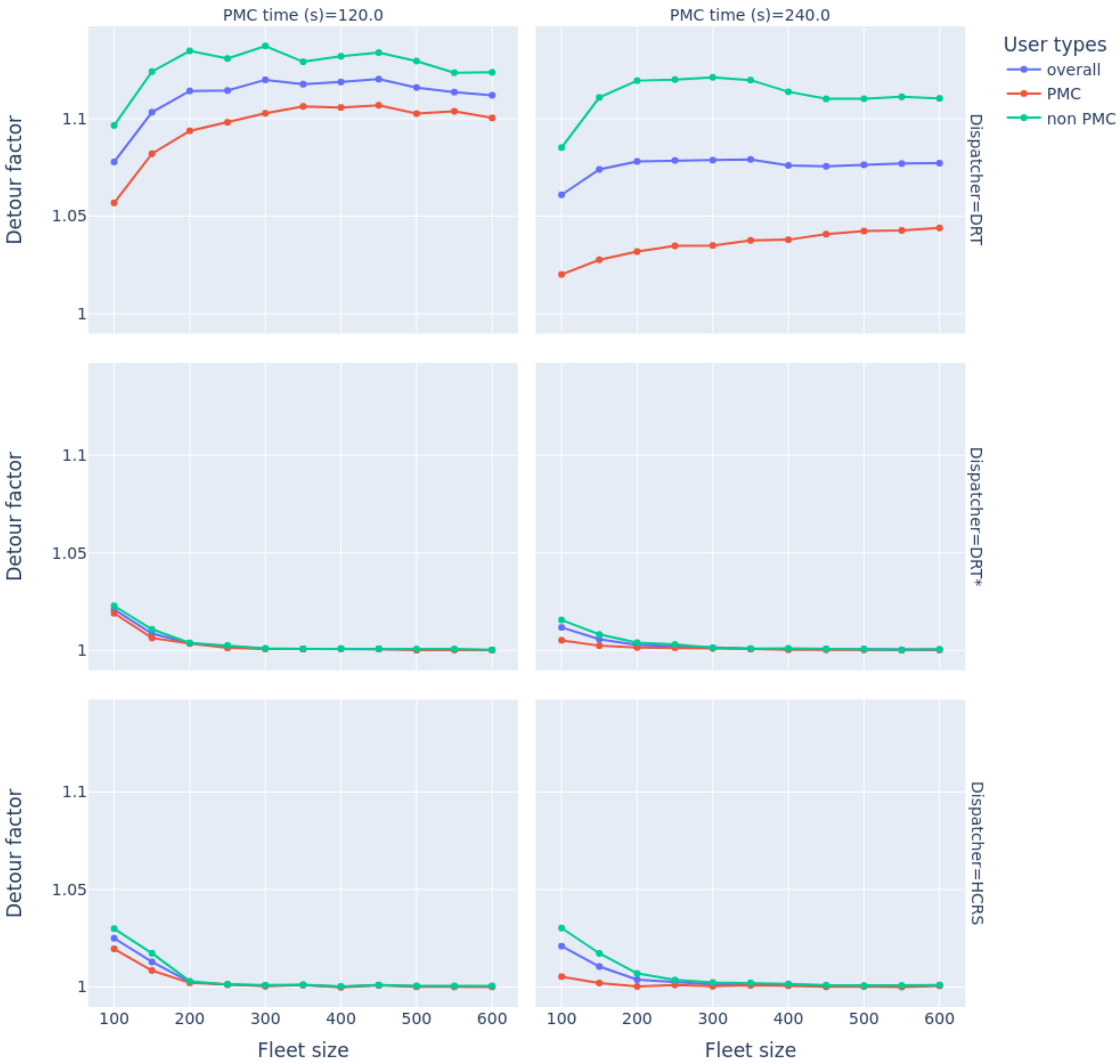

Figure 22: Detour factors observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across user types.

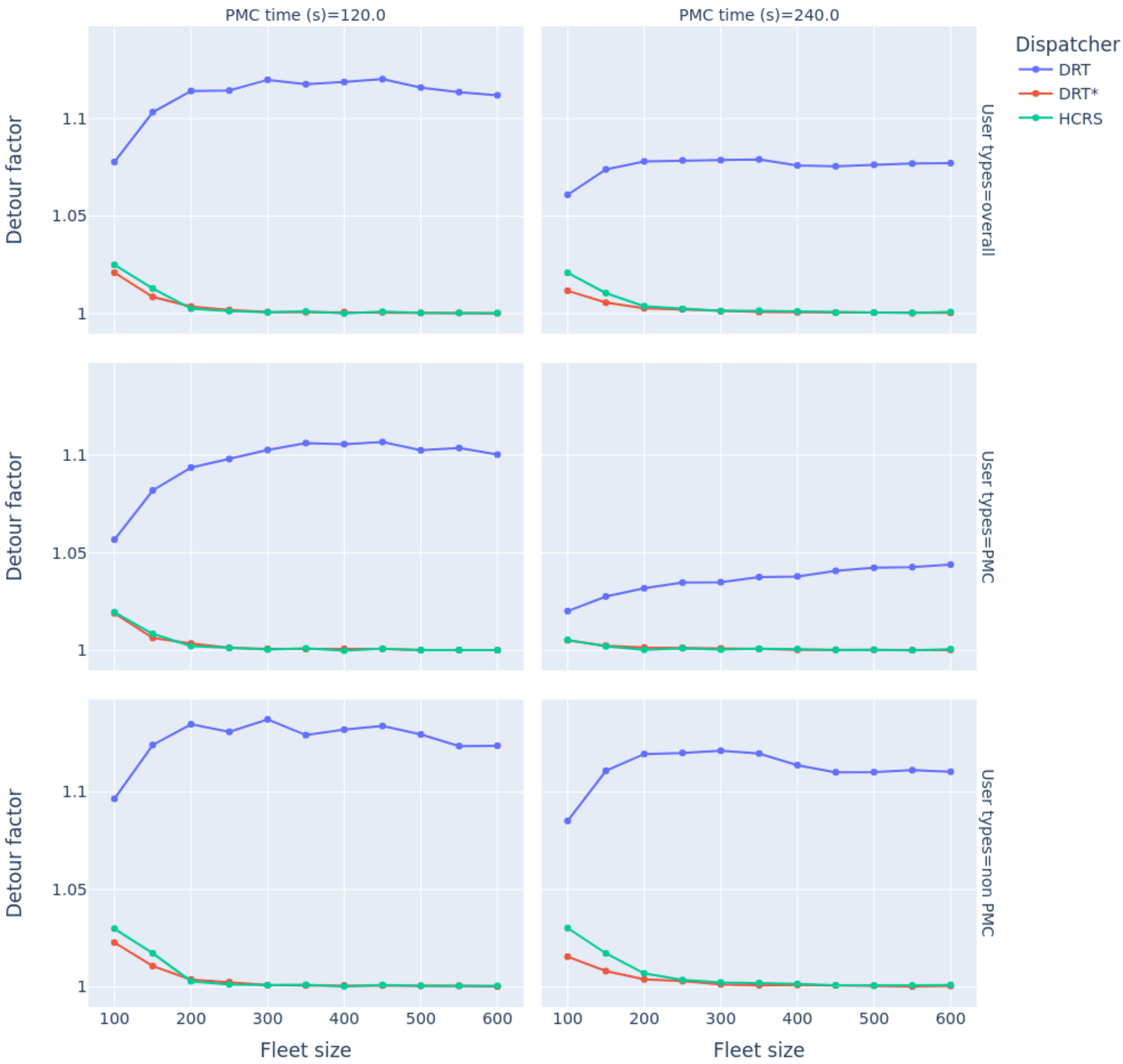

Figure 23: Detour factors observed with various fleet sizes for each of the algorithms on a setting with 50% of PMC users, comparing across algorithms.

5 Mitigation measures

The analysis presented in the previous chapter identifies inclusivity issues of existing fleet dispatching algorithms that cause PMC users to experience more request rejections and longer wait times than non-PMC ones. In some cases, increasing the fleet size reduces the gap in the service quality perceived by user types. However, this still does not increase the inclusivity as the algorithms since they will systematically favour non-PMC users against PMC ones. Strategies that explicitly consider PMC users are required.

As the assessment present in the previous chapter is the first in the literature to investigate the inclusivity implications of fleet dispatching algorithms and uncover issues related to the handling of PMC users, proposing strategies to address these issues can only be an exploratory activity. In the following, we attempt to provide mitigation strategies for the algorithms considered here. By mitigation we refer to being able to achieve a quality of service for PMC users that at least matches the quality of service for non-PMC ones.

The DRT and DRT* algorithms use an insertion heuristic to add traveller requests into vehicles' schedules once and for all. Whereas HCRS use MILP to continuously re-schedule the vehicles' plans. Different mitigation strategies are then possible or efficient according to the algorithms. In this chapter, we detail our efforts performed in this direction and present one mitigation strategy per algorithm and the obtained results.

5.1 Demand-Responsive Transport (DRT) Algorithm

For each traveller request, the DRT (and DRT*) algorithm loops through the vehicle and calculates the set of possible insertions of the request into the vehicle's schedule. Insertions amongst all the vehicles are compared and the best one is selected (thus selecting one vehicle) for the request being processed. Since requests are processed sequentially in the order of arrival, insertions of different requests are never directly compared. This means that altering the insertion selection criterion to favour PMC users would have no impact.

One theoretically sound approach would be to keep requests pending until a large enough set of requests is available and then perform the dispatch decision as usual but starting by requests originating from PMC users first. This way PMC users will be treated with more priority. However the drawback behind this approach is that the timespan during which requests are kept pending is counted as waiting time, thus consuming from the available maximum wait time specified in the quality constraint. For instance, if the maximum wait time is set to 10 minutes and the request is kept pending for 5 minutes before being processed, the dispatching algorithm must deliver a vehicle to the user within 5 minutes. This increases the overall probability of rejections, for both PMC and non-PMC users. The implementation and evaluation of this approach show that neither overall performance nor inclusivity benefit from this strategy.

The approach that we chose to follow to mitigate the inclusivity consists in using prebooking exclusively for PMC users. The prebooking implementation described in Section 3.2 was able to show increased quality of service for prebooked requests when these are chosen randomly, we then propose to use it in a targeted manner for PMC users. In practice, the exclusive access of PMC users to prebooking could be enabled by PMC-specific accounts for accessing the CCAM service.

We evaluate here the impact of introducing prebooking for PMC users. As for the previous evaluations described in this deliverable, we evaluate this strategy varying both the share of PMC users, the PMC time and the fleet size. In our simulations, PMC users submit their requests for pre-booking 4 hours in advance. We focus in the following results on the DRT* algorithms as the results obtained with DRT are very similar.

Figure 24 shows the rejection rates per user type in function of the proportion of PMC users with a 100 vehicles fleet, with settings for PMC times in columns and the use or not of prebooking for PMC users in rows. The no prebooking parts of the figure are exactly the same as the ones shown in Figure 10 where PMC users experience higher rejection rates than non-vulnerable ones. Using prebooking for PMC users allows to turn around the quality of the service in favour of PMC users. The latter experience much less rejections than non-PMC ones and under low shares of PMC users, the rejection rate for this category is near-zero. The rejection rates for both user types increase when the proportion of PMC increases due to reaching the limit capacity of the fleet, the more PMC users there are the less the sharing of vehicles between users is possible. This causes less vehicles to be available later either for prebooked requests or for immediate ones.

The impact of prebooking on the rejection rate of each user-type is more apparent in Figure 25. Whereas non-PMC users always have worse rejection rates when prebooking of PMC users is activated than when it is not, the rejection rate for PMC users increases as their proportion increases until prebooking produces a similar rejection rate to no-prebooking and eventually a higher one at the limit of 100%.

Figure 26 shows the impact of this mitigation strategies on the relative quality perceived by each user-type in terms of wait times. Under the 120s PMC time setting, the use of prebooking for PMC users results in them having to wait less for the vehicles regardless of their share in the overall demand. However, in the 240s setting, this is true only with less than 40% of PMC users. Figure 27 focuses on the impact of prebooking on each user-type's wait times and shows that PMC users greatly benefit from the strategy in all settings. On the other hand non-PMC users see an increase in wait times when the proportion of PMC users is blow 50% but then decreases afterwards.

The impact of prebooking on the detour factor is shown in Figure 28 for each user-type. We note that the detour factors increase for both types after the introduction of prebooking for PMC users. This is due to the fact that the fleet is now able to serve more PMC users, which forces the vehicles to share more trips in order to satisfy the increased demand. Moreover, as the share of the PMC users converges to 100%, the detour factors for them converge to the detour factors without prebooking.

Regarding the fleet sizing under the prebooking strategy, we again fix the share of PMC users to 50% and vary the fleet size as well as the dispatching algorithm (DRT or DRT*). We notice in Figure 29 that even with low fleet sizes, PMC users still experience close to no rejections at all. The increase in number of vehicles mainly benefits the non-PMC users who see their rejection rate improve. The latter user-group is strongly impacted by this prioritisation of PMC users as their rejection rates dramatically increase with prebooking when the fleet size is less than 200 vehicles (see Figure 30).

In conclusion, by allowing PMC users only to book their trips in advance, the DRT (and DRT*) algorithm is able to secure the vehicles for these users and ensure a quality of service for them that is at least as good as the one for non-PMC users. Our results show that the drawbacks for non-PMC users can be large, especially with smaller fleets.

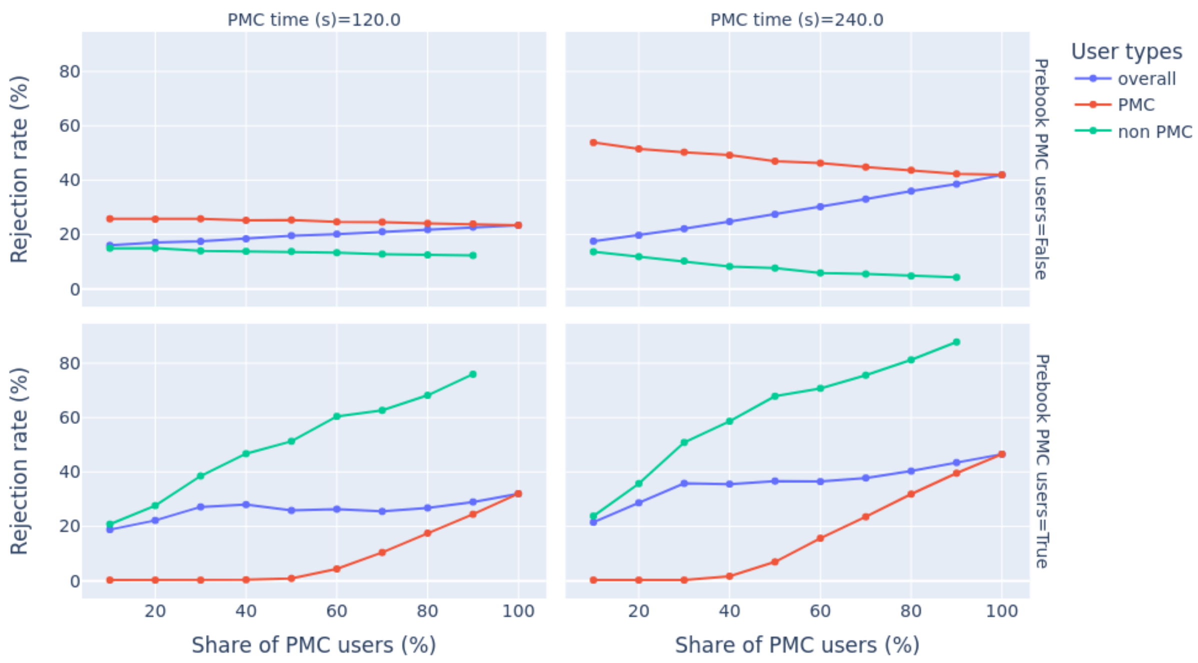

Figure 24: Rejection rates in function of the share of PMC users observed for with a 100 vehicle fleet sizes with and without prebooking of PMC users using the DRT algorithm, comparing between user types with and without prebooking.*

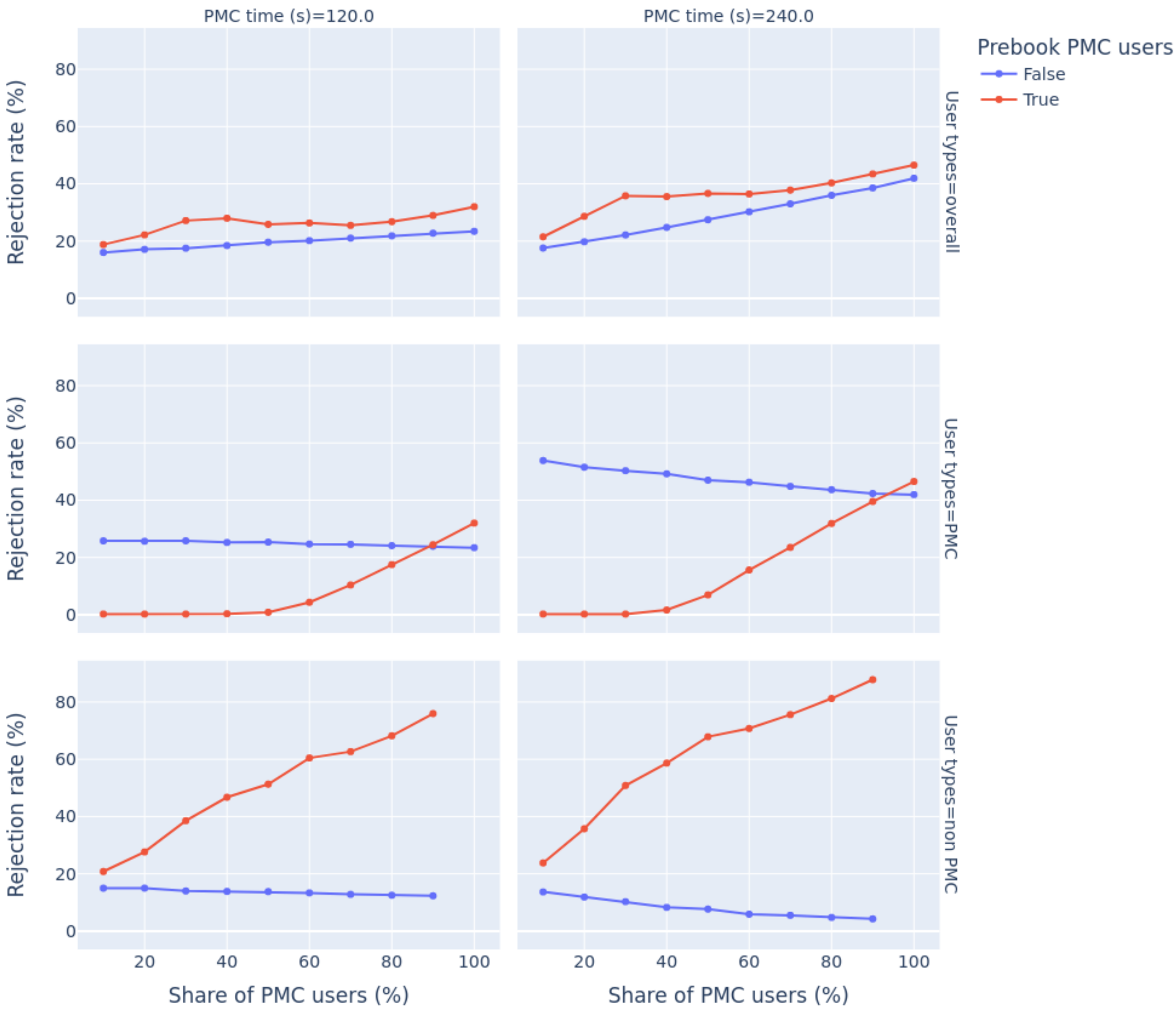

Figure 25: Rejection rates in function of the share of PMC users observed for with a 100 vehicle fleet sizes with and without prebooking of PMC users using the DRT algorithm, comparing between prebooking and no prebooking for each user type.*

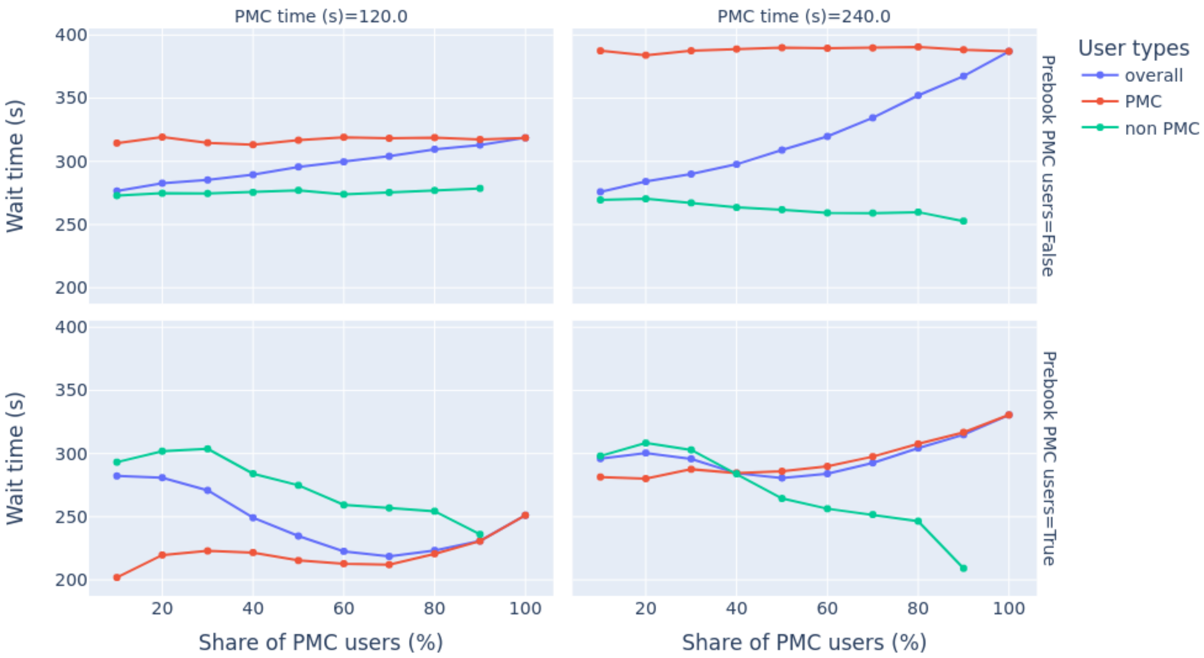

Figure 26: Wait times in function of the share of PMC users observed for with a 100 vehicle fleet sizes with and without prebooking of PMC users using the DRT algorithm, comparing between user types with and without prebooking.*

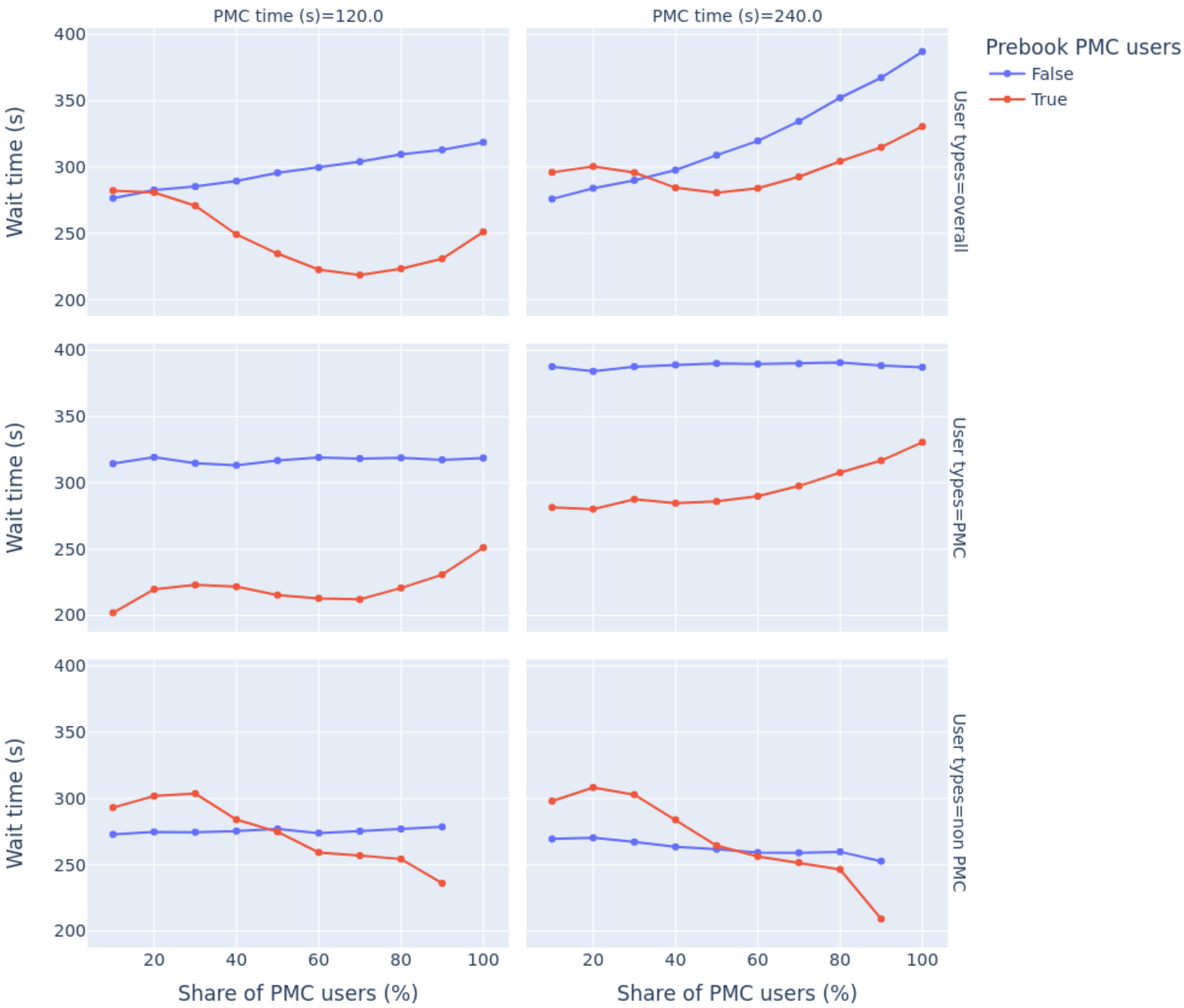

Figure 27: Wait times in function of the share of PMC users observed for with a 100 vehicle fleet sizes with and without prebooking of PMC users using the DRT algorithm, comparing between prebooking and no prebooking for each user type.*

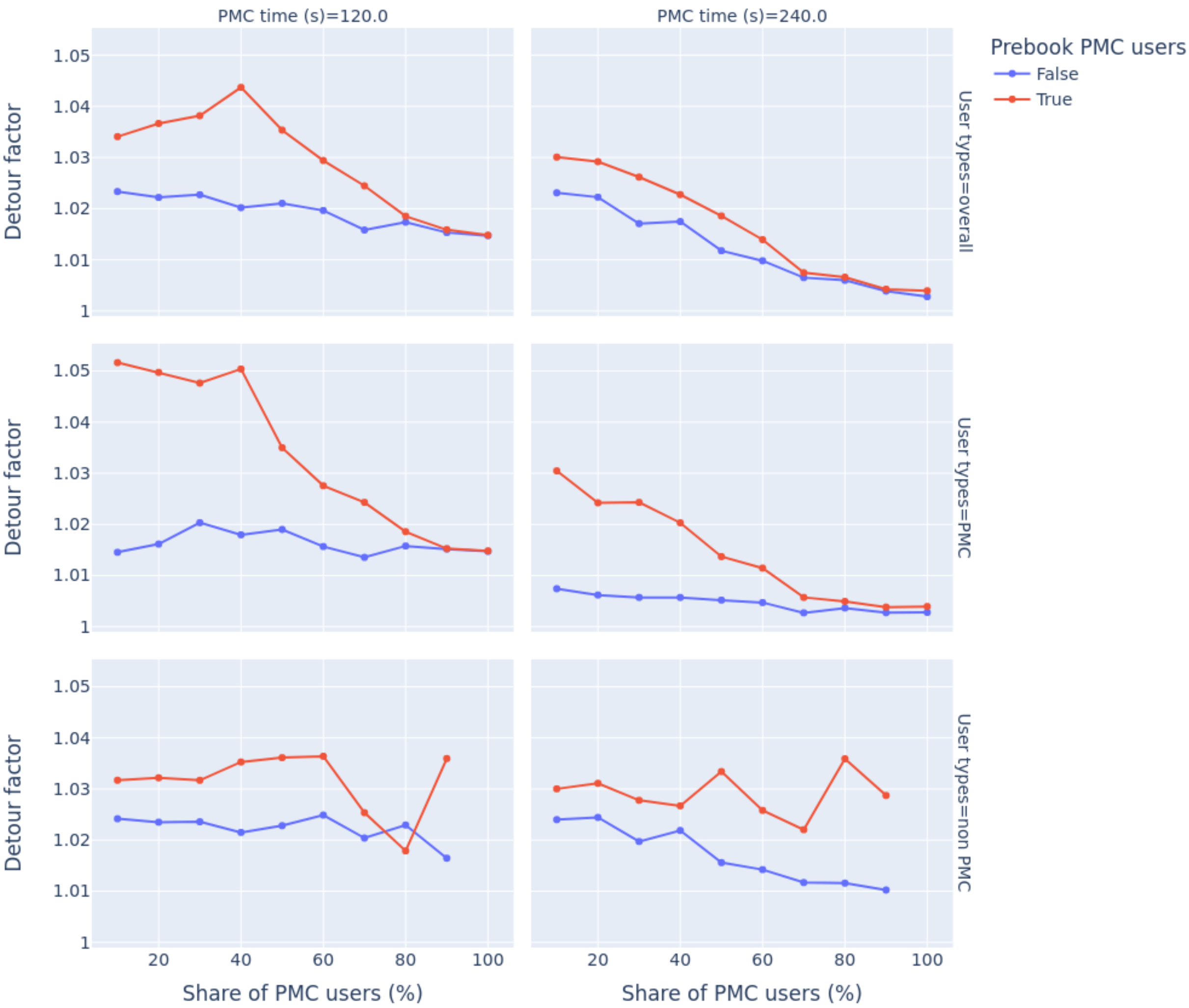

Figure 28: Detour factors in function of the share of PMC users observed for with a 100 vehicle fleet sizes with and without prebooking of PMC users using the DRT algorithm, comparing between prebooking and no prebooking for each user type.*

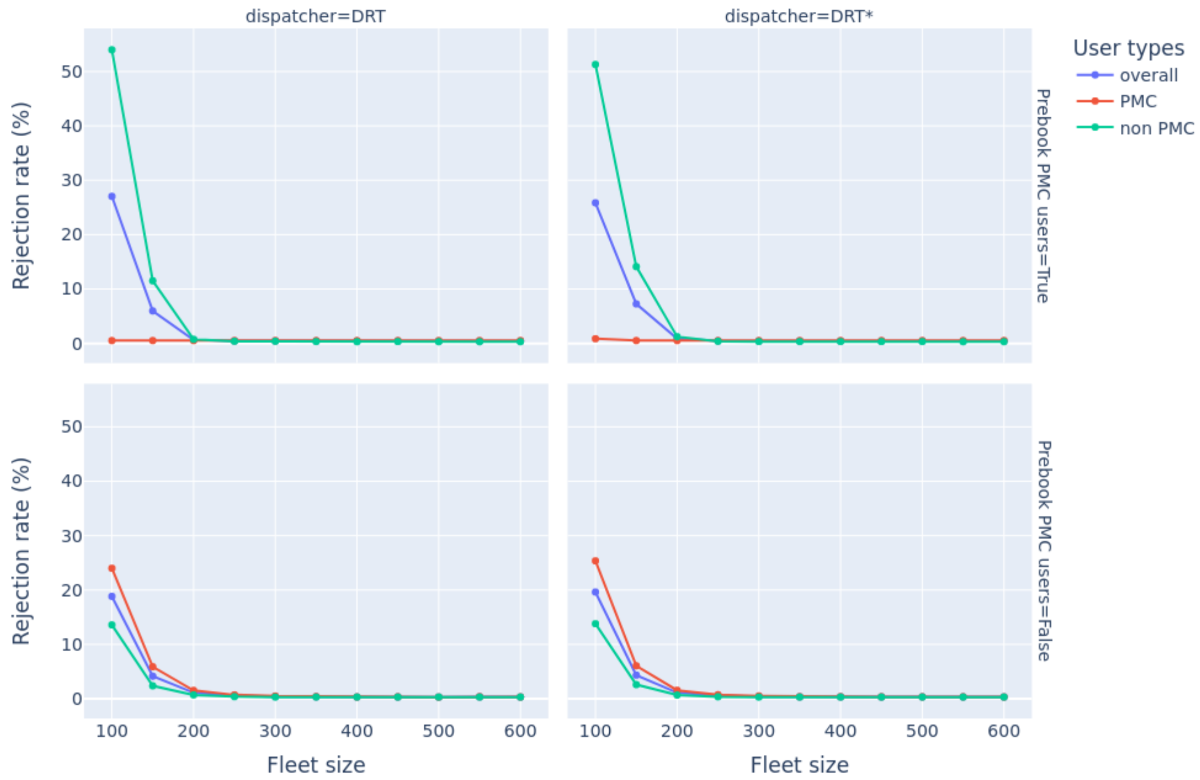

Figure 29: Rejection rates in function of the fleet size observed on simulations with 50% of PMC users with and without prebooking of PMC users. Comparing between user types.

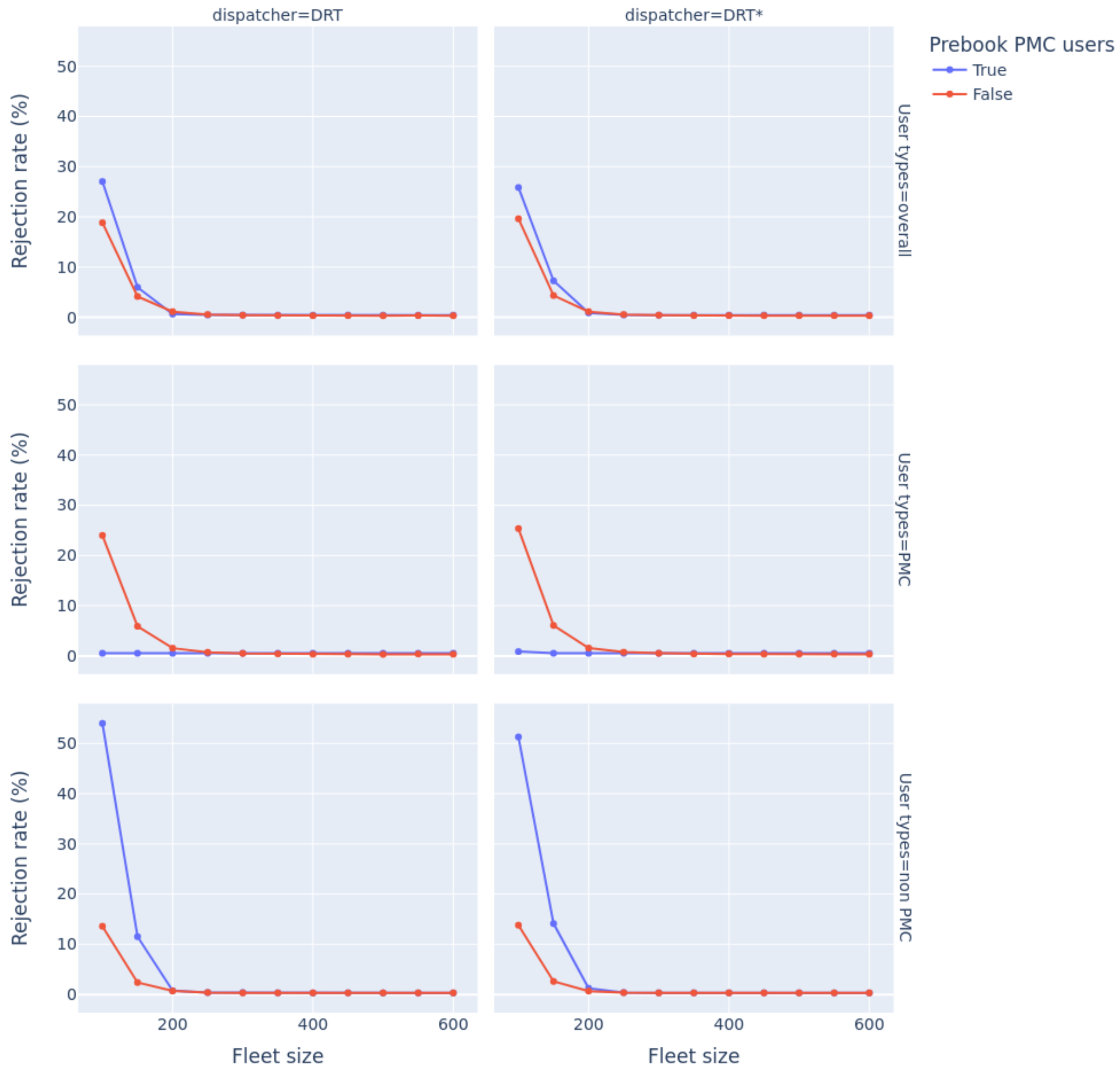

Figure 30: Rejection rates in function of the fleet size observed for each user type on simulations with 50% of PMC users. Comparing between prebooking and no prebooking for PMC users.

5.2 High-Capacity Ride Sharing (HCRS) Algorithm

As detailed in Section 4.2, the HCRS algorithm is fundamentally different from the DRT algorithm as the former uses an exact MILP approach to continuously re-schedule the vehicles' plans considering newly received requests, while the latter uses an insertion heuristic to match a received request to a vehicle without questioning the matching later. Due to these differences, the mitigation strategies envisioned for the DRT algorithm are not all suitable in the context of the HCRS algorithm.

Regarding the batching of requests, since the HCRS algorithm re-evaluates the interest of already planned assignments in the light of new requests, it would only hinder the service quality to keep coming requests pending for some time before performing a decision.

The prebooking mitigation strategy that is proposed for the DRT algorithm is also not suitable here. Even if it is theoretically possible to use the prebooking feature described in Section 3.2 with the HCRS algorithm, our experiments show that this is not practical due to the excessive run times that result from this. The MILP approach of HCRS algorithm is highly sensitive to the number of requests pending in the system and a prebooking strategy with a 4 hours horizon results in much more pending requests than the regular no-prebooking strategy that only keep the request for a maximum of 10 minutes (The maximum wait time).

Our mitigation strategy for the HCRS algorithm lies in the objective function of the HCRS algorithm presented in Equation 1. In this objective function, decisions regarding different requests are jointly considered and compared. The Q and Q' penalties are applied respectively for every rejection of a newly submitted requests or a previously assigned ones. In order to address the inclusivity issue of the HCRS algorithm, we introduce new penalties Q_v and Q'_v that are specific to PMC users such as:

$$Q_v \ (resp \ \ Q'_v) \ = \frac{t_v}{t_{min}} \cdot \alpha \cdot Q \ (resp \ Q')$$

Where t_v is the time required by PMC users for pick-up and drop-off, t_{min} the time required by non-PMC users and α a constant. The $\frac{t_v}{t_{min}}$ ratio is included in the formula to ensure that the more time is required by PMC users, the higher the penalty for rejecting their requests is.

We evaluate this approach by varying the setting-related aspects, as in the previous assessments, as well as the value of the α parameter in the equation above. Figure 31 shows the obtained rejection rates under three proportions of PMC users (20%, 40% and 60%) with 100 vehicles. In all settings, increasing the value of α decreases the rejection rate for PMC users. However, the value at which rejection rates converge and whether better rates are achieved for PMC users than non-PMC ones depend on the setting. With a 120s PMC time, the mitigation strategy is able to achieve a rejection rate for PMC users starting from a value of 500. With a PMC time of 240s, the PMC users' rejection rate converges at a value greater than the value at which the rejection rate of non-PMC ones converges.

In the following, we consider only settings with a PMC time of 120s and identify the value $\alpha \ = 800$ for the PMC specific penalty parameter as a value that achieves the best rejection rates for PMC users across all settings. We then vary the share of PMC users to assess how this mitigation strategy affects other quality aspects.

Figure 32 shows the impact of the PMC specific penalty on the relative levels of rejections between user-types across various proportions of PMC users with a 100 vehicles fleet. This strategy is able to always favour PMC users with less rejection rates than non-PMC ones. However, when the share of PMC users increases, the rejection rate substantially increases. In Figure 33 comparing rejection rates before and after the mitigation strategy for each user-type, we see that the impact on PMC users is always positive, with the gap to the no-mitigation values decreasing with more PMC users. Non-PMC users witness a major increase in rejections due to the mitigation strategy and the increase in PMC users further increases the gap.

Regarding wait times, we notice in the results depicted in Figure 34 that the PMC specific penalty never results in better wait times for PMC users compared to non-PMC ones. This contrasts with the mitigation strategy proposed for the DRT algorithm (see Figure 26) where it is the case in some settings. This is due to the fact that the strategy evaluated here only addresses rejections in the objective functions, the time spent by users waiting for and in the service are not weighted differently by user-type. This strategy then encourages the algorithm only to accept requests from PMC users and not necessarily to further consider other quality criteria. This can be seen in Figure 35 which shows that the wait times experience by the PMC users before the mitigation strategy are very similar to after.

Moreover, this mitigation strategy has very little impact on the detour factors experienced by all users. As seen in Subsection 4.4.1, the detours produced by the HCRS algorithms are very small compared to the DRT and DRT* algorithms. Although the gap between user-types increases as shown in Figure 36, the values produced by the mitigation strategy very only slightly. This can also be seen in Figure 37 where detour factors before and after the mitigation strategy are compared for each user type.

Finally, we perform a fleet sizing with our mitigation strategy by fixing the share of PMC users to 50%, keeping the value $\alpha\ = 800$ and varying the fleet size. Figure 38 shows the rejection rates per fleet size compared between user user-types before and after the mitigation strategy. These results suggest that this strategy does not impact the necessary fleet size to attain near-zero rejected requests for all users. However, with smaller fleet sizes, it allows to prioritise PMC users. Figure 39 shows how this strategy can help decrease the likelihood of rejection for a PMC user, the counterpart being higher rejections for non-PMC users.

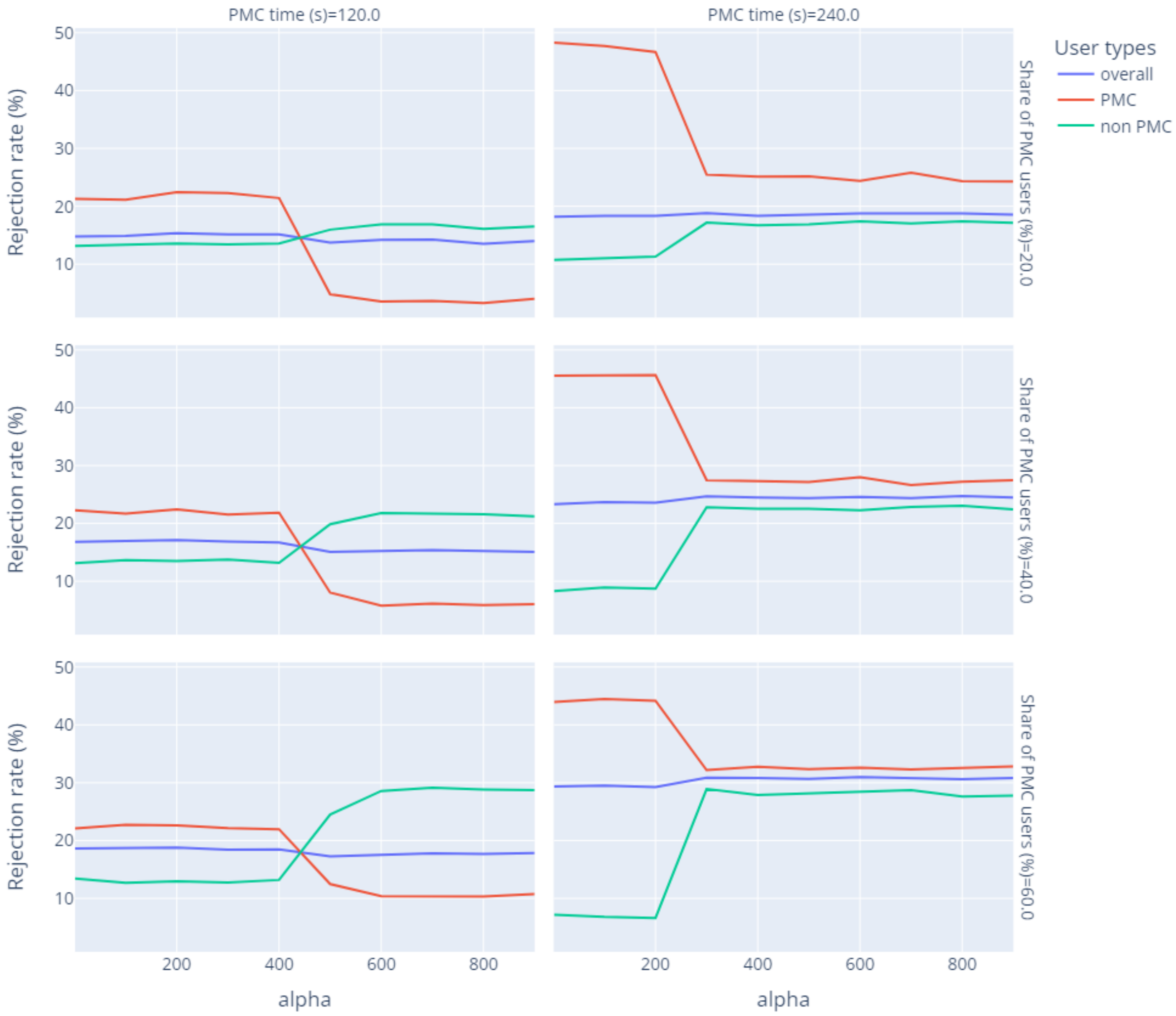

Figure 31: Rejection rates observed in simulations with various values of the α parameter of the PMC specific penalty in the HCRS algorithm.

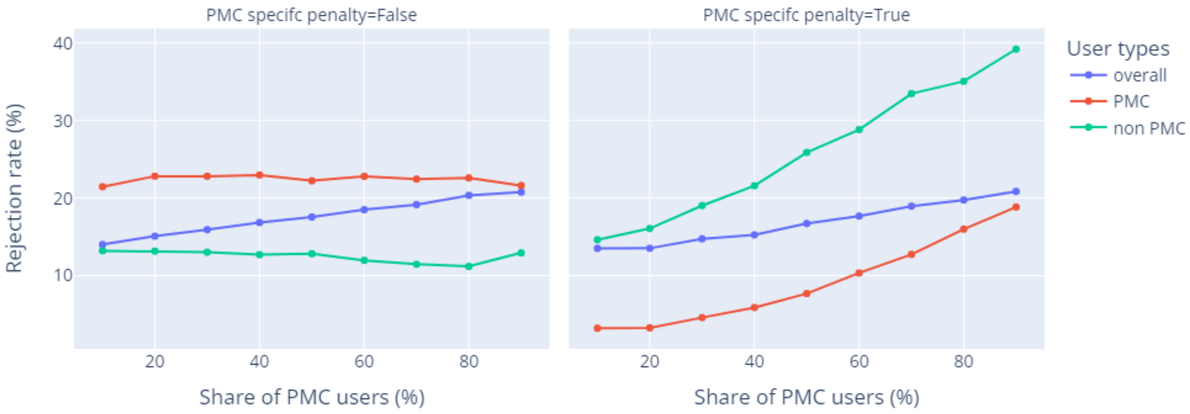

Figure 32: Rejection rates in function of the share of PMC users with and without the PMC specific penalty in the HCRS algorithm.

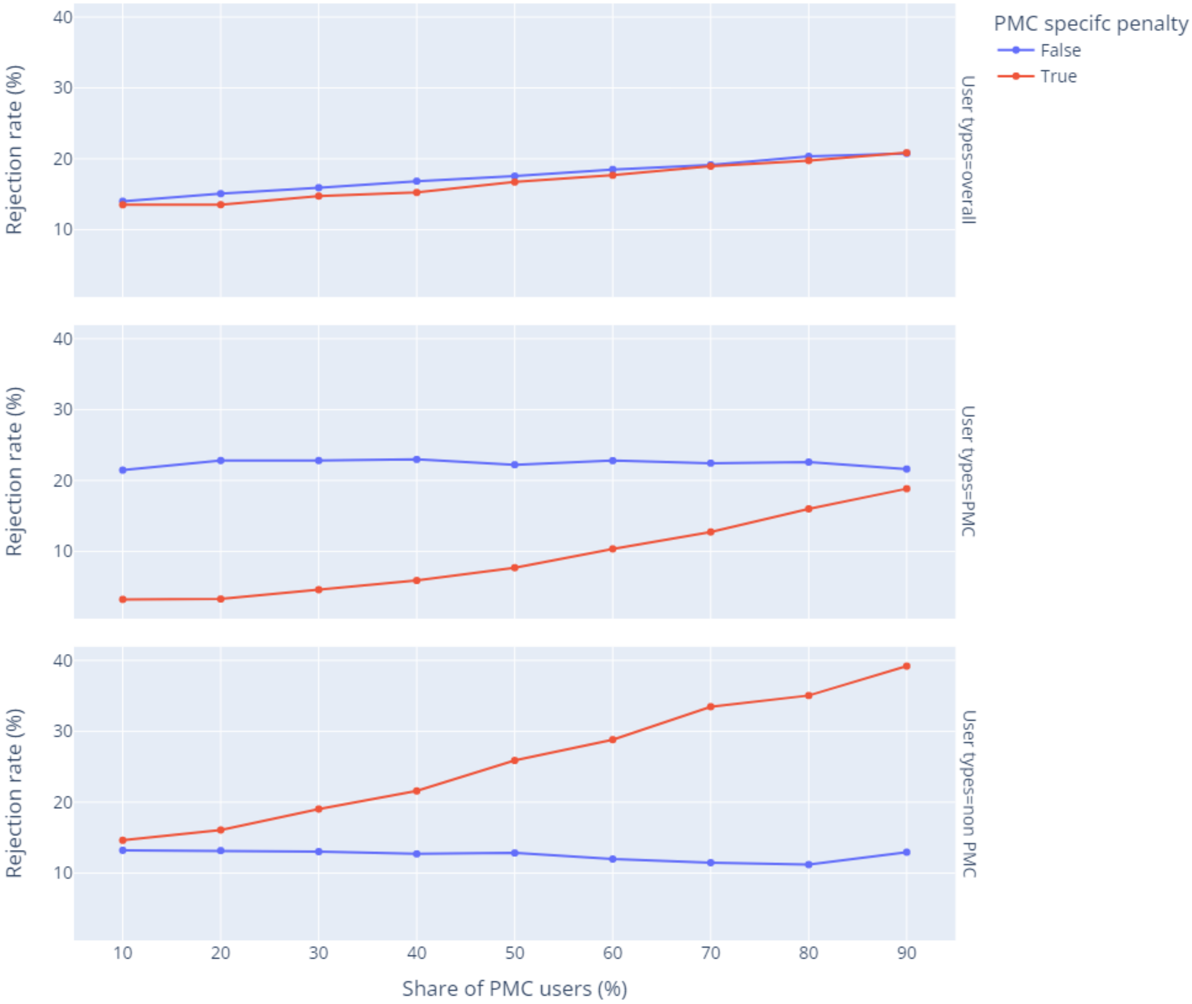

Figure 33: Rejection rates in function of the proportion of PMC compared before and after the PMC specific penalty in the HCRS algorithm.

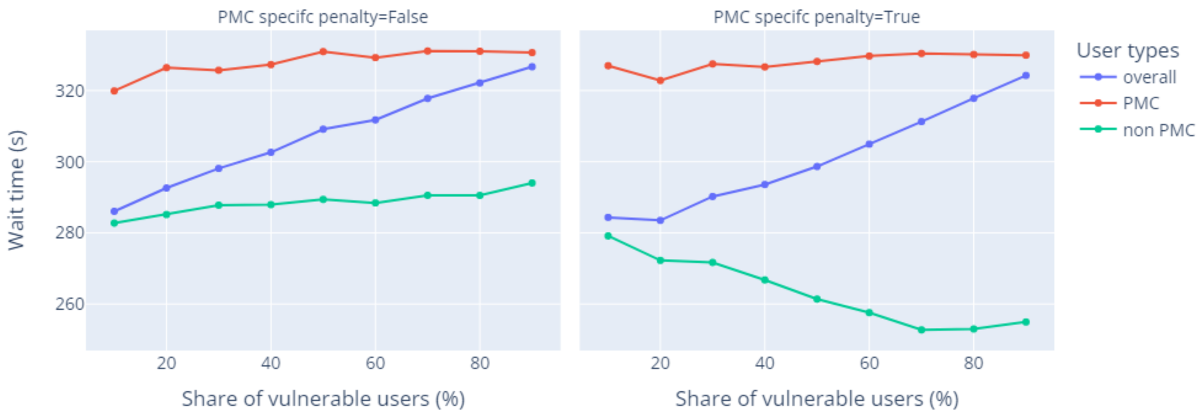

Figure 34: Wait times in function of the share of PMC users with and without the PMC specific penalty in the HCRS algorithm.

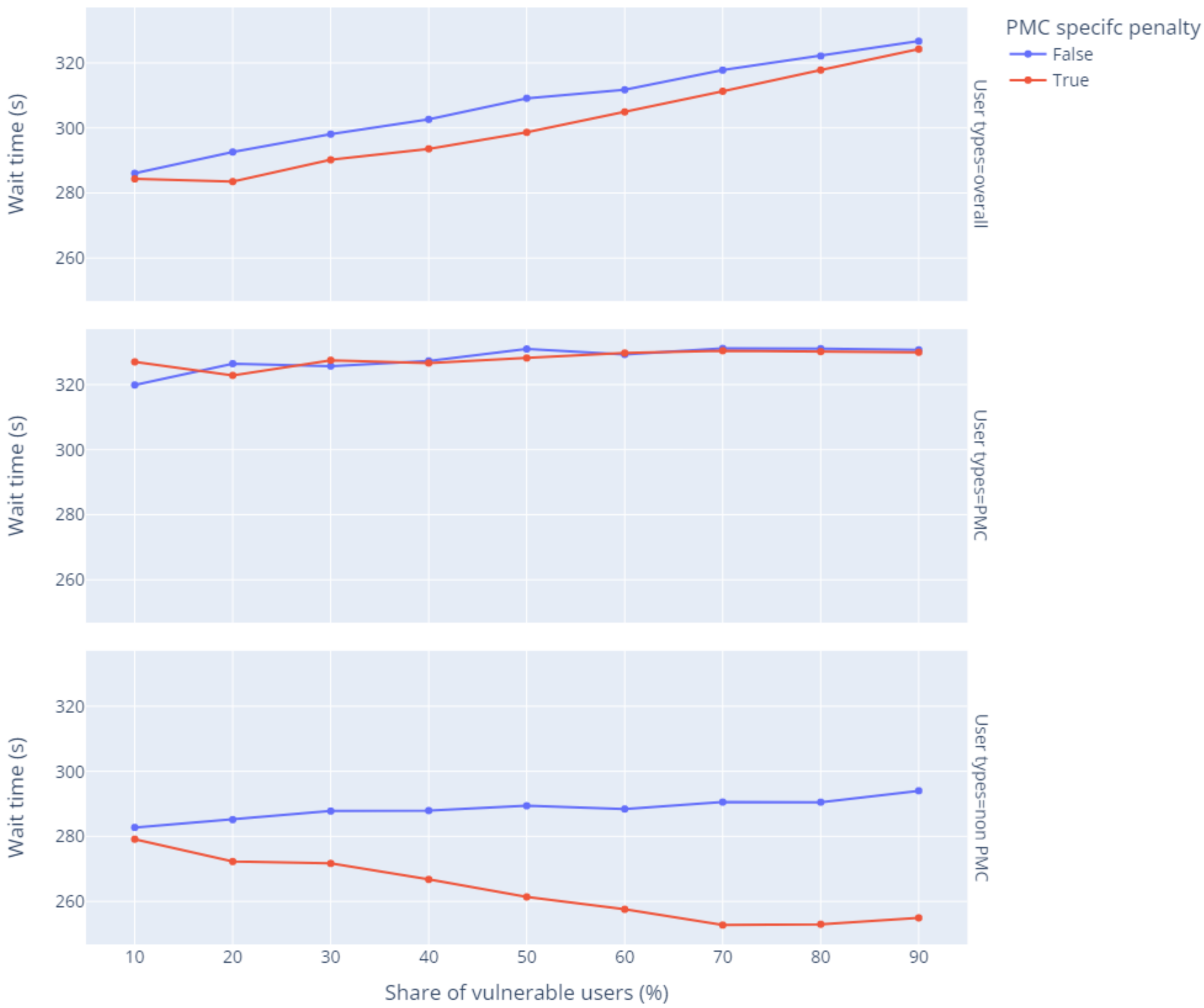

Figure 35: Wait times in function of the proportion of PMC compared before and after the PMC specific penalty in the HCRS algorithm.

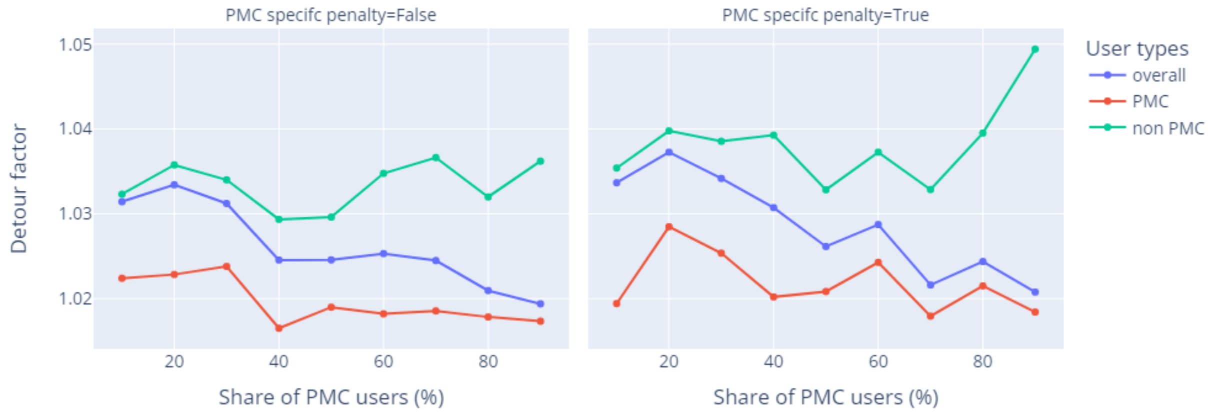

Figure 36: Detour factors in function of the share of PMC users with and without the PMC specific penalty in the HCRS algorithm.

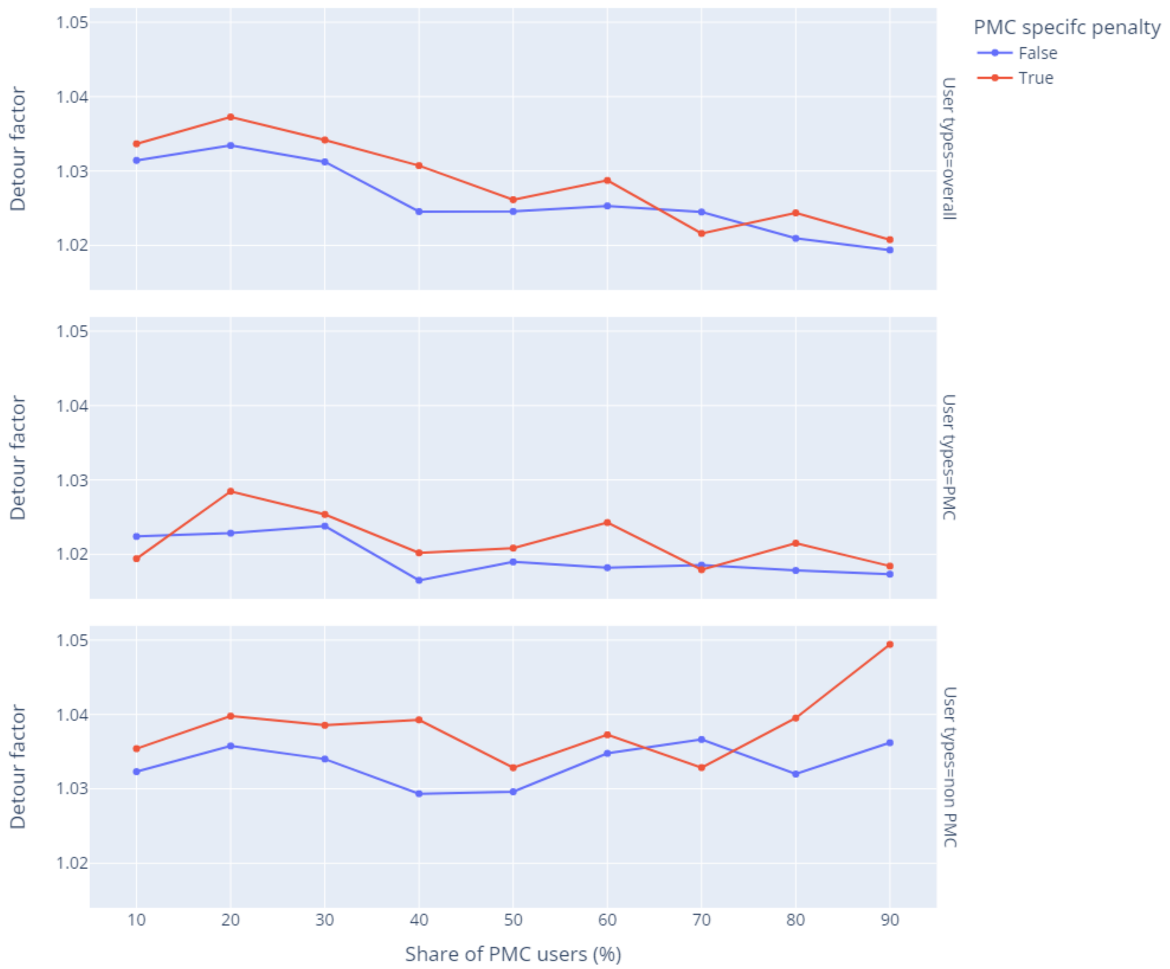

Figure 37: Detour factors in function of the proportion of PMC compared before and after the PMC specific penalty in the HCRS algorithm.

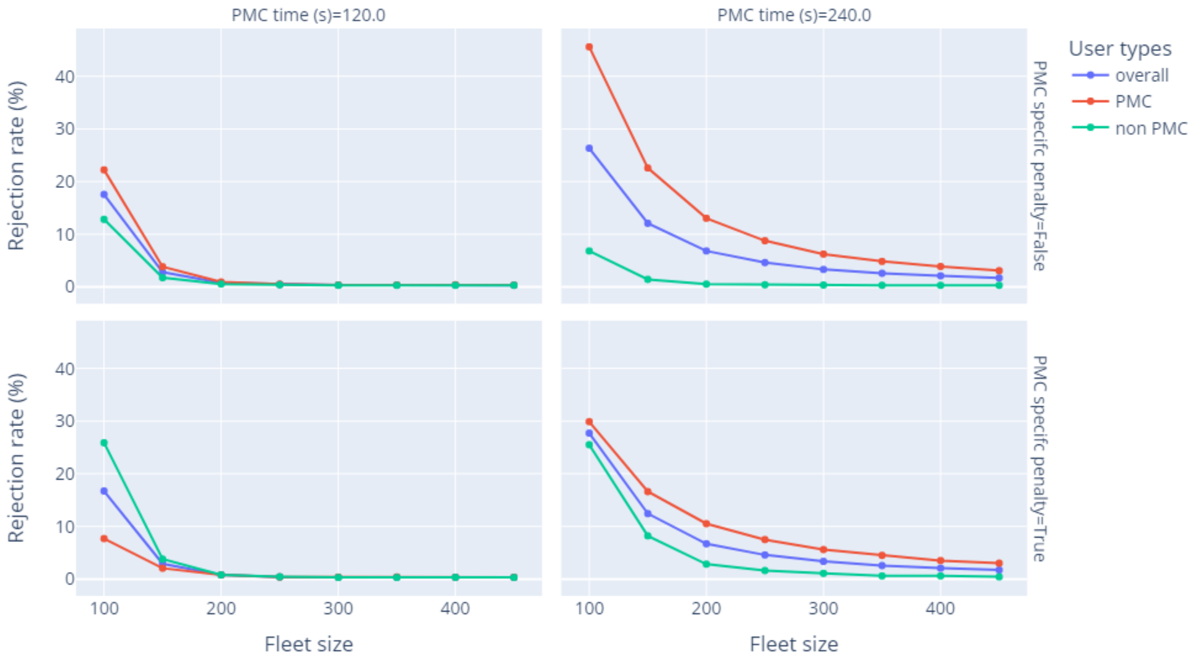

Figure 38: Rejection rates in function to fleet size compared between user-types with and without the PMC specific penalty in the HCRS algorithm.

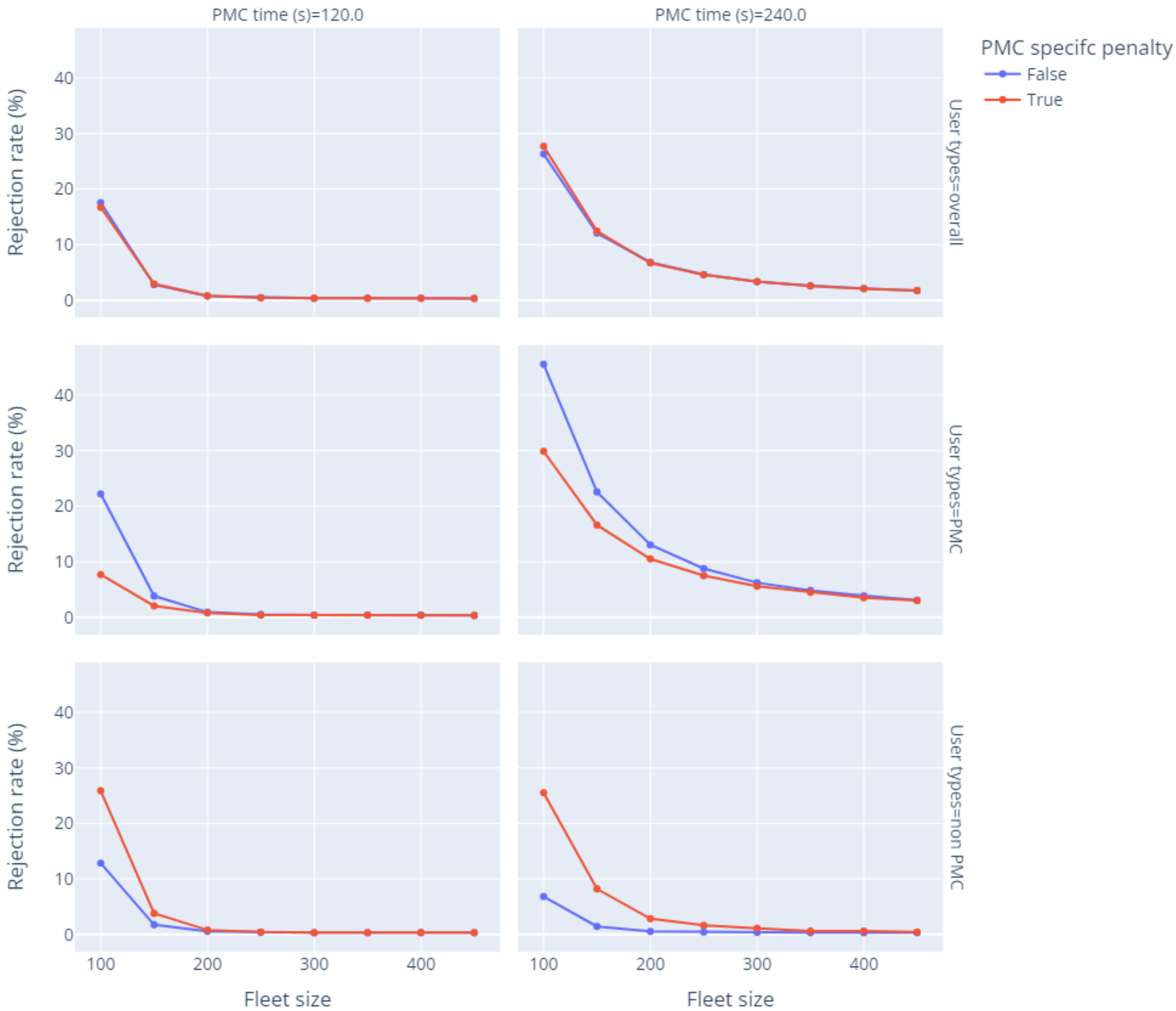

Figure 39: Rejection rates in function of the fleet size compared before and after the PMC specific penalty in the HCRS algorithm for each user-type.

6 Conclusion and next steps

In this deliverable, we successfully implemented the technical framework that will be applied in T3.7 of SINFONICA to assess different configurations of CCAM services on four concrete use cases in Hamburg, West Midlands, Noord-Brabant, and Trikala. While a large share of the base functionality is available now, we expect that the framework will be further extended during that activity, especially after having received specific input on the individual use cases with their local particularities.

In our analyses we already show various operational KPIs on the service that will also be studied in T3.7 (such as the rejection rate, wait times, detour factors). This analysis will be further extended with economic and social aspects in T3.7 which, for instance, also allows the spatial distribution of those indicators over the studies territory, discerning between areas with low and high level of service, depending on how a particular service has been parameterised. Given that the components developed for this first study are generic and can be used for SINFONICA research site, most of the effort in T3.7 will be directed towards building a user-friendly visualisation platform to allow non-experts to experiment first-hand.

In general, our contributions will allow much more detailed analyses of on-demand mobility systems using the MATSim framework in studies performed by us and other researchers. This way, the present deliverable has contributed considerably to open research.

On the specific topic of algorithmic discrimination, we have substantially advanced the state-of-the-art by showing that commonly used fleet management algorithms show discriminatory behaviour against requests with interaction times higher than average. While not shown in the present research, similar considerations can be thought of (and validated with our toolkit) with respect to requests that are located in rural and low-density areas and hence require an increased access time by the fleet vehicles.

For the particular case of increased interaction times, the question is how to interpret our findings beyond the pure numerical analysis for the sake of providing a full sensitivity analysis. On the one hand, we make a strong assumption that the algorithm knows in advance about the interaction time of a user. This is not necessarily the case in reality if an affected user makes a request through the usual booking channels. However, if a user needs to imply such information because he or she wants to order a wheelchair-accessible vehicle, the information is available to the operator. Even if the operator may not have the intention to discriminate against such users, they may feed this information into the dispatching systems as, otherwise, cascading effects in delays may be expected in the current functioning of those algorithms. If the off-the-shelf algorithms (in the lack of other options) are used naively, such requests will be disfavoured, despite the potential availability of properly equipped vehicles. Our experiments show that new algorithms with mitigation measures are required to avoid systematically providing lower rejection rates for PMC users.

In reality, it will most likely be the case that the operators are obliged in some way to provide inclusive mobility for PMC users, as part of their operating license. Such obligations may mean to guarantee transport of wheelchair-dependent people or to reach a certain minimum acceptance rate in rural and low-density areas. However, in practice, these guarantees may come with additional constraints, for instance, in the form of requests that need to be sent well in advance.

A key element to add to our assessment will be the economic analysis planned for T3.7. While our current operational simulations show that substantially increasing the fleet sizes allows an operator to provide similar acceptance rates for PMC and non-PMC users, in reality, this will be a large addition of capacity to serve (despite our full parametric tests) a relatively low number of requests. The resulting question is whether an operator could provide such a substantially higher fleet size just for the rare case of having to serve PMC requests on an infrequent basis. The answer may be "no" as the fleet size of the service would be highly limited by its economic feasibility. A service providing large additional capacity reserves on a perpetual basis may become prohibitively expensive such that the potential mobility of the entire user base may be impaired. Fairness should, hence, be assessed on a larger scope including operational and service design decisions. Such holistic analyses that explore the trade-offs between fairness, service offers, and economic feasibility will be performed in T3.7 of SINFONICA.

As a general finding from our research, we recommend further research into the algorithmic discrimination in the context of on-demand mobility systems. While the insights and mitigation ideas provided in this report can only be a beginning, we note that this research can be pushed substantially further in terms of exploring a larger variety of dispatching algorithms, systematically analysing other contexts of discrimination (such as low-density vs. high-density areas) and looking deeper into the design of dispatching algorithms, or even notions of "inclusivity-by-design" for algorithms that are built around the goal of increasing fairness. Furthermore, as can be seen from our considerations above, there is a lack in today's literature in clear definitions of what fairness and inclusivity mean in the context of operational fleet management and service design, which should be closed in future interdisciplinary research.

7 References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 114, 462–467. https://doi.org/10.1073/pnas.1611675114

Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transp. Res. Part C Emerg. Technol. 64, 101–116. https://doi.org/10.1016/j.trc.2015.07.017

Bischoff, J., Führer, K., Maciejewski, M., 2019. Impact assessment of autonomous DRT systems. Transp. Res. Procedia, Urban Mobility – Shaping the Future Together mobil.TUM 2018 – International Scientific Conference on Mobility and Transport Conference Proceedings 41, 440–446. https://doi.org/10.1016/j.trpro.2019.09.074

Hörl, S., Chouaki, T., Ludwig, O., Rewald, H., Axer, S., 2024. Evaluating prebooked on-demand mobility services using MATSim. 13th Int. Workshop Agent-Based Mobil. Traffic Transp. Models ABMTRANS 2024.

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

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. https://doi.org/10.1109/ACCESS.2020.2990295

Lopez, P.A., Wiessner, E., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flotterod, Y.-P., Hilbrich, R., Lucken, L., Rummel, J., Wagner, P., 2018. Microscopic Traffic Simulation using SUMO, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, Maui, HI, pp. 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Maciejewski, M., Bischoff, J., Hörl, S., Nagel, K., 2017. Towards a Testbed for Dynamic Vehicle Routing Algorithms, in: Bajo, J., Vale, Z., Hallenborg, K., Rocha, A.P., Mathieu, P., Pawlewski, P., Del Val, E., Novais, P., Lopes, F., Duque Méndez, N.D., Julián, V., Holmgren, J. (Eds.), Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems, Communications in Computer and Information Science. Springer International Publishing, Cham, pp. 69–79. https://doi.org/10.1007/978-3-319-60285-1_6

Molenbruch, Y., Braekers, K., Caris, A., 2017. Typology and literature review for dial-a-ride problems. Ann. Oper. Res. 259, 295–325. https://doi.org/10.1007/s10479-017-2525-0

Mourad, A., Puchinger, J., Chu, C., 2019. A survey of models and algorithms for optimizing shared mobility. Transp. Res. Part B Methodol. 123, 323–346. https://doi.org/10.1016/j.trb.2019.02.003

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. Part C Emerg. Technol. 111, 255–293. https://doi.org/10.1016/j.trc.2019.12.008

Ruch, C., Horl, S., Frazzoli, E., 2018. AMoDeus, a Simulation-Based Testbed for Autonomous Mobility-on-Demand Systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, Maui, HI, pp. 3639–3644. https://doi.org/10.1109/ITSC.2018.8569961

For more information

SINFONICA Project Coordinator

UNIMORE – University of Modena and Reggio Emilia

Via Giovanni Amendola, 2

42122 Reggio Emilia, IT

sinfonica@sinfonica.eu

www.sinfonica.eu

Funded by
the European Union

SINFONICA Project has received funding under the Horizon Europe Research and Innovation Program (Grant Agreement n° 101064988).