Dynamic Fleet Management for Autonomous Vehicles

Learning- and optimization-based strategies

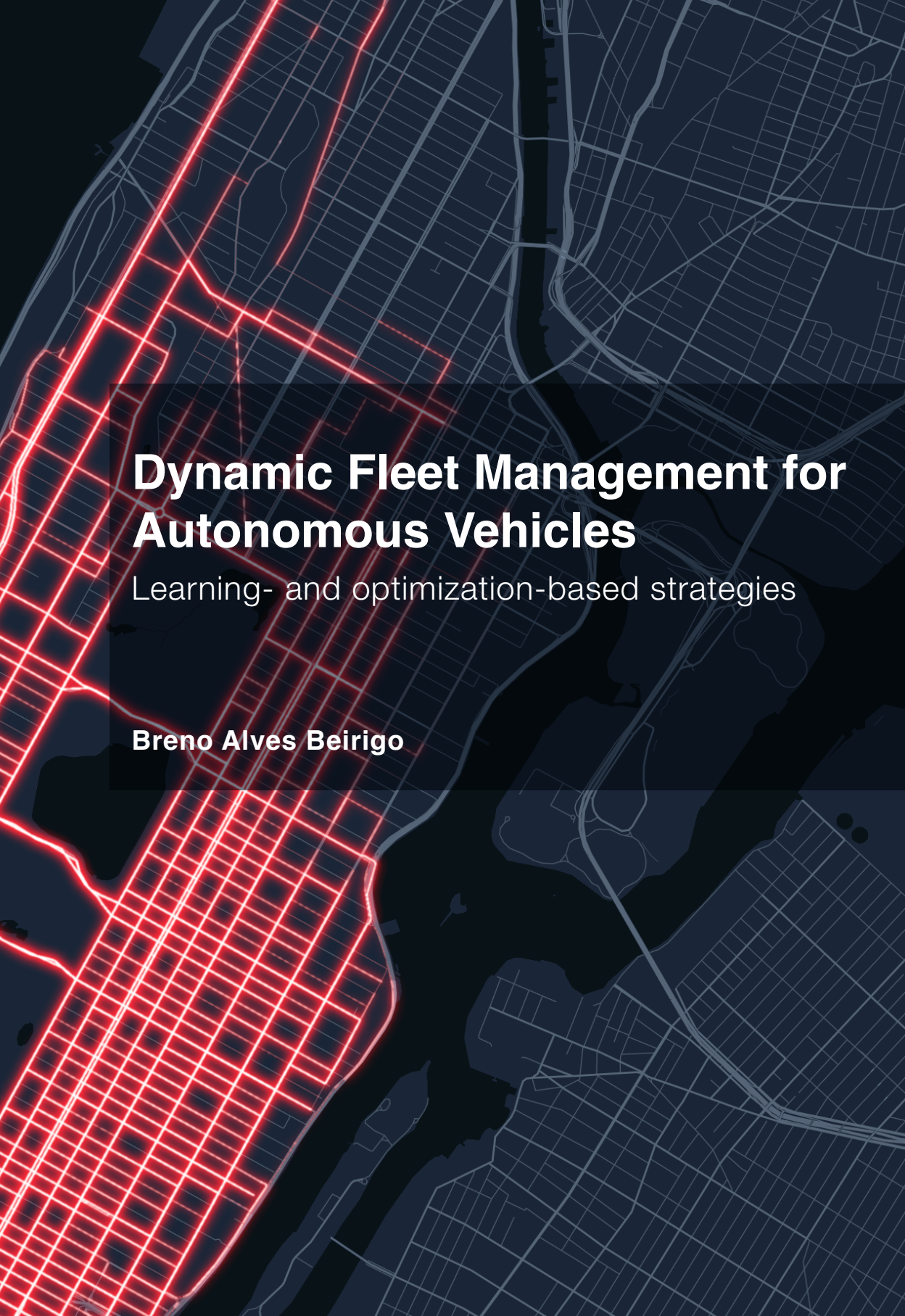
Beirigo, Breno A.

DOI
10.4233/uuid:a6b591ed-e102-4ba4-8ebc-5f1dee7833ff

Publication date
2021

Document Version
Final published version

Citation (APA)
Beirigo, B. A. (2021). *Dynamic Fleet Management for Autonomous Vehicles: Learning- and optimization-based strategies*. [Dissertation (TU Delft), Delft University of Technology]. Delft University of Technology. https://doi.org/10.4233/uuid:a6b591ed-e102-4ba4-8ebc-5f1dee7833ff



Dynamic Fleet Management for Autonomous Vehicles

Learning- and optimization-based strategies

Breno Alves Beirigo

Dynamic Fleet Management for Autonomous Vehicles

Learning- and optimization-based strategies

B. A. Beirigo

Dynamic Fleet Management for Autonomous Vehicles

Learning- and optimization-based strategies

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus Prof. dr. ir. T. H. J. J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
Tuesday 9 March 2021 at 12:30 o'clock

by

Breno ALVES BEIRIGO
Master of Science in Computer Science, Federal University of Viçosa, Brazil
born in Divinópolis, Brazil

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. R. R. Negenborn	Delft University of Technology, promotor
Dr. F. Schulte	Delft University of Technology, copromotor

Independent members:

Prof. dr. M. Gansterer	University of Klagenfurt, Austria
Dr. J. de Armas	Pompeu Fabra University, Spain
Prof. dr. H. Nijmeijer	Eindhoven University of Technology
Prof. dr. ir. B. van Arem	Delft University of Technology
Prof. dr. ir. A. Verbraeck	Delft University of Technology

This thesis is supported by the project "Dynamic Fleet Management (P14-18 –project 3)" (project 14894) of the research programme i-CAVE, partly financed by the Netherlands Organization for Scientific Research (NWO), domain Applied and Engineering Sciences (TTW).

TRAIL Thesis Series T2021/12, the Netherlands TRAIL Research School

Published and distributed by: B. A. Beirigo
Cover design by: B. A. Beirigo (map: ©Mapbox, ©OpenStreetMap)
E-mail: brenobeirigo@gmail.com

ISBN 978-90-5584-286-5

Keywords: Dynamic fleet management, autonomous vehicles, service quality, on-demand hiring, approximate dynamic programming.

Printed in the Netherlands

“Pathemata mathemata.”
Greek proverb

Preface

Kaizen (改善) is a Chinese word meaning “change for the better” that later became a Japanese philosophy championing continuous (small) improvements. Imbued by a kaizen mindset (and a generous amount of oriental patience), one can always finish the day with a sense of accomplishment in having progressed a bit in professional and personal life. I want to take this opportunity to express my deepest gratitude to all the fantastic people that have provided so many kaizen moments throughout the last four years. This thesis is a compound effect of your support, insights, and advice.

To begin with, my great appreciation goes to my promotors, Rudy Negenborn and Frederik Schulte. Dear Rudy, you were the catalyst of my Ph.D. journey, and I cannot thank you enough for that. Under your guidance, I have increasingly become more practical, organized, and driven. Dear Frederik, thank you for always encouraging me and showing me the value of being boldly optimistic. Our lively discussions were essential to hone the ideas put forward in this thesis and further my research.

On helping to shape my ideas, thanks should also go to Hans Hopman, Matthijs Spaan, and Javier Alonso-Mora, for their valuable contributions. Also, I am extremely grateful to the user committee of the “Dynamic Fleet Management” module of the i-CAVE research project for their helpful input during our meetings. My gratitude also extends to the other project participants, who helped me acquire a broader view of vehicle automation research. Furthermore, I acknowledge the efforts of my doctoral committee for their constructive remarks on my research.

I cannot underestimate the value of all insights (both on life and research) I have had during informal conversations in my time at TU Delft. In this regard, I owe many thanks to Bilge Atasoy, Bijan Ranjbar, Eduardo Souza, Felipe Moraes, Javad Mohajeri, Johan Los, Jie Cai, Linying Chen, Marc Fransen, Rie Larsen, Xiao Lin, and Wenjing Guo. Special thanks go to my loyal office mates, first Kanu Jain and Guangming Chen, next Qinqin Zeng and Chris Benson, and more recently Pang Fang.

In addition, I would like to recognize the assistance that I received from the MTT and TRAIL staff, which was essential to complete this project smoothly. Finally, I want to thank my family, especially my mother, for her unwavering support, and my lovely wife, for her warmth and unbound patience during the writing of this thesis.

Breno A. Beirigo,
Delft, March 2021

Contents

Chapter 1

Introduction

Current mobility services cannot compete on equal terms with self-owned mobility products concerning service quality. Due to supply and demand imbalances, ridesharing users invariably experience delays, price surges, and rejections. Traditional approaches often fail to respond to demand fluctuations adequately since service levels are, to some extent, bounded by fleet size. With the emergence of *autonomous vehicles* (AVs), however, the characteristics of mobility services change, and new opportunities to overcome the prevailing limitations arise.

This thesis proposes a series of strategies to help autonomous transportation providers meet the service quality expectations of diversified user bases. We show how *autonomous mobility-on-demand* (AMoD) systems can develop to revolutionize urban transportation, improving reliability, efficiency, and accessibility. First, in Section 1.1, we introduce the research on the dynamic fleet management of AVs and present the autonomous mobility scenarios and stakeholders considered in this thesis. Then, in Section 1.2, we describe a series of research challenges not yet addressed in the related literature. Next, Section 1.3 presents our research questions and Section 1.4 defines our research approach. Finally, in Section 1.5, we conclude with an overview of this thesis.

1.1 Background

The following sections introduce the concepts and background research on fleet management of autonomous vehicles. First, in Section 1.1.1, we describe the main implications of AV implementation. Next, in Section 1.1.2, we discuss the possible impacts of a shared approach to urban mobility. Then, in Section 1.1.3, we present the existing concepts on dynamic fleet management, highlighting the close relation with the classic *vehicle routing problem* (VRP), and identifying the particularities of AV's implementation. Finally, Section 1.1.4 further details the autonomous mobility scenarios considered in this thesis, and Section 1.1.5 identifies the requirements and goals of AMoD systems' stakeholders.

1.1.1 Autonomous vehicles

The advent of autonomous vehicles represents a disruptive change to transportation systems with uncertain repercussions. Full vehicle automation, that is, the ability to deal with any roadway and environmental conditions [85], is currently the most anticipated vehicle technology: major automotive and technology companies (e.g., Ford, Mercedes-Benz, Tesla, Google, and Uber) have all declared that fully autonomous vehicles will rollout on US roads in the next decade [43]. Automation is seemingly a natural step forward on today's available vehicles. A culmination of technologies such as geolocalization, self-parking, *adaptive cruise control* (ACC), surrounding alerts and stop-and-go self-steering [99].

In the following, we summarize the major impacts widespread AV adoption will potentially cause in different areas:

- *Mobility:* AVs could facilitate personal independence and mobility, attending non-drivers, such as the teenagers, the elderly, and the disabled [33].
- *Parking:* AVs can self-park in less expensive areas and even communicate with parking infrastructures beforehand. This feature avoids cruising for parking, a practice that usually incurs additional fuel expenses and unwanted delays [33].
- *Car-ownership:* AVs may reduce car ownership, especially in densely-populated areas, allowing travelers to rely on shared autonomous vehicles [59]. Conversely, some motorists may prefer to keep their vehicles for convenience or privacy's sake.
- *Trip making:* Increased convenience, affordability, and people's willingness to travel may induce additional vehicle travel, increasing external costs such as parking, congestion, and pollution [59, 99].
- *Time expenditure:* Commuters may enjoy their in-vehicle time differently, either relaxing or being productive. Therefore, personal *value of travel times* (VOTTs) are likely to decrease [33].
- *Infrastructure:* AVs could move in coordination with intelligent infrastructure, allowing quicker reaction times and closer spacing between vehicles to counteract increased demand [96].
- *Land use patterns:* Public spaces may be redesigned to reflect a mobility future with decreased parking needs, and possibly fewer vehicles, due to shared on-demand transportation [99]. Additionally, stimulated by AVs' convenience, customers may find it advantageous to purchase cheaper houses in exurban locations [59].
- *Traffic:* AVs are expected to efficiently use existing lanes and intersections, keep shorter gaps between vehicles, and select more efficient route choices, ultimately leading to reduced congestion [33]. However, these benefits only add up to become a useful congestion-relieving tool if a significant share of the fleet is autonomous. The success of shared on-demand transportation services may also decrease traffic, as they become a convenient alternative to the ownership model.
- *Public transit:* in case AVs stimulate more sprawled land use development patterns, public transit travel demands may be reduced. Overall, customers may find it more convenient to rely on door-to-door on-demand services [59].

Nevertheless, the precise time AVs will become completely reliable, affordable, and broadly adopted is still uncertain. Many side aspects still need to be addressed before AVs successfully debut, among them:

- *Technical reliability:* AVs will probably require more years of development and testing before safety is proven under all possible conditions [59].
- *Consumer adoption:* Besides strict safety requirements, varying degrees of consumer trust and purchasing power may result in a slow fleet turnover, creating a mixed-traffic environment with roadway management problems [59].
- *Regulation:* Since legal and supporting infrastructures are generally unable to evolve as fast as technology development, AVs are likely to be unevenly deployed. Ultimately, during the implementation phase, authorities are expected to establish performance standards to guarantee that AVs can safely operate on public roads [33].
- *Cost:* AVs may require additional sensors (e.g., optical, infrared, radar, ultrasonic, laser), services (e.g., navigation, computation), maintenance (for cleaning or repairing critical components), and testing [59]. These elements add up to the final price, making AVs unaffordable for potential buyers and ultimately delaying mass adoption.

1.1.2 Shared autonomous vehicles

The world level of urbanization is likely to increase steadily in the next decades [109]. This growth tends to be accompanied by a series of underlying repercussions. While urban land will increasingly become scarce, the demand for city services and infrastructure will probably also rise. As a result, current urban mobility deficiencies, such as lack of parking spaces, congestion, and low vehicle occupation rates, may rise if the current mobility paradigm remains unaltered [79]. One of the leading strategies to cope with these unwanted repercussions is adopting shared, *mobility-on-demand* (MoD) services. These services may significantly reduce road congestion and emissions once they lie in between public and private transportation modes, being shared and flexible at the same time [100]. However, Tachet et al. [100] show that most world urban centers have a high unexplored "shareability" rate. The majority of their current single-passenger rides could seamlessly be combined, even for low trip density scenarios and considering no longer than five-minute delays.

Previously, the technological revolution that led to smartphone mass-adoption enabled the development of today's MoD services. Perhaps, the missing component to enable widespread vehicle sharing and ultimately make cities keep pace with a growing transportation demand lies in another technological advance, namely, autonomous vehicles [98]. Recent studies have shown that an *autonomous mobility-on-demand* (AMoD) system employing a fleet of *shared autonomous vehicles* (SAVs) could significantly improve mobility in urban environments. Despite the uncertainties regarding autonomous vehicles implementation, most SAV implementation simulations (e.g., [2, 12, 34]) indicate that replacing current transportation modes may be extremely beneficial in urban environments, decreasing the active number of running vehicles while keeping a short

response time. Smart fleet routing may even avoid congestion, as opposed to the claim that AV relocation protocols would worsen traffic conditions due to additional trips of empty vehicles [84].

1.1.3 Dynamic fleet management

Fleet management is an activity that encompasses the fields of transportation, distribution, and logistics. Its primordial objective consists of improving operational efficiency and service quality while minimizing overall costs [11]. As pointed out by Zeimpekis et al. [119], at the core of fleet management and supply-chain coordination lies the VRP, a highly challenging optimization combinatorial optimization problem first proposed in [29]. In broad terms, it deals with the optimal assignment of transportation orders to a fleet of vehicles and the sequencing of stops for each vehicle representing the formation of routes.

Due to the remarkable amount of logistical contexts a fleet may operate, many VRP variants have been studied in the literature. Next, we present the VRP variants that best capture the characteristics of the transportation system considered in this thesis:

- *Dynamic VRP (DVRP):* refers to environments in which events, such as customer requests, traffic congestion, order update, and delays, are dynamically revealed to the decision maker over time [119].
- *Dynamic stochastic VRP (DSVRP):* When it comes to defining the way dynamic information evolves during the optimization process, DVRPs can be separated into two categories: *dynamic and deterministic* and *dynamic and stochastic*. While the former deals with an unknown input, the latter assumes that exploitable stochastic knowledge is available on the dynamically revealed information [80]. The main phenomena considered in the literature to create such knowledge are related to (i) demands (space-time likelihood of occurrence), (ii) customers (present or absent), and (iii) times (service or traveling) [38]. Hence, predictive analytics, that is, the processing of the system's historical data to create stochastic distribution models, can ultimately improve the system's future states [51].
- *VRP with pickups and deliveries (VRPPD):* transportation requests consist of point-to-point transports, that is, movements of people or cargo between *origins and destinations* (ODs) [10, 104]. Depending on the way vehicles move between points, such problems may be classified as (i) many-to-many, (iii) one-to-many-to-one, and (ii) one-to-one [10]. In (i), any point can serve as a source or destination for any commodity. In (ii), commodities may be transported from the depot to the customers and vice-versa. Finally, in (iii), each commodity has a given origin and a given destination (e.g., door-to-door transportation systems).
- *VRP with time windows (VRPTW):* Requests are expected to be serviced within predetermined *time windows* (TWs), defined by earliest and latest arrival times. When arriving within a TW is unfeasible, a provider can wait until the earliest arrival time (for static demands placed in advance), allow flexible latest arrival times, or deny service.

- *Capacitated VRP (CVRP):* Vehicles have capacity constraints (e.g., weight, volume, number of seats) that restrict unlimited loading [104].
- *Heterogeneous fleet VRP (HVRP):* Fleet may be composed of vehicles with different capacities [80].

Hence, a VRP variant able to seize all the different properties of AMoD systems can be identified as DSHCVRPPDTW, that is, dynamic stochastic heterogeneous fleet capacitated vehicle routing problem with pickups, deliveries, and time windows. Variants such as the DSHCVRPPDTW, whose purpose is to deal with real-life applications, are generally denominated as *rich vehicle routing problems* (RVRPs). The term "rich" embodies the different decision levels (strategical, tactical, and operational) considered [51].

Unlike VRPs, in which all data is known in advance, DVRP variants require that scheduling plans are adjusted online as additional information is received. As a result, optimization strategies are usually run repeatedly [51]. However, Psaraftis et al. [83] point out that most DVRP studies propose adaptations of known algorithms for the static version of the problem. The authors argue that the pressing time constraints typical in dynamic settings justify the predominance of (meta)heuristic approaches such as tabu search, neighborhood search, insertion methods, nearest neighbor, genetic algorithms, ant colony optimization, particle swarm optimization, and column generation. In contrast, few approaches are essentially dynamic and stochastic, among them: Markov decision processes, approximate dynamic programming, reinforcement learning, waiting-relocation strategies, and queueing-polling strategies.

1.1.4 Future mobility scenarios

In the following, we describe the future logistical scenarios enabled by vehicle automation that we intend to address in this thesis. The predominance of one scenario over another may be greatly influenced by particular characteristics of each area, such as geography, traffic conditions, and the existence of other transit modes. For example, in densely populated urban centers, AMoD services may reach a break-even point sooner, representing a cost-efficient alternative to owning a vehicle. In contrast, suburban inhabitants may find it more convenient to commute in their own vehicles. Sharing and ownership scenarios, however, are not exclusive. As pointed out by Campbell [14], incentives to simultaneously own and share AVs could be higher than solely relying on ridesharing services, such that future AV fleets could ultimately be a mosaic of privately-owned vehicles working as freelancers at idle times.

The triumph of ridesharing

Litman [59] estimates that AVs are likely to become affordable by 2040, which may further expand taxi and carsharing services for low-income passengers. McKinsey & Company and Bloomberg New Energy Finance [65] agree that the costs of commuting via shared, self-driving vehicles may eventually become as affordable as public transit modes. Consequently, stimulated by the additional convenience of a door-to-door on-demand service, many passengers may be compelled to subscribe to an AMoD provider, choosing SAVs as their main transportation mode or as first- and last-mile options. Stone

et al. [99] suggests that public transportation could even change its form: from shared and large vehicles to personal and small capacity vehicles designed to attend one individual transportation demand at a time. Sharing services reduce personal mobility costs once all expenses of purchasing, maintaining, and insuring vehicles are distributed across a large user base [98]. Litman [59] points out that once benefits are proven, many households would likely be keen on relying on such services, reducing their vehicle ownership, and as a result, cities' congestion and parking requirements.

The predominance of vehicle ownership

Although autonomous carsharing, ride-hailing, and ridesharing programs may attend a remarkable share of future transportation demands, the added convenience brought by automation may maintain consumers' appetite for private vehicles [65]. Besides privacy concerns and independence, some motorists may prefer a personal vehicle rather than a shared one if they drive high annual miles, frequently carry personal belongings, or simply want to portray status [59]. Moreover, people may reject taking a trip with strangers due to personal security concerns or dissatisfaction with ride partners' features (e.g., gender, age) [55]. Hence, most of the current commuting patterns may remain essentially unchanged. A household AV, for example, may become the family's assistant. It could move goods back and forth, pick up orders from stores, and be summoned by other family members on-demand. Parking would also be facilitated since an AV can drop off passengers at their destinations and park elsewhere (possibly a cheaper location) to wait for the next demand [54]. Finally, as Krueger et al. [50] points out, the VOTT of former drivers would likely decrease. They would be freed for purposes other than driving, increasing in-vehicle productivity, and allowing people to live in farther but affordable areas.

1.1.5 Mobility stakeholders

The main stakeholders involved in a transportation system are presented in the following sections. Each stakeholder has different requirements and often conflicting goals.

Users

A user aims to transport something or someone (possibly him- or herself) from one place to another. This transportation demand is ultimately translated into a request that is expected to be timely and efficiently addressed by an adequate vehicle. In a highly connected environment powered by widespread sensor technologies, users' freight transportation demands can be equipped with specific preferences. For instance, perishable goods are time sensitive, requiring a series of accommodating conditions to preserve their quality (e.g., temperature, humidity). In turn, fragile goods or livestock cargo are impact sensitive and may require smoother maneuvers. Hence, such preferences have to be considered by a transportation system when scheduling and routing vehicles.

Independent autonomous vehicle owners

Private AV owners differ from users because they are also concerned about the costs associated with acquiring and maintaining their assets. Therefore, besides minimizing transport travel time, a vehicle owner also seeks to minimize fuel, parking fees, tolls, and deterioration. Additionally, owners also expect that most of these parameters are automatically considered by the AV navigation system when attending to household demands. Owners may also be keen to make money out of their vehicles' idle time, occasionally leasing them to AMoD systems.

Autonomous mobility-on-demand providers

The concerns related to owning a single AV are scaled up for AMoD providers owning multiple vehicles. Instead of optimizing their single trajectories, vehicles are expected to work in harmony to fulfill users' transportation demands and owners' underlying goals. In turn, such goals are related to the general profile of a fleet owner as a private or public operator. These profiles are essentially distinguished by how much weight is placed either on transportation equity or system profitability. For instance, when profit is the most prominent driving force, equity concerns may occasionally be sidelined. As shown in [17], some very profitable strategies perpetually ignore upcoming demands that are less lucrative.

Cities

Cities, or, more precisely, city managers and transportation authorities, are mainly concerned with the repercussions of the other stakeholder goals in the urban environment. Hence, traffic management strategies, infrastructural deployment, and restrictive/supportive policies may become necessary to mitigate congestion and improve accessibility.

1.2 Research challenges

In the following sections, we present and motivate the research questions arising from specific logistical challenges regarding the context of dynamic fleet management of shared autonomous vehicles.

1.2.1 Transitioning to full automation

For the most part, studies on autonomous vehicles investigate solutions in a full-automation setting. However, this mobility landscape is far from reality when we consider the current status of the autonomous systems currently being tested in the market. Figure 1.1 shows the well-known *Society of Automotive Engineers* (SAE) taxonomy to distinguish between levels of autonomy. As pointed out by Litman [59], companies have still been testing SAE level 3 vehicles in which special conditions apply (e.g., mapped routes, fair weather, possible human intervention). Additionally, although some manufacturers have claimed level 4 vehicles will be available in the near future, early versions are likely to be limited to more controlled environments (e.g., freeways, "geofenced" areas), probably requiring human intervention on surface streets.

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 1.1: *Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (SAE International standard) [85].*

Moreover, regulatory barriers can restrict operations only within certain zones where safety is guaranteed. Chen et al. [19], for instance, advocates that government agencies can dedicate certain areas of road networks exclusively to AVs. Such *autonomous vehicle zones* (AVZs) would be capable of enhancing the transportation network performance by, for example, facilitating the formation of platoons. Besides, inside zones or selected streets, AVs could facilitate parcel delivery by automatically handing over goods to pickup stations [108]. To improve service coverage in a particular area, providers could also combine a heterogeneous fleet of autonomous- and manually-driven vehicles. For instance, Scherr et al. [90, 91, 92] consider a logistic scenario where manually operated vehicles guide AV platoons outside AVZs to move parcels from distribution centers to transshipment points within a city. As some of these points are inside AVZs, inter-zone crossing requires that the provider coordinate the mixed fleet to carry out platoon operations from time to time.

However, to the best of our knowledge, the problem of servicing demands in heterogeneous autonomous and non-autonomous networks has not yet been addressed in a passenger transportation setting. In this setting, the extra delay caused by platoon operations, for example, can be unacceptably high for on-demand services. Hence, to operate within such mixed-zone scenarios successfully, alternative fleet compositions may be required.

1.2.2 Heterogeneous service quality requirements

Considering service levels that mirror the personal VOTT of user segments may leverage AMoD systems' overall performance, especially in terms of occupancy levels and profitability. Not accidentally, the most common metrics applied to assess service levels from the customers' perspective are concerned with time, namely, (i) the pickup delay and (ii) the in-vehicle delay due to ridesharing. Although minimizing (i) and (ii) improves user experience, shared transportation systems generally benefit from having more flexible time constraints. For instance, Fagnant and Kockelman [34] showed that increasing the maximum travel delay led to reduced empty-SAV trips while also shrinking the fleet size. Santi et al. [86] also found that increasing the maximum waiting time increased the number of shared, non-vacant trips from around 30% to almost 100%. In summary, increasing delays can improve vehicle shareability (a desirable property for cities and operators).

Hence, a private fleet owner aiming to cater to a broader market section may want to offer extra mobility capabilities to serve more diversified transportation requests. For instance, some users may prefer to pay higher fares for faster service or private rides. Alternatively, other users would not mind waiting for more affordable shared rides. Hence, in a more flexible AMoD system, users could be economically incentivized to join rides that increase sharing and occupancy. Current on-demand transportation companies are already attempting to explore different demand segments by offering various riding options, from car-pooling to luxury rides.

Few approaches (e.g., [18, 60, 123]) examine the outcome of fulfilling the needs of diverse user profiles on AV fleet management strategies. However, none offers mechanisms to prioritize user segments (e.g., business users) or is capable of fully meeting user expectations.

1.2.3 Short-term fleet size elasticity

Finding the minimum fleet size capable of sufficiently meeting the transportation demand is an overarching discipline in the AMoD literature. Commonly, authors rely on performance-driven fleet sizing approaches that aim to ensure that the fleet can meet the service quality requirements of a particular transportation demand [98].

Due to typical passenger demand fluctuation, however, this method faces two significant limitations. First, once fleet-sizing simulations generally run on whole-day travel data, the final number of vehicles ultimately reflects daily demand peaks. Since these peaks typically occur only twice a day, most vehicles remain underutilized most of the time. Second, in the face of unexpected/unusual demand spikes (e.g., due to bad weather or crowded concerts), there are not enough vehicles to adequately fulfill all requests. Therefore, entirely relying on fixed fleets may be detrimental for future AMoD providers, harming both fleet productivity and reliability.

Today's on-demand transportation companies are already adopting more flexible strategies. For instance, Uber's current business model benefits from being solely a matching platform for drivers and riders. Since drivers are responsible for owning and maintaining their vehicles, the company does not need to buy and maintain a private fleet. This characteristic was crucial to their business expansion once starting opera-

tions in different cities does not require considerable outlay. To actively control service quality, the company applies dynamic pricing strategies to incentivize drivers to move to undersupplied locations to improve their profits [107].

In an AV-dominant future, providers can still continue to outsource demand to privately owned operators. Some owners may profit from leasing their vehicles to AMoD systems occasionally (e.g., during idle periods of the day) or continuously, as independent contractors [14]. Therefore, apart from the expected scheduling and routing services, a future on-demand transportation company may become a mediator between car-owners and riders, both with particular and heterogeneous constraints. Additionally, car-renting companies and car automakers, which will supposedly own large AV fleets, could also provide subscription plans to private users or associate themselves with third-party on-demand transportation services. Customers would also gain further freedom of choice. There would be many more options to choose from instead of a uniform fleet of vehicles besides other renting methods (e.g., booking a vehicle for an entire day or week). Moreover, different car owners could stipulate different profit margins or minimum compensations. Therefore, the best possible ride for a transportation demand would consider both customers' requirements and owners' constraints. On the other hand, car owners could promote their vehicles in the scoring system by guaranteeing a regular maintenance/cleaning routine.

Nevertheless, as pointed out in [47, 72], research on autonomous transportation systems considering such a short-term fleet size elasticity is still lacking in the literature. Although optimization approaches can generally deal with vehicle surpluses over a given planning horizon, active fleet size management on the operational level (as a means to guarantee service quality) has not yet been investigated.

1.2.4 Dynamic stochastic supply and demand

On-demand transportation systems can further improve their efficiency and profitability if they can adjust imbalances between supply and demand [114]. A fleet management system might correct these imbalances by applying relocation strategies, moving vehicles from low-demand areas to high-demand ones when applicable. Likewise, the initial fleet distribution may also have a strong influence on user service levels if vehicles can start close to upcoming demands [12]. However, although demand-anticipatory relocations are critical to lower customer's waiting time, the strategy must be well balanced to avoid excessive *vehicle kilometers traveled* (VKT) of empty-SAVs [34]. Ideally, relocation costs should compensate for the additional earnings, unless the loss is part of a deliberate strategy to outperform competitors to gain visibility among customers [114].

Reactive approaches use ongoing imbalances to reposition vehicles between areas. For instance, Chen et al. [18] use a price-based strategy that encourages trips originating in a cell with a surplus of vehicles and penalizes trips originating in a cell with a deficit of vehicles. In comparison, Fagnant and Kockelman [34] push or pull unoccupied SAVs to or from adjacent geographical blocks, prioritizing shifts to blocks exhibiting complementary imbalances.

Conversely, stochastic approaches use historical data to anticipate future events. Due to the popularization of *information and communication technologies* (ICTs), obtaining knowledge about stochastic phenomena has become facilitated in the field of transporta-

tion [38]. Companies owning historical information can apply predictive analytics techniques (e.g., machine learning and data mining) to assess the likelihood of future events. Uber, for instance, uses historical data on traffic conditions to establish an *estimated time of arrival* (ETA) to waiting customers [73]. Hence, planning can be improved by using data from all entities involved in the transportation system to determine the best possible decisions for a predicted behavior.

Studies leveraging stochastic demand information to rebalance vehicles abound in the literature. Authors typically employ methods such as reinforcement learning (e.g., [41, 58, 115]), model predictive control (e.g., [48, 106, 121]), and approximate dynamic programming (e.g., [1], [94]), to estimate the value of future system states. However, most related literature does not account for in-demand stochastic patterns originating from heterogeneous user base composition. Consequently, vehicles ultimately rebalance to places where the number of upcoming homogeneous requests is expected to be the highest. Notwithstanding, when the patterns of a diversified user base are considered (see Section 1.2.2), novel rebalancing strategies may arise.

Moreover, research on the benefits of incorporating the stochastic information surrounding the supply of privately-owned AV is still lacking in the literature. Such a gap stems directly from the challenge presented in Section 1.2.3, on the elasticity of AV fleets. By harnessing the knowledge on both demand and supply patterns, AMoD systems are substantially better equipped to meet user needs with the added advantage of not necessarily owning large AV fleets.

1.2.5 Equitable access to autonomous mobility

Contrary to the claim that autonomous vehicles will substantially improve accessibility (e.g., [66]), there is still much concern about the actual impact of automated driving on transportation equity (e.g., [30]). First, the potential necessity of deploying special infrastructure (as hypothesized in Section 1.2.1) can lead to increased inequality. As pointed out by Cohn et al. [22], if AVs have barriers to entry due to initial unaffordability, investing in AV infrastructure could result in transportation resources being inequitably distributed. Creger et al. [25] agrees that if AVs' deployment is not principled in equity, they will likely exacerbate transportation injustices. Likewise, Dean et al. [30] suggests that if the private vehicle ownership scenario prevails (see 1.1.4), AMoD services could further perpetuate existing societal inequalities, primarily benefiting high-income populations.

Typically, classic performance measures and evaluation criteria used in transportation modeling fail to account for differences in demographics[22]. For example, most vehicle repositioning strategies in the literature rely on supply-demand imbalances or historical profitability as stimuli to distribute idle vehicles. As a result, a rebalancing bias develops towards densely-populated areas, systematically disadvantaging passengers in low-demand zones [116]. Besides, provided that AV fleets are primarily operated by private companies, low-income areas can be further neglected in favor of affluent areas, where profits are more likely to occur. In light of these problems, there is a growing consensus on the need for public transportation authorities to ensure future mobility services converge towards maximizing benefits for all city residents [14, 20, 20, 21, 25, 76]. Nevertheless, equity goals are mostly overlooked in AV fleet management research.

1.2.6 Short-haul integration of passenger and freight flows

Although some long-haul modes (e.g., aircrafts, ferries) already integrate passenger and freight flows, short-haul integration is hardly observed in practice [89]. Daduna [26] points out that current attempts to integrate freight transport into urban public transportation have failed because of two main reasons. First, the integration inevitably leads to further loading/unloading operations, possibly incurring downstream delays that may cause the loss of connections at transfers. Second, to operate a given timetable at off-peak hours, new vehicles and drivers may be necessary, leading to additional costs.

Moreover, as pointed out by Mourad et al. [69], few studies consider the joint use of transport resources between passengers and goods flows. Among them, fewer model the problem of servicing the two demand types interchangeably and simultaneously. To the best of our knowledge, such a transportation setting is explored only by Li et al. [55, 56, 57] and Do et al. [31]. The authors consider a ride-hailing setting where taxis can pick up parcels while servicing customers, but limit the ridesharing capabilities such that multiple customers cannot share a ride.

People-and-freight integration, however, can be further facilitated by future autonomous technologies. As technology develops, vehicle bodies can be adapted to carry out a range of logistics operations. AVs equipped with parcel lockers are expected to dominate parcel delivery in urban areas [65]. Ford automaker, for instance, envisages an "autolivery" future in which self-driving vans could be used to quickly transport goods within a city, potentially partnered with drones to realize the final leg of a given journey [71]. In turn, Toyota presents an AV concept where bodies can be customized following user specifications (e.g., hotel room, retail shopping, ridesharing) [105].

Hence, given the expected versatility of future body types, providers could ultimately adopt mixed-purpose SAVs, which can consolidate multiple people and freight transportation requests simultaneously. The extent to which the performance gains brought by such an autonomous *people and freight integrated transportation* (PFIT) system justify its adoption has not been studied in the literature yet.

1.3 Research questions

Assuming the future mobility scenarios described in Section 1.1.4, the role of the mobility stakeholders described in Section 1.1.5, and the research challenges presented in Section 1.2, we present the main research question to be addressed in this thesis:

RQ: How can AMoD systems leverage supply and demand information as well as cities' infrastructure to balance the goals of all mobility stakeholders?

To handle the research challenges encompassed by this question thoroughly, we propose the following key research sub-questions (SQs) and subsequently identify the challenges they address:

SQ1: How can fleet operators deal with the operational restrictions arising in the early stages of AV deployment? (1.2.1)

SQ2: How can AMoD systems guarantee service quality, in terms of responsiveness, reliability, and privacy, while improving fleet productivity? (1.2.2, 1.2.3)

SQ3: How can AMoD systems explore the stochastic information surrounding privately-owned vehicle supply and heterogeneous demand? (1.2.4)

SQ4: How can cities steer providers towards achieving equity goals? (1.2.5)

SQ5: How can AMoD systems handle passenger and cargo demands interchangeably to improve fleet productivity? (1.2.6)

1.4 Research approach

In order to answer the research questions presented in Section 1.3, this thesis presents a series of optimization methods to manage fleets of autonomous vehicles in diverse operational environments. In Figure 1.2, we show all the components that integrate a *fleet management system* (FMS). First, a solving strategy processes all available data (e.g., travel times, demand, vehicle statuses) to improve the objective functions entailed by a particular logistical scenario, which encompasses the characteristics of a transportation setting. Next, event cases, that is, entities that register the current system's information, are continuously recorded in a knowledge base. This module comprises facts about the world (i.e., historical data) and an inference engine that can reason about these facts (i.e., data analytics methods) [44]. Finally, the knowledge built or updated from the event cases can be exploited by the solver during the optimization process through a predictive model [119].

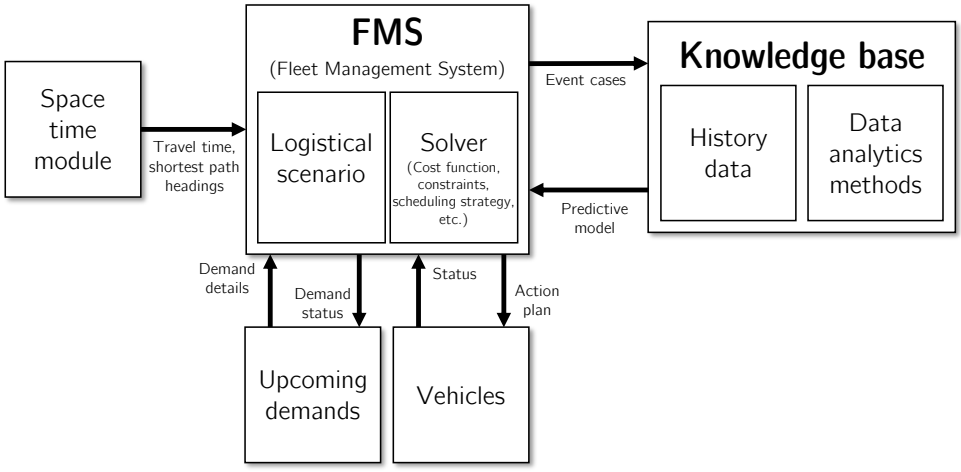


Figure 1.2: *Detailed structure of a fleet management system based on [119].*

We propose *mixed integer linear programming* (MILP) models for each logistical scenario and solve them to optimality. To deal with real-world instances of one of the investigated scenarios, we also design a matheuristic. In turn, to build the knowledge base and enable anticipatory optimization, we propose an *approximate dynamic programming* (ADP) algorithm.

Regarding the data feeding the FMS system, upcoming demands are collected from third-party sources (e.g., taxicab data) or generated. In turn, the space-time module considers real-world, high-granularity street networks from areas such as Manhattan, New York City (US), Delft (NL), and Rotterdam (NL). We simulate the movements of the AV fleet throughout these areas, updating vehicle action plans frequently. The only source of uncertainty associated with vehicles is the availability of the third-party fleet (i.e., location, announcement time, service time window). Other disturbances, such as vehicle breakdown and travel time delays, are not considered.

1.5 Thesis outline

The chapter's order aims to reflect the increasingly complex logistic operations required at different AV technology development levels. As AVs evolve and become widespread, providers can exploit different alternatives to meet user service quality requirements. Figure 1.3 provides an overview of the structure of this thesis.

Initially, a precondition to providing service quality in autonomous transportation is safety. To guarantee safe operations, the transition to a full automation setting can take several decades. Infrastructure will gradually evolve to accommodate AV movements, whereas AVs will improve their fitness to deal with complex traffic. *Chapter 2* presents a MILP model for this initial phase of AV deployment, where mobility services will have to deal with AV-ready and not AV-ready areas. Subsequently, we focus on AV-only scenarios, where AVs can drive everywhere.

In *Chapter 3*, we model a rich transportation scenario comprised of heterogeneous users and vehicles. Analogously to other transportation modes, we consider that the system must deal with a diversified user base with different service quality expectations. We propose a multi-objective matheuristic to dynamically hire third-party AVs whenever company vehicles are incapable of sustaining the service level requirements purchased by users. To guarantee high fleet productivity, we design a reactive rebalancing algorithm that uses user's service level violations as stimuli to rebalance vehicles to low demand areas.

Using the same elements of the scenario proposed in *Chapter 3* (i.e., heterogeneous service requirements and vehicle hiring), in *Chapter 4*, we exploit both user and third-party vehicle information to enable anticipatory decision making. We propose an ADP algorithm that builds value function approximations of future system states iteratively. We use these values in our objective function to weigh the downstream impact of the system's decisions (dispatch, rebalance, or hire vehicles) in the future. Additionally, we assess the influence of penalization schemes in reducing user service-level violations.

In *Chapter 5*, we approach service quality from a different perspective. We extend the method presented in *Chapter 4* to investigate how to overcome the demand patterns naturally incorporated into value functions to improve the service levels of targeted city areas. This way, providers can implement more equity-aware rebalancing strategies, allowing that all city residents enjoy the benefits of autonomous mobility.

Next, in *Chapter 6*, we present a MILP model for a highly flexible environment, where both people and parcel demands, with varying service level requirements, are combined in mixed-purpose AVs. Once parcel transportation requests are more amenable to waiting, new opportunities to optimize fleet usage arise.

Finally, in *Chapter 7*, we conclude the thesis, summarize our contributions, and present an outlook for future research.

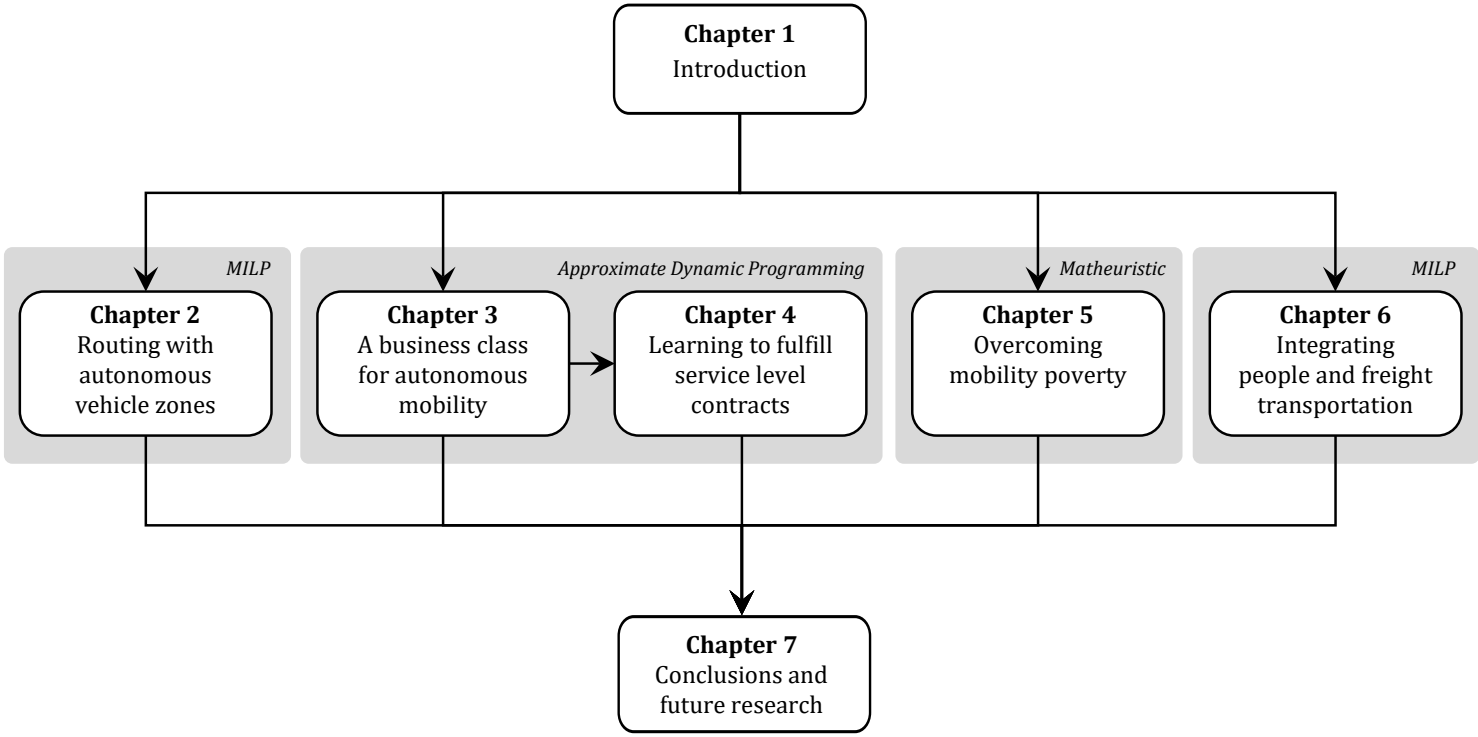


Figure 1.3: *The outline of this thesis. In the background, we indicate the solution strategies used in each chapter.*

Chapter 2

Routing with autonomous vehicle zones

Throughout Chapter 1, we presented how widespread AV adoption is expected to reshape transportation systems. This chapter shows how service providers can guarantee full coverage in a mixed autonomous and non-autonomous environment. We model detailed automated driving areas and consider a heterogeneous fleet comprised of three vehicle types: autonomous, conventional, and dual-mode. While autonomous and conventional vehicles can only operate in their designated areas, dual-mode vehicles service zone-crossing demands in which both human and autonomous driving are required. For such a hybrid network, we introduce a new mathematical planning model based on a site-dependent variant of the *heterogeneous dial-a-ride problem* (HDARP).

This chapter is organized as follows. We motivate the adoption of *autonomous vehicles zones* (AVZ) in Section 2.1, describe the problem in detail in Section 2.2, and propose a mathematical model for it in Section 2.3. Next, in Section 2.4, we present the parameters we use to design a series of AV deployment scenarios in the city of Delft, the Netherlands. Then, in Section 2.5, we evaluate how operational costs, service levels, and fleet utilization develop across scenarios, concluding with a summary of key insights and outlook on future work in Section 2.6. Parts of this chapter have been published in [7]:

B. A. Beirigo, F. Schulte, and R. R. Negenborn. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1325–1330, Maui, HI, United States, 2018.

2.1 Introduction

During a transition phase to full-automation, the introduction of AVs is likely to happen gradually, following not only technological advances but also the spread of automation-friendly infrastructures. However, most studies on SAV management assume a full-automation setting, a mobility scenario that is currently far from reality. Many companies have been still testing SAE level 3 vehicles in which special conditions apply (e.g., mapped routes, fair weather, possible human intervention), and early versions of level 4 vehicles are likely to be limited to more controlled environments (e.g., freeways, restricted zones) [59, 64]. Hence, in the early stages of vehicle automation, regulatory barriers are likely to prevent AVs from operating in areas requiring advanced driving capabilities (e.g., shared spaces). Chen et al. [19], for instance, suggest that government agencies can dedicate certain areas of road networks exclusively to AVs. Such *autonomous vehicle zones* (AVZs) could, for example, enhance the performance of transportation networks by facilitating the formation of platoons. In essence, until automation level 5 is achieved, fleet operators have to employ both conventional and autonomous vehicles to guarantee maximum service coverage on partially autonomous infrastructures.

This chapter investigates how the gradual evolution of autonomous infrastructures influences the fleet composition and vehicle routing in a mobility system. We simulate the spread of *automated driving* (AD) areas in urban networks and analyze the operational performance of a heterogeneous fleet comprised of AVs, *conventional vehicles* (CVs), and *dual-mode vehicles* (DVs). While CVs and AVs are only allowed to operate in their respective areas, DVs can freely drive throughout the entire network. We carry out the analyses for the city of Delft, the Netherlands, by creating various autonomous driving areas in the city's mobility network. Figure 2.1 illustrates a possible setting with autonomous and conventional driving zones in the example of Delft.

Figure 2.1: *Example of autonomous vehicle zone (AVZ) deployment in Delft, the Netherlands. Inside AVZs, infrastructure is ready to support automated driving.*

2.2 Problem description

In this chapter, we propose a *multi-depot site-dependent dial-a-ride problem* (MDSDDARP), an extension of the *heterogeneous dial-a-ride problem* (HDARP) introduced by Parragh [78]. Similarly to HDARP, the MDSDDARP consists of designing a cost-effective routing plan for a fleet of heterogeneous vehicles to service a series of pickup and delivery requests with different transportation modes. However, in most HDARP variants, the source of vehicle heterogeneity is associated with the transportation requirements of hospitals' patients (e.g., wheelchair space, stretcher, patient seat). In contrast, in MDSDDARP, the compatibility relationship between users and vehicles depends on the vehicle's ability to access user locations. This concept was first explored by Nag et al. [70] in the *site-dependent vehicle routing problem* (SDVRP), in which certain sites (e.g., congested areas) could only be serviced by specific types of vehicles (e.g., small-capacity vehicles). However, rather than relying on vehicle dimensions or user preferences to determine user-vehicle compatibility, we rely on vehicles' driving capabilities (automated, conventional, and dual-mode) to decide whether they are allowed to service users in automated or conventional driving areas.

We summarize the MDSDDARP as follows. Given

- a hybrid street network comprised of an AVZ and a *conventional vehicle zone* (CVZ),
- a heterogeneous fleet comprised of autonomous, conventional, and dual-mode vehicles,
- a set of time-constrained transportation requests arising from either a CVZ or an AVZ,

the MDSDDARP consists of constructing a set of vehicle routes in such a way that

- DVs can pick up and deliver users in the entire network, whereas AVs and CVs can only operate in automated and non-automated driving areas,
- vehicles depart from multiple locations and can stop at the delivery location of their last serviced user,
- the capacity of a vehicle is not exceeded along its route,
- the ride time delay of a route does not exceed a limit w_{ride},
- the pickup time delay of a request does not exceed a limit w_{pickup},
- a subset of the requests is serviced (i.e., service denial is allowed),
- the total profit is maximized.

Figure 2.2(a) illustrates the problem for a fleet of three vehicles (A, C, and D) of different types (autonomous, conventional and dual-mode), and three requests (1, 2, and 3) spread over a hybrid street network. While pickup and delivery points of requests 2 and 3 lie entirely inside a single zone, passenger 1 must be picked up inside an AVZ and delivered in a CVZ location. Next, Figure 2.2(b) shows how we simplify this setup. Besides eliminating intermediate nodes of the real-world street network, we create a viable transportation network where vehicles and passengers are connected by their shortest paths, according to their site compatibility. While vehicle D is allowed to visit every pickup and delivery node, A can only visit the nodes inside the AVZ, and C can only visit the nodes inside the CVZ. Notice that although the pickup point of request 1 is inside the AVZ, vehicle A is not connected to it since A cannot reach the destination of request 1. Undirected lines represent two-way paths between nodes (possibly non-

symmetric), and directed lines highlight some of the problem's operational constraints, such as (i) vehicles can only start their route by visiting pickup points, and (ii) there are not paths from request destinations to origins.

2.3 Problem definition

The multi-depot site-dependent DARP is modeled on a directed graph $G=(N,E)$. The node set N is partitioned into $\{P,D,O\}$ where $P=\{1,...,n\}$ is both the set of pickup nodes and request indices, $D=\{n+1,...,2n\}$ is the set of destination nodes and O is the set of origins o^k of vehicles $k \in K$. We define the set O to better simulate a free-floating mobility service in which vehicles depart from distinct points within the service area (rather than departing from a central station) and park nearby the delivery point of the last serviced request upon finishing the service.

We consider that each vehicle $k \in K$ with capacity Q^k is from a type $m^k \in L$, and every transportation request i can be served by a subset of vehicle types $L_i \subseteq L$. Consequently, the arc set E is defined as $E=\{(i,j,m) \mid i \in O, j \in P \text{ or } i,j \in P \cup D, i \neq j \text{ and } i \neq n+j \text{ for } m \in L\}$, such that there may exist up to $|L|$ paths from i to j, each one having a travel time $t^m_{i,j}$. We assume that the set of types L coincides with the driving modes allowed in our hybrid maps (i.e., $L=\{\text{AV, CV, DV}\}$). To each node $i \in N$ is associated a load q_i, corresponding to the number of passengers, so that $q_i \geq 0 \; \forall i \in P$, $q_i = -q_{i-n}$ $\forall i \in D$ and $q_i = 0 \; \forall i \in O$. Additionally, the service duration d_i is a function of the number of passengers q_i entering/leaving a vehicle at node $i \in N$.

Moreover, let w_{pickup} be the maximum pickup delay, w_{ride} the maximum ride delay of all requests, and t_i the revealing time of request i. For a pickup and delivery pair (i,j) where $i \in P$ and $j \in D$, the earliest times (e_i and e_j) and latest times (l_i and l_j) to visit i and j are defined as follows: $(e_i,l_i)=(t_i,t_i+w_{\text{pickup}})$ and $(e_j,l_j)=(e_i+d_i+t^m_{i,j},e_j+w_{\text{ride}})$ for driving modes $m \in L$.

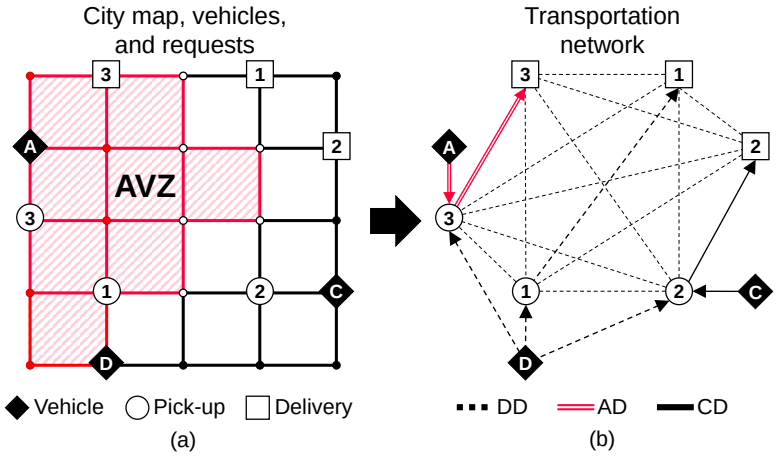


Figure 2.2: *(a) Real world input (hybrid street map, customers' pickup and delivery locations, and vehicle positions) and (b) corresponding viable transportation network.*

The decision variable $x_{i,j}^k$ is equal to 1 if the arc $(i,j,m^k) \in E$ is traversed by vehicle $k \in K$ and the load of a vehicle k upon leaving node $i \in N$ is ω_i^k. Regarding the time related variables, Δ_i^k is the ride time of request $i \in P$ in vehicle k and τ_i^k is the time at which vehicle k arrives at node $i \in N$. Ultimately, to streamline model execution, a preprocessing phase is carried out to eliminate decision variables that violate ride time, site-dependent, and capacity constraints. We define $\mathscr{X}$ as the set of valid rides comprised of feasible (k,i,j) combinations and an auxiliary set of valid visits $\mathscr{Q} = \{(k,i) \mid (k,i,j) \in \mathscr{X} \text{ or } (k,j,i) \in \mathscr{X}\}$. Table 2.1 summarizes the sets, variables, and parameters.

Table 2.1: *Sets, parameters, and variables of the MDSDDARP.*

Sets	
K	Vehicles
P	Pick-up nodes and request indices
D	Delivery nodes
O	Origin nodes o^k of vehicles $k \in K$
N	$= P \cup D \cup O$
$\mathscr{Q}$	Valid visits (k,i) for $k \in K$ and $i \in N$
$\mathscr{X}$	Valid rides (k,i,j) for $k \in K$ and $i,j \in N$
L	Vehicle types and driving modes
Parameters	
Vehicles	
m^k	Type of vehicle $k \in K$
o^k	Origin point of vehicle $k \in K$
Q^k	Capacity of vehicle $k \in K$
$p_{\text{base}}^{m^k}$	Base fare for servicing a passenger using vehicle k
$p_{\text{time}}^{m^k}$	Time-dependent rate for servicing passenger using vehicle k
$c_{\text{time}}^{m^k}$	Time-dependent operational cost of vehicle k
Requests	
d_i	Service duration at node $i \in N$
q_i	Number of passengers of request i
w_{pickup}	Maximum pickup time delay
w_{ride}	Maximum ride time delay
e_i	Earliest time at node i
l_i	Latest time at node i
Distances	
$t_{i,j}^m$	Travel time from node i to node j in mode $m \in L$
Variables	
$x_{i,j}^k$	(Binary) 1 if vehicle k traverses arc (i,j), 0 otherwise
τ_i^k	Arrival time of vehicle k at point i
Δ_i^k	In-vehicle delay of user i in vehicle k
ω_i^k	Load of vehicle k after visiting node i

The formulation of the MDSDDARP is as follows:

Maximize:

$$\sum_{\substack{(k,i,j)\in\mathcal{X}\\ i\in P}} (p^{m^k}_{\text{base}} + p^{m^k}_{\text{time}} \cdot t^{m^k}_{i,n+i}) \cdot x^k_{i,j} - \sum_{(k,i,j)\in\mathcal{X}} c^{m^k}_{\text{time}} \cdot t^{m^k}_{i,j} \cdot x^k_{i,j} \tag{2.1}$$

Subject to:

$$\sum_{(k,i,j)\in\mathcal{X}} x^k_{i,j} \leq 1 \qquad i \in P \tag{2.2}$$

$$\sum_{(k,o^k,j)\in\mathcal{X}} x^k_{o^k,j} \leq 1 \qquad k \in K \tag{2.3}$$

$$\sum_{(k,i,j)\in\mathcal{X}} x^k_{i,j} - \sum_{(k,i,j)\in\mathcal{X}} x^k_{i,n+j} = 0 \qquad k \in K, j \in P \tag{2.4}$$

$$\sum_{(k,i,j)\in\mathcal{X}} x^k_{i,j} - \sum_{(k,j,i)\in\mathcal{X}} x^k_{j,i} = 0 \qquad k \in K, j \in P \tag{2.5}$$

$$\sum_{(k,i,j)\in\mathcal{X}} x^k_{i,j} - \sum_{(k,j,i)\in\mathcal{X}} x^k_{j,i} \geq 0 \qquad k \in K, j \in D \tag{2.6}$$

$$\tau^k_j - \tau^k_i \geq t^k_{i,j} + d_i + M^k_{i,j}(x^k_{i,j} - 1) \qquad (k,i,j) \in \mathcal{X} \tag{2.7}$$

$$e_i \leq \tau^k_i \leq l_i \qquad (k,i) \in \mathcal{Q} \tag{2.8}$$

$$\Delta^k_i = \tau^k_{n+i} - (\tau^k_i + d_i) \qquad i \in P, (k,i) \in \mathcal{Q} \tag{2.9}$$

$$t^{m^k}_{i,n+i} \leq \Delta^k_i \leq t^{m^k}_{i,n+i} + w_{\text{ride}} \qquad i \in P, (k,i) \in \mathcal{Q} \tag{2.10}$$

$$w^k_j - \omega^k_i \geq q_j + W^k_{i,j}(x^k_{i,j} - 1) \qquad (k,i,j) \in \mathcal{X} \tag{2.11}$$

$$\max\{0, q_i\} \leq \omega^k_i \leq \min\{Q^k, Q^k + q_i\} \qquad (k,i) \in \mathcal{Q} \tag{2.12}$$

$$x^k_{i,j} \in \{0,1\} \qquad (k,i,j) \in \mathcal{X} \tag{2.13}$$

$$\omega^k_i, \tau^k_i \in \mathbb{N} \qquad (k,i) \in \mathcal{Q} \tag{2.14}$$

$$\Delta^k_i \in \mathbb{N} \qquad i \in P, (k,i) \in \mathcal{Q} \tag{2.15}$$

The objective function (2.1) maximizes the revenue obtained from passenger delivery while minimizing operational costs. Constraint (2.2) allows service denial, since not all customers need to be picked up, and constraint (2.3) defines that vehicles can potentially stay still in their origin nodes in case they are not scheduled. Constraint (2.4) imposes that if a vehicle visits a request pickup node it must also visit the associated delivery node. In turn, the flow constraints (2.5) and (2.6) ensure vehicles arrive and exit pickup nodes but may arrive and stay at delivery nodes, reflecting occasions in which a vehicle delivers its last user and waits in the vicinity for incoming requests. Constraints (2.7) and (2.8) guarantee adequate arrival times at nodes within predetermined time windows while constraints (2.9) and (2.10) define and limit the ride time of each request. In turn, feasible load flows are guaranteed by constraints (2.11) and (2.12). Finally, the validity of W and M at the linearized constraints (2.7) and (2.11) is ensured by setting $W^k_{i,j} \geq \min\{2Q^k, 2Q^k + q_i\}$ and $M^k_{i,j} \geq \max\{0, l_i + t^k_{i,j} + d_i - e_j\}$ $\forall (i,j,k) \in \mathscr{X}$.

2.4 Experimental study

Assuming automated driving areas are gradually implemented, we establish three key elements that may influence the performance of future heterogeneous fleets: the cost depreciation of autonomous technologies (2.4.1), the configuration of mixed-zone street networks (2.4.2), and the particular demand patterns arising from such zoned environments (2.4.3). In Section 2.4.4, we show how these elements are combined to generate our instance set.

2.4.1 Operational cost scenarios

It is widely assumed that AV technologies will become increasingly affordable until they eventually reach market saturation. Thus, we consider this gradual price depreciation to build three operational cost scenarios. In our study, operational costs vary according to the distance traveled (in seconds, considering an average speed of 40km/h) and are comprised of (i) general automotive costs (e.g., maintenance, parking, fuel, insurance), (ii) driver's labor and (iii) automation-related costs (e.g., maintenance of extra sensors, software, data storage, computing power). In Table 2.2, we show how these elements compose the total operational cost of each vehicle type for the potential scenarios $S01$, $S02$, and $S03$. Regardless of the vehicle type and scenario, we only vary the automation related costs (iii), from 0.003 €/s to 0.001 €/s, while keeping general automotive costs (i) and driver's labor (ii) stable at 0.001 €/s each.

Table 2.2: *Operational cost scenarios for vehicle types in relation to automation technology (€/s).*

	Vehicle type		
Cost scenario	AV	CV	DV
S01 (large price premium)	0.004	0.002	0.005
S02 (moderate price premium)	0.003	0.002	0.004
S03 (minimal price premium)	0.002	0.002	0.003

2.4.2 Mixed-zone street network

Since the development of dedicated automated areas is uncertain, we simulate the deployment of AVZs. First, we extract the map of Delft from OpenStreetMap and save it as a directed graph comprised of edges (streets) and nodes (intersections). To guarantee any two nodes are connected to each other, we eliminate all nodes not belonging to the graph's largest strongly connected component, resulting in a final street network with 2,123 nodes and 4,964 edges.

Generating AVZs consists of choosing z random nodes of this graph to be the zone origins and iteratively adding the neighboring edges and nodes from these origins, one level at a time, until at least an overarching coverage percentage p of strongly connected nodes within zones is achieved. In turn, all zones are interconnected by the shortest paths between their origins, yielding a strongly connected subnetwork representing a possible AVZ deployment. Figure 2.3 shows nine potential AVZ configurations for the street network in Delft considering a coverage percentage $p \in \{10\%, 25\%, 50\%\}$ and number of zone origins $z \in \{1, 2, 4\}$. While z varies the spatial configuration of AVZs, p simulates their expansion, so that different transition scenarios can be assessed. To broaden the variability of our test cases even further, we repeat the generating process 5 times, resulting in 45 transportation networks with distinct configurations of automated driving areas.

Finally, assuming three modes of driving are available, automated driving, conventional driving, and dual-mode driving, we create a look-up structure to store the shortest distances between every node and its reachable neighbors for each driving mode. Hence, a node belonging to an AVZ, for example, may access every node within the AVZ via automated driving paths, whereas nodes outside the AVZ can only be accessed via dual-mode driving paths.

2.4.3 Transportation demand versus zone configuration scenarios

We generate demand data to assess the influence of different passenger transportation patterns throughout different AD area configurations. This way, we can investigate the logistical outcome when the demand is restricted to a particular area (intra-zone transportation) or when the origin and destination nodes belong to different zones (inter-zone transportation). To do so, we consider three zone-crossing frequencies (low, moderate, and high) as shown in Table 2.3. Additionally, we investigate the implication of busy operational environments by uniformly distributing such demands on different time intervals, namely, 1, 5, 10, and 20 minutes.

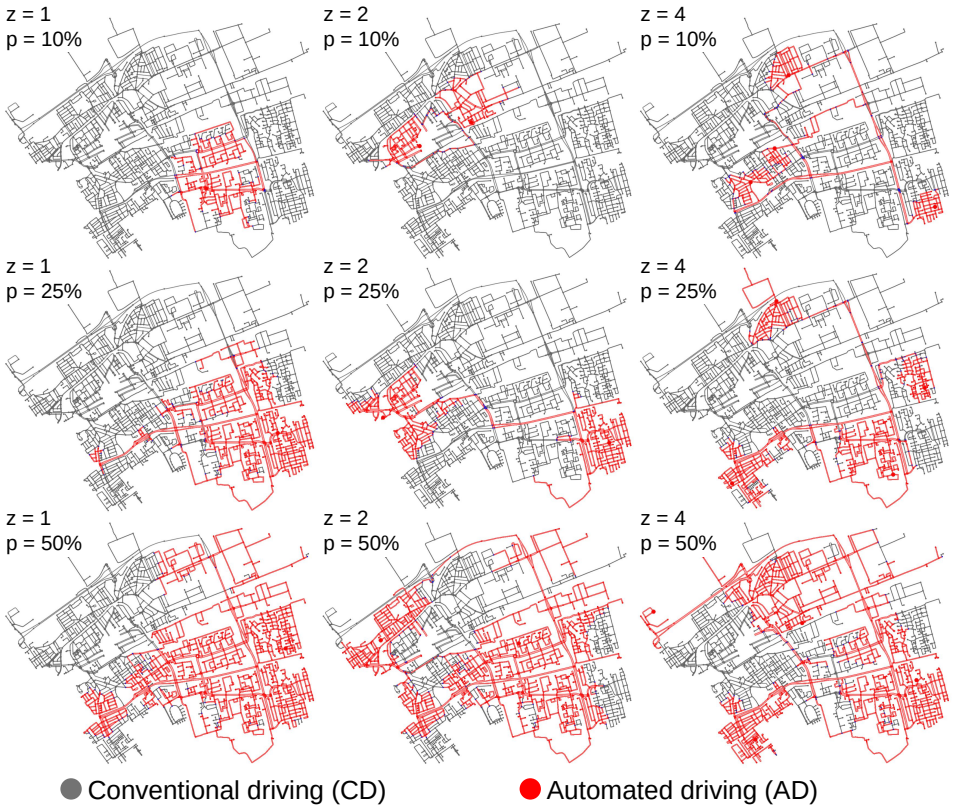


Figure 2.3: *Potential deployment of automated driving areas in Delft street network considering different combinations of number of zones z and coverage percentage p.*

Table 2.3: *Zone-crossing frequencies.*

	Zone-crossing		
Transportation pattern	High	Moderate	Low
intra-zone (AVZ - AVZ)	10%	30%	40%
intra-zone (CVZ - CVZ)	10%	30%	40%
inter-zone (AVZ - CVZ)	80%	40%	20%

2.4.4 Test cases

Table 2.4 summarizes the parameters we consider to generate a total of 14,580 test cases, and Table 2.5 defines values for service and fleet configuration parameters presented in Section 2.3.

Table 2.4: *Summary of test case settings totaling 14,580 instances.*

Parameter	Values
Number of vehicles	{15, 30, 60}
Number of requests	{10, 20, 40}
Operational cost scenarios	{S01, S02, S03}
AD coverage percentage	{10%, 25%, 50%}
Number of AD zones	{1, 2, 4}
Zone-crossing	{high, moderate, low}
Time interval (min)	{1, 5, 10, 20}
Number of zone configurations	5

Table 2.5: *Service and fleet configuration parameters.*

Parameter	Value
Base fare $p^m_{\text{base}}(\forall m \in L)$	€3.0
Time-dependent rate $p^m_{\text{time}}(\forall m \in L)$	€0.001/s
Vehicle capacity Q^k	5
Service delay per passenger	30 s
Pickup delay w_{pickup}	5 min
Ride delay w_{ride}	10 min

2.5 Results

Instances are run for up to 10 min (not including preprocessing times) using an Intel Xeon 3.7Ghz computer with 32 GB RAM, and the MILP model is implemented using the Python interface of the Gurobi 7.5.2 optimizer. Results are expressed in terms of the following performance markers:

- *Service rate* – The percentage of serviced requests.
- *Fleet utilization* – The percentage of the fleet actually used to service requests.
- *Mobility cost* – The relative operational cost to service each request.
- *Execution time* – The sum of preprocessing time (for creating a suitable transportation network and setting up the MILP model) and the solver runtime.
- *Fleet composition* – Percentages of each vehicle type that compose the final make-up of used vehicles.

Under the time boundary specified, optimal results were obtained in 91% of the test cases. We then use this optimal subset of results to carry out our analysis, assessing mean values of test cases grouped by different parameters (e.g., a result defined by the number of vehicles, number of requests, and operational cost scenarios, can be the mean of up to 540 tests cases). In the following sections, we present our main findings.

Table 2.6: *Service rate and fleet utilization on different time intervals, number of vehicles, and requests.*

		Service rate				Fleet utilization			
#R	#V	1 min	5 min	10 min	20 min	1 min	5 min	10 min	20 min
10	15	93.9%	96.5%	97.9%	99.4%	60.2%	58.6%	56.1%	51.8%
	30	99.4%	99.7%	99.7%	100.0%	31.9%	31.1%	30.3%	29.1%
	60	99.8%	100.0%	99.8%	100.0%	16.1%	16.0%	15.9%	15.6%
	Avg.	97.7%	98.7%	99.1%	99.8%	36.1%	35.2%	34.1%	32.1%
20	15	71.3%	80.4%	89.4%	96.5%	86.4%	87.2%	84.8%	80.7%
	30	97.1%	98.2%	99.6%	99.8%	60.2%	57.3%	54.5%	50.9%
	60	99.8%	99.9%	100.0%	100.0%	31.5%	30.6%	29.9%	28.9%
	Avg.	89.4%	92.9%	96.3%	98.8%	59.4%	58.4%	56.4%	53.5%
40	15	41.8%	49.8%	63.4%	84.5%	94.6%	93.6%	95.9%	95.8%
	30	75.5%	87.1%	94.2%	98.6%	87.8%	88.1%	82.1%	76.9%
	60	97.7%	99.3%	99.7%	99.8%	59.2%	56.1%	52.6%	49.1%
	Avg.	71.6%	78.7%	85.8%	94.3%	80.5%	79.2%	76.8%	73.9%

Time interval and fleet performance. The busier the logistical scenario, the higher the fleet utilization, and the fewer requests can be serviced. This trend can be verified in the average percentages of service rate and fleet utilization presented in Table 2.6, and it is especially remarkable when a large number of requests must be serviced by few vehicles.

Fleet composition & AD coverage. Fleet composition depends on the AD coverage once AVs and DVs are more prone to be scheduled when larger areas of the transportation network can accommodate autonomous driving. Figure 2.4 illustrates this trend. Each square represents the average percentage of a vehicle type for each combination of AD coverage, zone-crossing frequency, and operational cost scenario. Notice that within each zone-crossing category, the share of AV and DV vehicles tends to grow while the opposite occurs to CVs. Larger AVZs may lead to farther AV trips, making vehicles busier for longer periods and preventing them from servicing other customers. Hence, to comply with the service time constraints, the size of AV fleets must follow the growth of AVZs, whereas the size of CV fleets must follow the shrinkage of CVZs.

Fleet composition and zone-crossing frequencies. Although 80% of the requests must be serviced by DVs when a high crossing frequency is considered, this share is not directly reflected in the shares of DVs actually scheduled, which were around 50% no matter the AD coverage and cost scenario. This subpar representation may be associated with our transportation network's dimension: travel times between pickup and delivery nodes are generally short, such that fewer vehicles can service several users. In contrast, actual DV shares for the low and moderate inter-zone frequencies more closely resemble their correspondent zone-crossing frequencies in Table 2.3.

Fleet composition & operational cost scenarios. The depreciation of automated driving was found to be virtually irrelevant, especially at high zone-crossing frequencies. In such cases, there is no leeway for replacing conventional vehicles once most of the trips expressly require dual-mode vehicles. Consequently, it is more likely that such vehicles

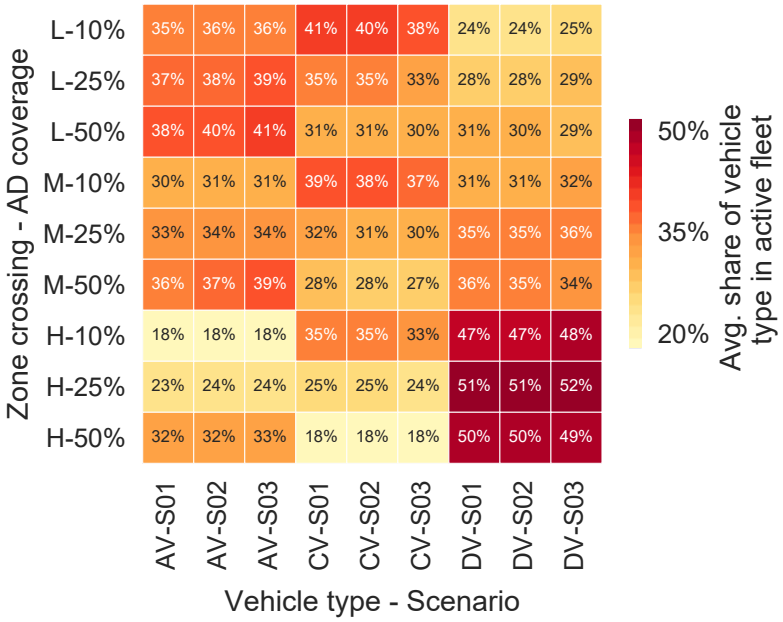


Figure 2.4: *Fleet composition according to automated driving coverage (10%, 25%, and 50%), zone-crossing frequency (L=Low, M=Moderate, and H=High), and operational cost scenarios (S01, S02, and S03).*

end up being used to also service intra-zone requests, no matter the operational costs in place. In contrast, if a low zone-crossing frequency is considered, the share of AVs and DVs slightly increases, whereas the share of CVs decreases: former DV rides are replaced by AV rides, and former CV rides are replaced by DVs.

The influence of operational costs. As expected, the depreciation of autonomous vehicle operational costs decreases mobility costs (see Table 2.7). Independently of the operational cost scenario, mobility costs tend to decrease as more vehicles are available since trip distances can be shorter. Additionally, the results also help define the tradeoff between fleet size and operational costs for a certain number of requests.

Execution time. As shown in Table 2.7, the preprocessing time is directly related to the number of vehicles and requests since it consists of looping through all decision variables to eliminate unfeasible answers. In contrast, the run time seemed to be more sensitive to the scarcity condition posed by certain operational environments than by the number of decision variables and constraints. In fact, the busier the operational environment, the longer the runtime: small fleets dealing with a far superior number of requests (e.g., 15 vehicles and 40 requests) are a more complex challenge to the branch-and-bound method of Gurobi solver.

Table 2.7: *Overall results for each operational cost scenario and number of vehicles and requests.*

#V	#R	Service rate	Fleet utilization	Mobility cost (€)			Pre-processing time(s)	Run time(s)
				S01	S02	S03		
15	10	96.9%	56.7%	1.43	1.19	0.92	0.0	0.1
	20	84.4%	84.8%	1.43	1.17	0.91	1.5	2.7
	40	59.8%	94.4%	1.29	1.07	0.83	9.2	459.3
30	10	99.7%	30.6%	1.27	1.05	0.82	0.5	0.3
	20	98.7%	55.7%	1.30	1.08	0.84	3.4	1.5
	40	88.2%	83.3%	1.31	1.08	0.84	18.9	188.9
60	10	99.9%	15.9%	1.17	0.96	0.75	2.0	0.7
	20	99.9%	30.2%	1.17	0.97	0.75	7.2	3.2
	40	99.0%	54.2%	1.21	1.00	0.78	38.8	72.9

2.6 Conclusions

This chapter addresses the research sub-question SQ1 by physically guaranteeing accessibility, a precondition to other service quality markers. We investigated how mixed-zone transportation networks can affect mobility services in light of the gradual development of autonomous infrastructures. We model the routing for such services considering a heterogeneous fleet comprised of conventional, autonomous, and dual-mode vehicles. We assume that only the latter can freely access every location in the network, whereas autonomous and conventional vehicles are restricted to operate within automated driving areas and non-automated driving areas, respectively. Then, such a vehicle/infrastructure compatibility requirement is used to formulate the problem as a variant of the heterogeneous dial-a-ride problem in which site-dependencies are taken into consideration.

The work builds a foundation for the increasingly important problem domain of partially autonomous vehicle routing. The results obtained with numerical experiments for the city of Delft, the Netherlands, provide detailed insights into how operational costs, service rates, and fleet utilization develop under scenarios of multiple infrastructural characteristics, fleet configurations, and technology costs in the next decades of vehicular automation. In particular, we show that fleet composition is strongly associated with the autonomous vehicle zones coverage and inter- or intra-zone demand patterns, which indicates that advanced data analytics will be essential for successfully deploying such (partially) autonomous mobility systems. Our findings help urban planners understand the importance of infrastructural decisions for the quality of local mobility services, and transportation providers gain fundamental insights on how to adjust a fleet to the infrastructural conditions of cities.

Throughout the subsequent chapters, we focus on logistical scenarios where the shortcomings of AV technology have been either partially or completely overcome. We set out to improve service quality and fleet productivity, considering that AVs can safely access all request locations. Hence, our case studies are designed for fleets of either SAE level 4 AVs (operating inside AVZs) or SAE level 5 AVs. In Chapter 3, we begin to explore the unique benefits that vehicle automation can bring to all mobility stakeholders.

Chapter 3

A business class for autonomous mobility

In Chapter 2 we showed how mobility-on-demand providers can guarantee full coverage in a mixed autonomous and non-autonomous environment. In a fully autonomous setting, however, the success of such providers depends on their ability to create mobility services that challenge traditional mobility products in terms of service quality, adequately catering to different segments of their passenger base. Planning models presented in *autonomous mobility-on-demand* (AMoD) literature, nonetheless, do not enable active control of service quality, sometimes allowing extensive delays and ultimately service rejection. In this chapter, we propose an AMoD solution able to control service quality across a heterogeneous user base on an operational planning level. To this end, we consider an elastic vehicle supply, that is, third-party AVs can be hired on-the-fly on a contract to address excess demand.

This chapter is organized as follows. Section 3.1 introduces the problem and Section 3.2 reviews the concept of service quality across related literature. Section 3.3 presents a multi-objective *mixed integer linear programming* (MILP) formulation for the *dial-a-ride problem with service quality contracts* (DARP-SQC) as well as a dynamic formulation for it, and describe how objectives optimized hierarchically in order of importance. Section 3.4 introduces a multi-objective matheuristic for the dynamic version of the problem. Then, Section 3.5 describes the experimental settings we use to create a series of case studies. These case studies comprise different user base compositions, service-level enforcement rates, and service quality contract settings. Next, Section 3.6 reports the results for both static and dynamic formulations, highlighting the tradeoffs between fulfilling user expectations and hiring new vehicles. Finally, Section 3.7 concludes our work and presents an outlook for future research on service quality and fleet size elasticity. Parts of this chapter have been submitted to a journal:

B. A. Beirigo, F. Schulte, and R. R. Negenborn. A business class for autonomous mobility-on-demand: Modeling service quality contracts in ridesharing systems. *Submitted to a journal.*

3.1 Introduction

The upcoming *autonomous vehicle* (AV) transition is expected to re-shape urban transportation in the next decades. The current personal mobility paradigm, mainly based on vehicle ownership, is likely to be phased out as *autonomous mobility-on-demand* (AMoD) systems develop [59]. All major car manufacturers have already anticipated this shift, announcing their plans to use *shared autonomous vehicle* (SAV) fleets to provide seamless transportation solutions to travelers rather than selling individual vehicles [47].

However, most AMoD systems are unable to control service reliability actively. Once a particular service level is defined (e.g., in terms of maximum waiting times), they determine a fleet size that maintains such level at a reasonable rate (see, e.g., [2, 12, 34, 112]). As high this rate might be, even a small lack of reliability can undermine the acceptance of mobility-on-demand services, especially for customers who cannot afford service rejection or excess delays. User acceptance, however, is a crucial factor in ridesharing systems: peak performance can only be achieved when a sufficiently large number of riders are willing to participate [50].

Moreover, ridesharing research generally does not consider that different customer groups widely vary in their behavior and desires [120]. Companies from various sectors acknowledge that profits and perceived *service quality* (SQ) can be improved by directly catering to segments of their customer base. For example, airline passengers can generally choose from first, business, or economy class, whereas current on-demand urban *transportation network companies* (TNCs), such as Uber and Lyft, offer a range of service tiers, spanning from pooled riders up to executive limousine services. Similarly, future AMoD systems are also likely to consider customer preferences when matching overlapping itineraries to construct viable routes.

this chapter proposes a hierarchical multi-objective approach where service quality, defined in terms of responsiveness, reliability, and privacy, can be thoroughly tuned to meet user needs. To this end, we take the perspective of a ridesharing AMoD company servicing a diversified user base where riders have both different sharing preferences and *service level* (SL) requirements, which are expressed in terms of maximum tolerance delays. By varying such parameters, we define a strict *service quality contract* (SQC) that specifies the minimum SQ-settings for three user classes, namely, *business* (B), *standard* (S), and *low-cost* (L). Whenever the operating fleet cannot sustain the service quality demanded by incoming requests, third-party vehicles are hired online to guarantee the expected user experience required by each SQ class. We assume that in a highly automated scenario, third-party owned, *freelance autonomous vehicles* (FAVs) are readily available to join the operating fleet, working on a contract during predefined periods or idle moments.

Figure 3.1 illustrates how a batch of requests from different SQ classes are serviced under this scenario, using a set of working and hireable vehicles. It can be seen from the output sub-figure 3.1-(b) that the high-SQ class request B5 prompts a hiring operation ($H3 \rightarrow W3$) and enjoys a private ride. In contrast, the low-SQ class request L4 waits longer to be picked up and does not travel directly to its destination (the ride is shared with user S2 momentarily). The working vehicle 2 stops rebalancing to service ODs 1, 2, and 4, whereas candidate vehicle 3 is hired to service ODs 5 and 3. Since working vehicle 1 cannot access any users in time, it repositions to the origin of the hireable

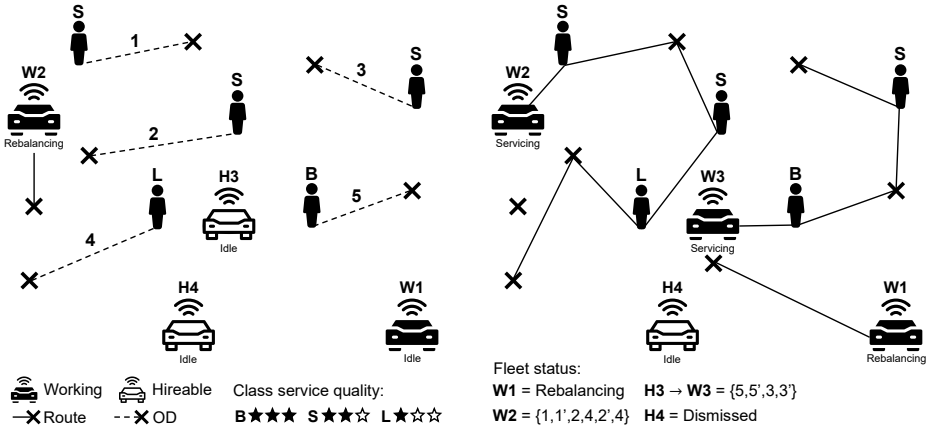


Figure 3.1: *(a) The user base is segmented into three classes, namely, business (B), standard (S), and low-cost (L), which have different expectations towards service quality. The fleet comprises working vehicles (third-party- or company-owned) and hireable vehicles (third-party-owned). (b) To adequately fulfill user expectations across classes, we rely on on-demand vehicle hiring, inflating the fleet size to avoid service-level violations.*

vehicle 3 (presumably, an undersupplied region), whereas the idle FAV 4 is dismissed. Therefore, hireable FAVs join the working fleet to help providers honoring service quality contracts by preventing delays and rejections from happening.

Our contribution is fourfold. First, we propose a new approach to control service quality in AMoD systems, which allows us to significantly better meet user service-level expectations of all considered user categories. Second, we acknowledge the inherent differences between user profiles by segmenting the demand into service quality classes and formalize the requirements of these profiles using detailed SQCs, which are upheld fully on an operational level using online hiring. This setup allows providers greater leeway since they can trade off class-specific user delay tolerance and on-demand hiring. Third, we contribute to the DARP literature by modeling such SQCs through service quality constraints and show how a typical AMoD provider can benefit from enforcing service quality in a real-world scenario. Finally, to deal with large-scale real-world instances, we propose a matheuristic that creates a graph of feasible visiting plans and optimally assigns these plans to active vehicles, hierarchically minimizing vehicle hiring and user dissatisfaction.

3.2 Related work

The ridesharing problem of pooling multiple users in a single ride has its roots in the dynamic variant of the classic *dial-a-ride problem* (DARP), introduced by Cordeau [23]. It belongs to a broader class of *transportation on demand* (TOD) problems related to the transportation of passengers or goods between specific origins and destinations at the

request of users [24]. What differs DARP from its TOD counterparts is the explicit concern with user inconvenience, generally controlled through service quality constraints [10].

For small instances (i.e., a few requests and vehicles), static DARP formulations have been solved to optimality using exact methods (e.g., branch-and-bound). In contrast, larger instances have been commonly addressed by combining construction heuristics (e.g., insertion algorithms), local search methods (e.g., 2-Opt), and metaheuristics (e.g., genetic algorithms, large-neighborhood search) [45]. Subsequently, we review the concept of service quality across several vehicle routing studies, highlighting the characteristics that enable us to develop the AMoD system we propose.

3.2.1 Service levels & service rate

In the realm of operations research, *service level*, *quality of service* and *service quality* are terms used interchangeably in DARP formulations to denote the degree to which a system can avoid excessive delays or comply with prespecified time window constraints (see [77]). However, compliance with these service levels may differ among typical DARP formulations. In a comprehensive literature review, Ho et al. [45] point out that studies occasionally adopt soft constraints, that is, they allow violations from the preferred service levels but penalize their occurrence in the objective function (e.g., [49]). When rejections are permitted, service providers can make selective visits and decide which requests to accommodate [45]. This is the case for most recent AMoD studies, in which the quality of proposed matching and schedule operations is also measured regarding the *service rate* (i.e., the percentage of requests serviced).

Service levels and service rates have been consistently shown to be inextricably linked. Molenbruch et al. [67] describe an approximately linear relationship between service level settings and various performance indicators, such as fleet size, total distance, empty mileage, passenger miles, and idle time. Accordingly, in an analysis of the minimum fleet sizing problem, Vazifeh et al. [112] have shown that the shorter is the connection time (i.e., the maximum delay between two consecutive trips), the bigger is the fleet required to keep service at a reasonable rate. Conversely, allowing longer connection times reduces fleet size, but at the expense of traffic and operational efficiency once vehicles may travel longer without any passenger on board. Ultimately, as Vazifeh et al. point out, setting service levels is a crucial design choice which can be more adequately addressed by mobility operators. Molenbruch et al. demonstrate that operational efficiency can be improved even further if such design choice considers heterogeneous user expectations regarding service levels.

Since providers have to find a compromise between fully meeting user expectations and operational costs, we set up this compromise using service-level enforcement rates. These rates determine the extent to which the system is committed to fulfilling each SQ class expectations. Aiming to decrease extra costs due to vehicle hiring, a provider may, for instance, strive to achieve the expected service levels of at least 90% of the users of a certain class, allowing longer waiting for the remaining 10%. By configuring these rates, providers can establish a service quality hierarchy across users from different classes, using the available fleet resources to first meet the needs of the most demanding users.

3.2.2 Heterogeneous users

The literature on classic routing problems provides many variants regarding user heterogeneity formulations. On *heterogeneous dial-a-ride* (HDARP) formulations, for example, a heterogeneous fleet has to service multiple user categories with different transportation requirements (viz., patient seat, stretcher, wheelchair place, accompanying staff seat) [78]. However, as pointed out by Ho et al., for most DARP variants, these categories focus on vehicle layout or equipment availability; service levels generally do not vary among users.

As argued by Molenbruch et al., it is plausible to assume service level requirements depend on why a trip is undertaken (e.g., scheduled appointment, leisure activity, commuting). Moreover, as shared autonomous transportation start to compete with current modes, users may adopt AMoD services that best reflect or improve their current ride experience. For example, through a migration analysis on Dutch commuters, Winter et al. [116] shows that preference towards SAVs is highest for those currently combining car and public transport.

By explicitly addressing heterogeneous user preferences, transportation providers can segment their user base and improve their operational efficiency and profitability by catering to each class accordingly. For instance, Wong et al. [117] extend a model of urban taxi services in congested networks to the case of multiple user classes (low- and high-income), multiple taxi modes (normal and luxury) with distinct combinations of service area restrictions (urban or rural), and fare levels (mileage- and congestion-based). Chen et al. [18] show that AMoD providers can find a compromise between profitability and service coverage by offering high *value of travel time* (VOTT) users, refined, work-enhancing vehicle environments at higher fares. Lokhandwala and Cai [60] allow users to choose either private or shared rides and the extent to which they can tolerate deviations from the original trip distance. Zhang et al. [123] also model the user's willingness to share rides and adapt their waiting tolerances from random hourly incomes.

Addressing heterogeneous user requirements is also common outside TOD formulations. For example, Smith et al. [97] investigate a generalization of the *dynamic traveling repairperson problem* (DTRP) that heavily penalizes service delays of high-priority demand classes. Bulhões et al. [13] consider a *vehicle routing problem with service levels* (VRP-SL), in which a logistics provider is tied to contractual obligations regarding strict delivery deadlines for several business partners. In order to balance service quality and operational costs, some groups of partner requests are subjected to different minimum quality thresholds (in terms of the share of serviced demand), depending on their relative profitability.

Still, throughout all these studies, users either leave the system unserved or endure longer delays when these requirements cannot be fulfilled. In contrast, we consider rejections as an unacceptable breach of service quality contract, such that we seek to inflate fleet size to at least guarantee full coverage. Further, similarly to Bulhões et al. [13], we control service levels on the operational planning level. We establish minimum service quality requirements for three user classes and assume some classes have priority over others. This way, service providers can exert greater control over the management

of scarce vehicle supply, steering the fleet to first fulfill the expectations of the most lucrative customer segment.

3.2.3 Heterogeneous vehicles & fleet size elasticity

Although most studies on urban mobility assume that a single service provider takes care of all requests, this centralized setting is unlikely to happen in reality since multiple providers can exist in one operation area [45]. As pointed out by Vazifeh et al., centralizing all operations entails a transition from a diversified mobility market, comprised of several micro-operators (e.g., taxi drivers), to a monopolistic market. The authors argue that, although optimal from a vehicle operational perspective, this approach is undesirable since it may lead to less competition and, consequently, higher prices for passengers. However, they show that most efficiency benefits (concerning fleet size) can still be achieved in an oligopolistic market with up to three operators. Accordingly, in a dial-a-ride context, Molenbruch et al. [68] suggest that enabling horizontal cooperation (i.e., allowing operators to exchange customers' information) can lead to additional benefits regarding empty trip reduction.

Vehicle automation further diversifies the mobility market since idle privately-owned AVs may also become micro-operators. Once vehicles remain parked about 95% of the time [93], incentives for individuals to simultaneously own and share their AVs when they are not in use will likely be high [14]. These idle hireable vehicles, which we refer to as FAVs, may be readily available during predefined time windows to join larger, centrally controlled fleets in exchange for compensation. As suggested by Hyland and Mahmassani [47], AV fleet managers can significantly benefit from such short-term fleet size elasticity, increasing or decreasing the fleet to adequately meet the demand.

Therefore, apart from the expected scheduling and routing services, a future on-demand transportation company may become a mediator between car-owners and renters, both with particular and heterogeneous constraints. Additionally, car-renting companies and car automakers, which will supposedly own large AV fleets, could also provide subscription plans to private users or associate themselves with third-party on-demand transportation services. For instance, the latter option might be a natural next step for automakers that are currently partnering or investing in ride-hailing companies.

In this setting, the advantages of service diversification and efficient scheduling can be balanced. For instance, different car owners could stipulate different profit margins or make their vehicles hireable under certain conditions, such as a minimum profit. Customers would also gain further freedom of choice. There would be many more options to choose from (instead of a uniform vehicle fleet), and alternative renting methods could be available (e.g., booking a vehicle for an entire day or week). Therefore, the best possible ride for a transportation demand would consider both customers' requirements and owners' constraints.

Ultimately, such a crowd-sourced AMoD system is expected to guarantee that the interactions between all involved parties are mutually beneficial. Users find affordable, tailored rides, car owners are adequately compensated for renting their vehicles, and the system profits as the matching/routing platform.

3.2.4 Shared autonomous vehicles systems

Following the taxonomy of Narayanan et al. [72] concerning the operation of SAV systems, we model an on-demand booking service with dynamic vehicle assignment and mixed sharing system (both ridesharing and carsharing setups are possible).

Table 3.1 shows how our work fits into related literature concerning service quality, fleet characteristics, and objective function. First, the "Service quality" header comprises three columns where we identify whether users (i) can demand shared and/or private rides, (ii) can choose from heterogeneous service levels, and (iii) can have priority over others. The "Fleet" header groups four columns indicating the SAV fleet characteristics, namely vehicle homogeneity (concerning capacity), vehicle ownership (company or private), availability (permanent or time windowed), and fleet size inflation enabled by dynamic hiring. Finally, the "Objective" column summarizes the goal of each approach. Typically, works that do not aim to minimize rejections assume travelers are willing to wait indefinitely but penalize this waiting accordingly. For instance, Hyland and Mahmassani [46] prevent travelers from going from assigned to unassigned as well as being reassigned more than once, besides penalizing the assignment to busy vehicles picking up or dropping off. We refer to these penalties in the objective function as "Min. assignment to busy" since they aim to improve the service quality of users previously assigned. In contrast, when waiting is bounded, a sufficiently large fleet size is considered. In [34], for example, the minimum fleet size is defined using a "seed" day simulation in order to guarantee travelers are never rejected and do not wait for more than ten minutes. Regarding the objective function labels, when a method aims to minimize costs that depend solely on the distance traveled, we label it with "D" (i.e., min. travel distance). Conversely, if other elements are used to calculate the cost (e.g., hourly salaries, VOTT, travel times), we use the label "C".

In contrast to typical formulations, we analyze a highly diversified market scenario in which private AV owners provide freelance rides to a mobility firm in exchange for compensation. The firm is assumed to own a homogeneous fleet but occasionally hire third-party vehicles to avoid user service-level violations. To make it more realistic, we assume such a backup fleet comprises heterogeneous vehicles with different capacities. However, opposed to most HDARP formulations in which vehicle/user compatibility depends on physical characteristics, we treat seats as commodities. Any customer can be transported by any vehicle as long as service quality can be maintained.

3.3 Problem formulations

In this section, we introduce the mathematical model for the problem at hand (Section 3.3.1) and present the changes necessary to implement the dynamic version (Section 3.3.2). Since both formulations aim to minimize a multi-objective function hierarchically, Section 3.3.3 further explains how this process is carried out using a lexicographic method.

Table 3.1: *On-demand booking and dynamic vehicle assignment ridesharing systems classified according to service quality, fleet characteristics, and objective function. Objectives separated by "-" are linearly combined and objectives separated by ">" are optimized hierarchically.*

	Service quality			Fleet				
Reference	(S)hared (P)rivate rides	Het. service levels	User class priority	Het. vehicle capacity	(C)ompany (P)rivate vehicles	(P)ermanent (W)indowed availability	Dynamic hiring	Objective
Fagnant and Kockelman [32]	P	-	-	-	C	P	-	W
Fagnant and Kockelman [34]	S	-	-	-	C	P	-	W
Alonso-Mora et al. [2]	S	-	-	-	C	P	-	R - W
Fiedler et al. [36]	S	-	-	-	C	P	-	D
Gueriau and Dusparic [41]	S	-	-	-	C	P	-	P
Simonetto et al. [95]	S	-	-	-	C	P	-	W
Santos and Xavier [87]	S	-	-	-	C	P	-	R - C
Wallar et al. [113]	S	-	-	✓	C	P	-	H - U
Hyland and Mahmassani [46]	P	-	-	-	C	P	-	W - D - B
Gurumurthy and Kockelman [42]	P	-	-	-	C	P	-	W
Chen et al. [18]	P	✓	-	-	C	P	-	C
Lokhandwala and Cai [60]	S, P	✓	-	-	C	P	-	D - S
Zhang et al. [123]	S, P	✓	-	-	C	P	-	C
This chapter	S, P	✓	✓	✓	C, P	P, W	✓	R > H > S > W

Objectives: **W**: Min. waiting, **R**: Min. rejections, **H**: Min. fleet size, **S**: Min. service level violation, **D**: Min. travel distance, **C**: Min. cost, **P**: Max. vehicle pickups, **U**: Max. utilization vehicle capacity, **B**: Min. assignment to busy.

3.3.1 Static DARP-SQC

The DARP-SQC is modeled on a digraph $G=(N,E)$. The node set N is partitioned into $\{P,D,O\}$ where $P=\{1,...,n\}$ is both the set of pickup nodes and request indices, $D=\{n+1,...,2n\}$ is the set of destination nodes, and O is the set of origins o^k of vehicles $k \in K$. We consider a free-floating fleet where vehicles do not need to depart from or come back to a central station. They can start from different origins and park nearby the delivery location of the last dropped request upon finishing the service. We let Z be the set of discretized locations on a street network, such that all nodes in N map to a single location in Z, using the function $g: N \to Z$. The shortest path between nodes i and j in N is given by $Z_{g(i),g(j)} \subseteq Z$ which we denote as $Z_{i,j}$ for simplicity. In turn, the minimum travel time to traverse path $Z_{i,j}$ is $t_{i,j}$.

Each vehicle k has capacity Q^k and is available to work on a contract during a time window $[e^k, l^k]$. For company vehicles, this window spans the entire planning horizon, whereas, for third-party vehicles, various time windows may be set. Users are segmented into service quality classes $c \in C$ such that the set of request indices P can be further partitioned as $P = \bigcup_{c \in C} P^c$. The arc set E is defined as $E = \{(i,j) \mid i \in O,\ j \in P \text{ or } i,j \in P \cup D, i \neq j \text{ and } i \neq n+j\}$. To each node $i \in N$ is associated a load q_i, corresponding to the number of passengers, such that $q_i = 0\ \forall i \in O$. Regarding the pickup and destination nodes, $q_i \geq 0\ \forall i \in P$ and $q_i = -q_{i-n}\ \forall i \in D$, that is, the load acquired upon picking up a request has to be equally consumed in the request's destination. Nodes $i \in N$ are also associated with a service duration d_i, which may correspond to, for instance, embarking and disembarking delays, for $i \in P$ and $i \in D$, respectively. Service levels for SQ classes $c \in C$ are represented by constants w^c_{pickup} and $w^c_{\text{tolerance}}$. The former corresponds to the expected maximum pickup delay over the revealing time e_i of request $i \in P^c$. The latter is the maximum delay tolerated by user i, which corresponds to the sum of both pickup and in-vehicle delays. Hence, when $w^c_{\text{pickup}} < w^c_{\text{tolerance}}$, users from class c are satisfied if picked up within w^c_{pickup} but can tolerate up to a $w^c_{\text{tolerance}}$ delay. Consequently, $w^c_{\text{tolerance}} - w^c_{\text{pickup}}$ corresponds to the in-vehicle delay. The binary parameter ρ^c is 1, if users from SQ class $c \in C$ demand private rides (i.e., ridesharing is disabled).

The decision variable $x^k_{i,j}$ is 1 when vehicle $k \in K$ traverses arc $(i,j) \in E$ and the load of a vehicle k upon leaving node $i \in N$ is ω^k_i. On the other hand, the decision variable y_i determines whether the service level of user $i \in P$ is achieved, setting a viable range for i's pickup delay variable δ_i. If we can meet user i service levels, that is, variable y_i is 1, then $0 \leq \delta_i \leq w^{c_i}_{\text{pickup}}$. On the contrary, if y_i is 0, the minimum service-level rate σ^{c_i} of SQ class $c_i \in C$ does not cover user i. In this case, user i needs to wait longer, that is, $w^{c_i}_{\text{pickup}} < \delta_i \leq w^c_{\text{tolerance}}$. Regarding the time related variables, Δ^k_i is the delay of request $i \in P$ in vehicle k and τ^k_i is the time at which vehicle k arrives at node $i \in N$. All elements of the DARP-SQC problem are summarized in Table 3.2.

Table 3.2: *Sets, parameters, and variables of the DARP-SQC.*

Sets	
K	Vehicles
C	Sorted SQ classes (from highest to lowest priority)
P	Pickup nodes and request indices
P^c	Pickup nodes and request indices of SQ class c
D	Delivery nodes
O	Origin nodes o^k of vehicles k
N	$= P \cup D \cup O$
Z	Discrete street network locations
$Z_{i,j}$	Shortest path between locations i and j
Parameters	
Vehicles	
e^k	Contract start time of vehicle k
l^k	Contract end time of vehicle k
o^k	Origin node of vehicle k
Q^k	Capacity of vehicle k
Requests	
c_i	SQ class of request i
d_i	Service duration at node $i \in N$
q_i	Number of passengers of request i
e_i	Earliest time of request i
Service quality classes	
σ^c	Service-level enforcement rate of SQ class c
w^c_{pickup}	Expected max. pickup delay of users in SQ class c
$w^c_{\text{tolerance}}$	Total delay tolerance of users in SQ class c
ρ^c	(Binary) 1 if SQ class c does not allow ridesharing, 0 otherwise
Distances	
$t_{i,j}$	Travel time from node i to node j
Variables	
$x^k_{i,j}$	(Binary) 1 if vehicle k traverses arc (i,j), 0 otherwise
y_i	(Binary) 1 if user i service level is achieved, 0 otherwise
δ_i	Pickup delay of user i
Δ^k_i	In-vehicle delay of user i in vehicle k
τ^k_i	Arrival time of vehicle k at node i
ω^k_i	Load of vehicle k after visiting node i

Multi-objective function

Let $\mathscr{X}$ be the solution space, such that $s \in \mathscr{X}$ is given by $s = \{x^k_{i,j}, h_i, \delta_i, \Delta^k_i\}$. We consider a multi-objective function with non-commensurate objectives, ranked according to their respective priority. First, we aim to minimize the fleet size, by minimizing the total number of seats across all vehicles with different capacities

$$f_{\text{fleet}}(s) = \sum_{k \in K} \sum_{j \in P} Q^k x^k_{o^k,j}. \tag{3.1}$$

Next, assuming classes in C are sorted in decreasing priority order (e.g., business, standard, low-cost), we aim to minimize, the sum of service level violations

$$\{f^c_{\text{violate}}(s) = \sum_{i \in P^c} (y_i - 1) \mid \forall c \in C\}, \tag{3.2}$$

and then the total waiting time for each class

$$\{f^c_{\text{wait}}(s) = \sum_{i \in P^c} \delta_i + \sum_{i \in P^c} \sum_{k \in K} \Delta^k_i \mid \forall c \in C\}. \tag{3.3}$$

Finally, the multi-objective function is given by

$$f(s) = \{f_{\text{fleet}}(s),\ f^c_{\text{violate}}(s)\ \forall c \in C,\ f^c_{\text{wait}}(s)\ \forall c \in C\}. \tag{3.4}$$

MILP model

The formulation of the DARP-SQC is as follows:

Minimize:

$$f(x^k_{i,j}, h_i, \delta_i, \Delta^k_i) \tag{3.5}$$

Subject to:

$$\sum_{k \in K} \sum_{j \in N} x^k_{i,j} = 1 \qquad \forall i \in P \tag{3.6}$$

$$\sum_{i \in N} x^k_{i,j} - \sum_{i \in N} x^k_{i,n+j} = 0 \qquad \forall j \in P, k \in K \tag{3.7}$$

$$\sum_{i \in P} x^k_{o^k,i} \leq 1 \qquad \forall k \in K \tag{3.8}$$

$$\sum_{i \in N} x^k_{i,j} - \sum_{i \in N} x^k_{j,i} \geq 0 \qquad \forall k \in K, j \in P \cup D \tag{3.9}$$

$$e^k \leq \tau^k_i \leq l^k \qquad k \in K, \forall i \in N \tag{3.10}$$

$$\tau_j^k \geq (t_{i,j} + d_i + \tau_i^k) x_{i,j}^k \qquad \forall i,j \in N,\, k \in K \qquad (3.11)$$

$$\Delta_i^k = \tau_{n+i}^k - t_{i,n+i} - (\tau_i^k + d_i) \qquad \forall i \in P,\, k \in K \qquad (3.12)$$

$$\omega_j^k \geq (\omega_{n+i}^k + q_j) x_{i,j}^k \qquad \forall i,j \in N,\, k \in K \qquad (3.13)$$

$$\max\{0, q_i\} \leq \omega_i^k \leq \min\{Q^k, Q^k + q_i\} \qquad \forall i \in N,\, k \in K \qquad (3.14)$$

$$\sum_{i \in P^c} y_i \geq \left\lceil \sigma^c \cdot |P^c| \right\rceil \qquad \forall c \in C \qquad (3.15)$$

$$w_{\text{pickup}}^{c_i}(1 - y_i) \leq \delta_i \leq w_{\text{pickup}}^{c_i} y_i + w_{\text{tolerance}}^{c_i}(1 - y_i) \qquad \forall i \in P \qquad (3.16)$$

$$e_i \leq \tau_i^k \leq e_i + \delta_i \qquad \forall i \in P,\, k \in K \qquad (3.17)$$

$$\Delta_i^k + \delta_i \leq w_{\text{tolerance}}^{c_i} \qquad \forall i \in P,\, k \in K \qquad (3.18)$$

$$\sum_{k \in K} \omega_i^k = q_i \qquad \forall i \in P : \rho^{c_i} = 1 \qquad (3.19)$$

$$\sum_{k \in K} x_{i,n+i}^k = 1 \qquad \forall i \in P : \rho^{c_i} = 1 \qquad (3.20)$$

$$x_{i,j}^k \in \{0,1\} \qquad \forall k \in K,\, i,j \in N \qquad (3.21)$$

$$y_i \in \{0,1\},\, \delta_i \in \mathbb{N} \qquad \forall i \in P \qquad (3.22)$$

$$\Delta_i^k \in \mathbb{N} \qquad \forall k \in K, \forall i \in P \qquad (3.23)$$

$$\tau_i^k,\, \omega_i^k \in \mathbb{N} \qquad \forall k \in K,\, \forall i \in N \qquad (3.24)$$

The aim of the hierarchical multi-objective function (3.5) is first to determine the minimum fleet size and vehicle mix that satisfies the demand and then to improve class service levels. Constraints (3.6) and (3.7) ensure that all users are serviced exactly once and that the same vehicle visits their origin and destination nodes. Constraints (3.8) and (3.9) guarantee that every scheduled vehicle k departs from its origin o^k and stops at the delivery node of its last request. Constraints (3.10) impose that vehicles can only pick up and deliver users within their working time window. The consistency of arrival and ride times is ensured by constraints (3.11) and (3.12), whereas the consistency of load variables is ensured by inequalities (3.13) and (3.14). Constraints (3.15)-(3.20) implement the user SQCs. First, constraints (3.15) enforce that the service-level expectations of a minimum share of users from each class are met. Next, Constraints (3.16), (3.17), and (3.18) ensure that service levels are consistent with maximum pickup and in-vehicle delays. Then, equalities (3.19) and (3.20) ensure that users whose class entails a private ride are picked up by empty vehicles and travel directly to their destination. Finally, constraints (3.21)-(3.24) declare the variables.

The quadratic constraints (3.11) and (3.13) are linearized as in Cordeau et al. [24]:

$$\tau_j^k - \tau_i^k \geq t_{i,j} + d_i - M_{i,j}^k(1 - x_{i,j}^k) \qquad \forall i,j \in N,\, k \in K \tag{3.25}$$

$$Q_j^k - Q_{n+1}^k \geq q_j - W_{i,j}^k(1 - x_{i,j}^k) \qquad \forall i,j \in N,\, k \in K \tag{3.26}$$

The validity of the Big-M constants W and M in (3.25) and (3.26) is ensured by setting $W_{i,j}^k \geq \min\{2Q^k, 2Q^k + q_i\}$ and $M_{i,j}^k \geq \max\{0, l_i + t_{i,j} + d_i - e_j\}$ $\forall k \in K$ and $i,j \in N$.

3.3.2 Dynamic DARP-SQC

In order to deal with real-world instances, we break apart the time horizon into κ-second rounds $t \in \{0,1,2,\ldots,T\}$. At each round t, we process a request batch indexed by P_t, which is comprised of requests accumulated throughout the time slot $[\kappa\cdot(t-1),\ \kappa\cdot t)$. Additionally, we search for available freelance vehicles parked throughout locations in Z to construct the set of backup FAVs $K_t^{\text{FAV}} \subseteq K$, which can privately service all requests in P_t. In the following, we further detail the elements we use to expand the static DARP-SQC problem definition presented in Section 3.3.1 to accommodate dynamic execution. These elements are summarized in Table 3.3.

Requests. A request r_i is defined by a tuple $\{i, i', \tau_i^k, \tau_{i'}^k, e_i, e_{i'}, q_i, c_i\}$, which consists of origin $i \in P$, destination $i' \in D$, with respective expected arrival times τ_i^k and $\tau_{i'}^k$, placement time e_i, minimum delivery time $e_{i'} = e_i + t_{i,i'}$, load q_i, and service level class c_i.

Vehicles. We assume all vehicles k are associated to visiting plans $v^k = \{\mathscr{P}^k, \mathscr{R}^k, \mathscr{S}^k\}$. Sets $\mathscr{P}^k$ and $\mathscr{R}^k$ are comprised of passengers (i.e., picked up requests) and assigned requests, respectively. In turn, $\mathscr{S}^k = \{\mathscr{S}_0^k, \mathscr{S}_1^k, \ldots, \mathscr{S}_m^k\}$ is a sequence of nodes representing the vehicle's itinerary, such that $\mathscr{S}^k \subseteq N$. While $\mathscr{S}_0^k$ is vehicle k's last visited node, $\mathscr{S}_1^k, \ldots, \mathscr{S}_m^k$ are the subsequent m nodes to visit. Hence, vehicle k's current load is $Q_{\mathscr{S}_0^k}^k = \sum_{r_i \in \mathscr{P}^k} q_i$, which corresponds to the load after visiting S_0^k in the interval $[\tau_{\mathscr{S}_0^k}^k, \tau_{\mathscr{S}_0^k}^k + d_{\mathscr{S}_0^k}^k)$.

Visiting plan feasibility. A valid visiting plan v^k from vehicle k consists of a sequence of trip pairs $(i,j) \in \{(\mathscr{S}_0^k, \mathscr{S}_1^k), (\mathscr{S}_1^k, \mathscr{S}_2^k), \ldots, (\mathscr{S}_{m-1}^k, \mathscr{S}_m^k)\}$ that represent a feasible solution with respect to the following constraints:

C1) $e_j \leq \tau_i^k + d_i + t_{i,j} \leq \tau_j^k$ (arrival consistency)
C2) $\omega_i^k + q_j \leq Q^k$ (load consistency)
C3) if $i \in P \wedge \rho^{c_i} = 1$ then $j = i'$ (privacy requirement)
C4) $\tau_i^k \leq l^k$ (vehicle contract deadline)

It is worth noting that when a request is assigned to a vehicle, we assume that it can still be further delayed or even picked up by different vehicles but never rejected. Such flexibility allows a higher number of feasible rides to arise at each period, which ultimately favors the inclusion of high priority users by disrupting previous visiting plans.

Table 3.3: *Sets, parameters, and variables of the dynamic formulation.*

Sets	
Z^*	Regional centers of street network locations Z
K^{PAV}	Company vehicles
K_t^{FAV}	Hireable vehicles at time t
K_t^{H}	Vehicles hired at time t
K_t^{P}	Parked vehicles at time t
$\mathscr{P}^k$	Passengers (picked up requests) of vehicle k
$\mathscr{R}^k$	Assigned requests (non picked up) of vehicle k
$\mathscr{S}^k$	Node itinerary of vehicle k
v^k	Visiting plan $\{\mathscr{P}^k, \mathscr{R}^k, \mathscr{S}^k\}$ of vehicle k
$\mathscr{V}_R^k$	Candidate visiting plans where k is in the *rebalancing* state
$\mathscr{V}_S^k$	Candidate visiting plans where k is in the *servicing* state
v_{idle}^k	Candidate visiting plan where k is in the *idle* state
$\mathscr{V}^k$	Feasible visiting plans $\mathscr{V}_S^k \cup \mathscr{V}_R^k \cup \{v_{\text{idle}}^k\}$ for vehicle k
$\mathscr{V}$	Visiting plans $\bigcup_{k \in K} \mathscr{V}^k$ in ERTV graph
$\mathscr{V}_i$	Visiting plans in $\mathscr{V}$ including request i
$\mathscr{V}_i^{\text{SL}}$	Visiting plans in $\mathscr{V}$ that meet request i target SL
P^{A}	Requests previously assigned to vehicles
B_t	Request batch placed in period t
P_t	$= B_t \cup P^{\text{A}}$
P_t^c	Requests in P_t of class $c \in C$
O_t^*	Origins $o^k \in Z^*$ of hired vehicles K_t^{H}
P_t^{U}	Pickup nodes of service-level violated requests at time t
J_t	Rebalancing targets $O_t^* \cup P_t^{\text{U}}$
Parameters	
γ_{hire}	Hiring penalty
γ_{sl}	Service-level violation penalty
γ_{reject}	Rejection penalty
κ	Round duration
l^k	Contract deadline of vehicle k
T	Total time horizon
s	Maximal hiring delay
Variables	
h^k	(Binary) 1 if vehicle $k \in K_t^{\text{FAV}}$ is hired, 0 otherwise
x_v	(Binary) 1 if visiting plan v is chosen, 0 otherwise
y_i	(Binary) 1 if service level of request i is achieved, 0 otherwise
z_i	(Binary) 1 if request i is rejected, 0 otherwise
δ_{iv}	Pickup delay of user i in visiting plan v
Δ_v	Delay sum of all requests in visiting plan v
Δ_v^c	Delay sum of class c requests in visiting plan v
a^c	Number of service-level violations in class c

Vehicle states and visiting plan types. At each round, a vehicle k can execute three actions, namely, (i) move to pick up or deliver requests, (ii) stay idle at its current position, and (iii) move empty to another location. Accordingly, we consider that vehicles k executing (i), (ii), or (iii), are in states *servicing*, *idle*, or *rebalancing*, respectively. The best plan is chosen from a set of feasible candidates $\mathcal{V}^k$, which is partitioned based on the possible vehicle states using

- $\mathcal{V}_S^k$ Set of candidate visiting plans where vehicle is servicing users, that is, $|\mathcal{P}^k|>0$ or $|\mathcal{R}^k|>0$,
- v_{idle}^k Candidate visiting plan where vehicle is idle, that is, $\mathcal{R}^k=\mathcal{P}^k=\emptyset$ and $|\mathcal{S}^k|=1$, such that $\mathcal{S}_0^k$ is a parking place,
- $\mathcal{V}_R^k$ Set of candidate visiting plans where vehicle is rebalancing, that is, $\mathcal{R}^k=\mathcal{P}^k=\emptyset$ and $|\mathcal{S}^k|=2$, such that $\mathcal{S}_0^k$ is a parking place and $\mathcal{S}_1^k$ is a rebalance target.

Updating vehicle progress. We use the function `updateProgress(k,t)` to update visiting plans. The time period t is used to compute the progress of vehicle k throughout the itinerary $\mathcal{S}^k$ and update sets $\mathcal{R}^k$ and $\mathcal{P}^k$. First, regardless of node type, $\forall i \in \mathcal{S}^k$, if $\tau_i^k \geq e_i$, $\mathcal{S}^k=\mathcal{S}^k\setminus\{i\}$. If i is a pickup node, then $\mathcal{R}^k=\mathcal{R}^k\setminus\{r_i\}$ and $\mathcal{P}^k=\mathcal{P}^k\cup\{r_i\}$, whereas if i is a delivery node, then $\mathcal{P}^k=\mathcal{P}^k\setminus\{r_i\}$.

Interrupting visiting plans. By breaking apart the current leg $(i,j)=(g(\mathcal{S}_0^k),g(\mathcal{S}_1^k))$ of a vehicle k into a sequence of intermediate nodes throughout the shortest path $Z_{i,j}=\{i,b_1,b_2,\ldots,b_n,j\}\subseteq Z$, we can use the current time t, the arrival time $\tau_{\mathcal{S}_i^k}^k$, and travel times $t_{i,j}$ to determine the break point $b\in Z_{i,j}$ from where we can start a new itinerary. Using breakpoints adds flexibility to the scheduling process, since we can change visiting plans every period, before reaching location j. For instance, a servicing vehicle can take a turn at any break point b throughout $Z_{i,j}$ to pick up another user, or, provided that it has no passengers, empty the request list and rebalance to parking place b. In turn, a vehicle moving to a high demand region can interrupt the rebalancing process at b to pick up a request occurring nearby.

Optimal itineraries. Based on the passengers $\mathcal{P}^k$ of vehicle k and a candidate request set $\mathcal{R}$, there can be several feasible $\mathcal{S}$ itineraries created using nodes from $\mathcal{P}^k\cup\mathcal{R}$ that fulfill constraints C1, C2, C3, and C4. These itineraries compose candidate visiting plans $v=\{\mathcal{P}^k,\mathcal{R},\mathcal{S}\}$ in the set $\mathcal{V}_S^k$. The total delay associated with an itinerary $\mathcal{S}$ of visiting plan v consists of the sum of the arrival delays at destination nodes, which is given by

$$\Delta_v=\sum_{i'\in\ \mathcal{S}\cap D}\tau_{i'}^k-e_{i'}.$$

Using delays Δ_v, we can determine the visiting plan featuring the global minimum total delay

$$v_g^k=\underset{v\in\mathcal{V}_S^k}{\arg\min}\,\Delta_v. \tag{3.27}$$

From a SQ class perspective, the total delay of requests from class c consists of the sum of delays to reach these requests' destinations, which is given by

$$\Delta_v^c = \sum_{\substack{i' \in S \cap D \\ c_i = c}} \tau_{i'}^k - e_{i'}.$$

Likewise, using delays Δ_v^c, we can determine the visiting plan featuring the minimum total delay for class c

$$v_c^k = \arg\min_{v \in \mathcal{V}_S^k} \Delta_v^c. \tag{3.28}$$

Hiring. We outsource requests to privately-owned vehicles (FAVs) whenever the current fleet can no longer meet the minimum service level requirements of SQ classes. For each period t, we search regional centers for available FAVs that can back up the PAV fleet if it cannot service a request batch P_t adequately. We assume that such freelance vehicles are readily available to join the fleet in exchange for compensation, and their supply is sufficient to match the overall excess demand. We collect these vehicles in set $K_t^{\text{FAV}} = \{\texttt{closestFAVToRequest}(i, Z^*) \mid \forall i \in P_t\}$, where `closestFAVToRequest(`$i, Z^*$`)` is a function that returns the vehicle k parked at regional center location in $Z^* \subseteq Z$ that can pick up request i.

To determine regional centers in Z^*, we implement a variant of the facility-location problem proposed by Toregas et al. [103]. Our formulation aims to determine the minimum set of facilities that together can cover all other locations in Z within s time units. For instance, a long s results in fewer regional centers, increasing the pickup time of requests and, consequently, service-level violations. Conversely, a short s leads to many regional centers, allowing the hireable fleet to access users faster. In this chapter, we refer to s as the *maximal hiring delay* and adjust it such that hired vehicles can pick requests within their expected service level, regardless of class.

All vehicles $k \in K_t^{\text{FAV}}$ have contract deadlines $min\{l^k, T\}$, such that k is dismissed at a later time t' as long as $t' >= l^k$. Finally, to guarantee PAV availability, we set $l^k = T$ for all vehicles $k \in K^{\text{PAV}}$. Additionally, we consider hired vehicles to have precisely the capacity of the number of passengers determined by the request they are supposed to service. Therefore, throughout the simulation, the working fleet dynamically changes from homogeneous to heterogeneous, especially at high-demand times.

Decision variables. Decision variables are grouped in solution tuples $s = \{x_v, y_i, z_i, h^k\}$ with $x_v, y_i, z_i, h^k \in \{0,1\}$ such that the solution space is defined using

$$\mathcal{X} = \{x_v, y_i, z_i, h^k \mid \forall v \in \mathcal{V}^k,\ \forall k \in K \text{ and } \forall i \in P_t\}.$$

Variable $x_v = 1$ when the visiting plan $v \in \mathcal{V}^k$ from vehicle $k \in K$ is chosen, and variables y_i and z_i help to distinguish how user i is serviced. If $y_i = 1$, the system could meet request i class service levels, whereas if $z_i = 1$ request i is rejected. Finally, for vehicles $k \in K_t^{\text{FAV}}$, $h^k = 1$ signals that k was hired to carry out a visiting plan at the current period t.

Multi-objective function

For the dynamic version of our problem, we also propose a multi-objective function. First, we aim to minimize, in turn, the sum of rejection penalties γ_{reject} given by the ordered list of objectives

$$\{f^c_{\text{reject}}(s) = \sum_{i \in P^c} \gamma_{\text{reject}} \cdot z_i \mid \forall c \in C\}. \tag{3.29}$$

After preventing rejections, we aim to hire the least number of vehicles necessary to meet user service levels, such that we aim to minimize

$$f_{\text{hire}}(s) = \sum_{k \in K^{\text{FAV}}_t} Q^k h^k. \quad . \tag{3.30}$$

Next, we seek to minimize the sum of violation penalties γ_{sl} in the ordered list of objectives

$$\{f^c_{\text{violate}}(s) = \sum_{i \in P^c} \gamma_{\text{sl}} \cdot (y_i - 1) \mid \forall c \in C\}. \tag{3.31}$$

Then, we consider the minimization of waiting times across classes:

$$\{f^c_{\text{wait}}(s) = \sum_{v \in \mathcal{V}^k} \Delta^c_v \cdot x_v \mid \forall c \in C\}. \tag{3.32}$$

Finally, we define the multi-objective function of our dynamic formulation using

$$f(s) = \{f^c_{\text{reject}}(s)\ \forall c \in C,\ f_{\text{hire}}(s),\ f^c_{\text{violate}}(s)\ \forall c \in C,\ f^c_{\text{wait}}(s)\ \forall c \in C\}. \tag{3.33}$$

According to each class priority, the order of the objectives guarantees that visiting plans are set up first to avoid rejections. In the next steps, vehicles are hired only to prevent service level violations, which are minimized subsequently also following the order of C. Once optimal service level distribution is guaranteed, the solution featuring the minimum waiting time is chosen.

3.3.3 Lexicographic method for multi-objective optimization

We use the lexicographic method to determine the optimal solution for the multi-objective function $f(s)$ (for both static and dynamic models). This method consists of solving, in decreasing priority order, a sequence of single-objective problems

$$\text{Minimize } \{f_r(s) \mid f_l(s) \leq \psi_l,\ l = 1, \ldots, r-1,\ s \in \mathcal{X}\},$$

where $r \in \{1, 2, \ldots, |f|\}$ is the index of the current objective. A solution s is feasible only if it does not degrade the optimal values ψ_l found for previous objectives f_l with higher priority. Although we adopt a hierarchical optimization approach, it is worth noting that service quality objectives such as f^{violate} and f^{wait} are good candidates for linear combination. By deliberately weighing the tradeoffs, providers could establish a utility function that best represents their efforts to cater to users across classes.

3.4 A matheuristic for the dynamic DARP-SQC

The following sections describe how we build on the state-of-the-art method proposed by Alonso-Mora et al. [2] to solve the problem's dynamic version. The method was chosen because (i) its conventional formulation offers optimality guarantees and (ii) it is flexible to accommodate the contributions put forward in this chapter, namely, service level constraints and hiring capabilities.

First, request demand and vehicle supply information are combined into a pairwise-shareability graph (Section 3.4.1), which is subsequently used to iteratively compute a graph of feasible visiting plans (Section 3.4.2). Next, visiting plans are optimally assigned to vehicles using a multi-objective MILP formulation whose goal is to minimize user dissatisfaction hierarchically (according to the importance of each class) while hiring the least number of vehicles (Section 3.4.3). Then, a MILP formulation is used to optimally rebalance idle vehicles to underserved areas, where either service level violations (i.e., extra delays or rejections) or vehicle hirings occurred (Section 3.4.4).

The necessary steps to calculate a complete solution are described in Algorithm 3.1 and summarized in Figure 3.2, which provides an overview of our method using as an example the input presented in Figure 3.1 (a). Since the construction steps I and II (where feasible visiting plans are determined) are followed by assignment steps III and IV (where vehicles are optimally assigned to visiting plans), the complete strategy consists of a matheuristic, that is, a heuristic that incorporates phases where mathematical programming models are solved [4].

Algorithm 3.1: Dynamic DARP-SQC matheuristic

Input: Request index batches $P = \{B_1, B_2, B_3, \ldots, B_T\}$ for total horizon T, service level classes C, initial fleet K^{PAV} randomly distributed throughout street network locations in Z, and regional centers Z^*.

$K = K^{\text{PAV}}$
for $t = 0, 1, 2, \ldots, T$ **do**
  `updateProgress(`k,t`)`, $\forall k \in K$
  $K = K \setminus \{k\}$ if $t \geq l^k$ $\forall k \in K$ `//Remove vehicles whose contracts expired`
  $P^{\text{A}} = \{i \mid \forall r_i \in \bigcup_{k \in K} \mathscr{R}^k\}$ `//Build set of assigned requests`
  $P_t = B_t \cup P^{\text{A}}$ `//Build set of requests in progress`
  $K_t^{\text{FAV}} = \{$`closestFAVToRequest(`i, Z^*`)` $\mid \forall i \in P_t\}$
  $K = K \cup K_t^{\text{FAV}}$
  Build RV graph using vehicles K and requests P_t (Section 3.4.1)
  Build ERTV graph from RV graph (Section 3.4.2)
  (MILP) Match visiting plans to vehicles (Section 3.4.3)
  $K = K \setminus \{k\}$ if $h^k = 0$ $\forall k \in K_t^{\text{FAV}}$ `//Remove unhired FAVs`
  $O_t^* = \{o^k \mid h^k = 1 \ \forall k \in K_t^{\text{FAV}}\}$ `//Origins of hired vehicles`
  $P_t^{\text{U}} = \{i \mid y_i = 0,\ r_i \in P_t\}$ `//Origins of SL violated requests`
  $J_t = \{i \mid \forall\ i \in O_t^* \cup P_t^{\text{U}}\}$ `//Rebalancing targets`
  $K_t^{\text{P}} = \{k \mid$ if $v^k = v_{\text{idle}}^k \forall k \in K\}$ `//Idle vehicles`
  Create rebalancing visiting plans $\mathscr{V}_R^k$ using K_t^{P} and J_t
  (MILP) Match rebalancing visiting plans to vehicles (Section 3.4.4)
  Implement all visiting plans $v \in \mathscr{V}^k$ if $x_v = 1$ $\forall k \in K$

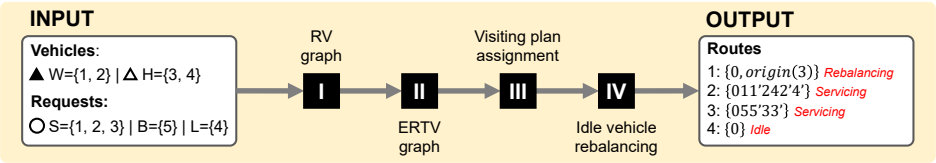


Figure 3.2: *Method overview for the input example presented in Figure 3.1 (a). Input is processed into a RV graph (Figure 3.3) where initial ridesharing opportunities are identified. Then, the RV graph is used to build an ERTV -graph (Figure 3.4), which connects all vehicles and requests through feasible visiting plans. Next, these plans are optimally assigned to available vehicles (Figure 3.5). Finally, remaining idle vehicles are optimally rebalanced to underserved areas (Figure 3.6). The output consists of updated vehicles plans, which are illustrated in Figure 3.1 (b).*

3.4.1 Pairwise request-vehicle graph

The pairwise *request-vehicle* (RV) graph is based on the idea of shareability graphs proposed by Santi et al. [86]. In the RV graph, two requests are connected only if they can be serviced by an empty virtual vehicle starting at the origin of one of them. Likewise, a vehicle is connected to a request if the request can be serviced by the vehicle.

Since we consider previously assigned requests can still be picked up by different vehicles, these requests also integrate the RV graph construction. At each time period t, after updating all vehicle tuples according to the current period t as well as dismissing hired vehicles with expired contract deadlines, we define the set of assigned requests $P^{\mathrm{A}} = \{i \mid \forall r_i \bigcup_{k \in K} \mathscr{R}^k\}$, and the set of all requests $P_t = B_t \cup P^{\mathrm{A}}$. Then, we use the total fleet K and requests P_t to build the RV graph.

Figure 3.3 presents the RV graph created from the problem input described in Figure 3.1(a). The graph features six request-request (RR) pairs and seven request-vehicle (RV) pairs.

3.4.2 Extended trip-request-vehicle graph

Similarly to Alonso-Mora et al. [2], we use the RV graph to compute the *request-trip-vehicle* (RTV) graph, where request and vehicle nodes are connected to trip nodes, which are comprised of request sets. Feasible trips featuring $q \in \{1,2\}$ request derive directly from the RV graph, whereas trips involving $q \geq 3$ requests are feasible only if all possible subtrips with $q-1$ requests are also feasible. By carrying out this sub-feasibility assessment process iteratively, increasingly higher q-trip combinations can be evaluated up to the capacity of vehicle k, that is, $q \in \{1,2,\ldots,Q^k\}$.

In contrast to the original RTV graph formulation, where trip nodes consist of a request set that refers to a unique minimum waiting visiting plan, we propose visiting plan nodes (v-nodes). We consider a single request set $\mathscr{R}$ can lead to up to $|C|+1$ visiting plans. Besides the minimum total delay plan v_g^k, we also consider the minimum total delay plans v_c^k for each class $c \in \{c_i \mid r_i \in \mathscr{R}\}$. Adding extra visiting plans focusing on SQ classes' total delay minimization adds flexibility to the assignment algorithm once it can consider a larger pool of itineraries to balance service level provision between classes.

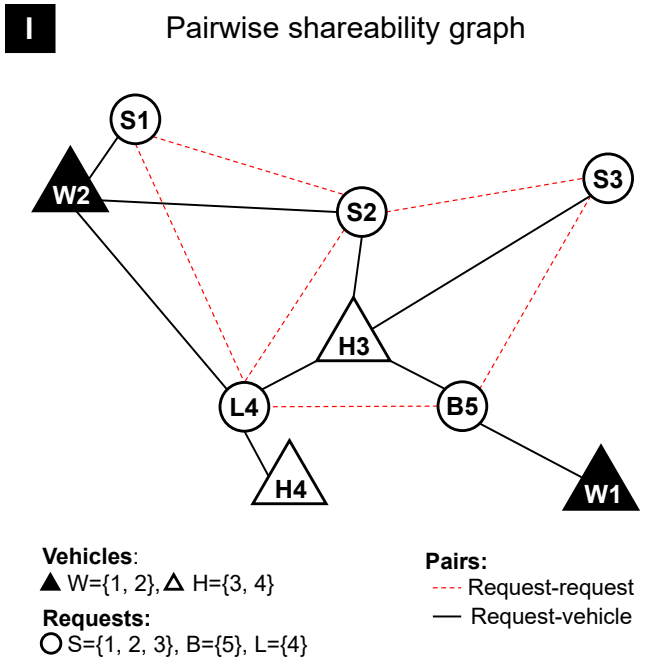


Figure 3.3: *RV graph resulting from the problem input presented in Figure 3.1(a).*

We refer to this expanded version, where request sets generate several visiting plans, as the *extended trip-request-vehicle* (ERTV) graph. Hence, whenever a vehicle k can successfully service a request set $\mathscr{R}$, instead of adding edges $e(r_i, \mathscr{R})\ \forall r_i \in \mathscr{R}$ and $e(\mathscr{R}, k)$, we add edges where $\mathscr{R}$ is replaced by visiting plans v.

Once these visiting plans entail a one-to-one relationship to both vehicles and requests, we can attribute weights to edges. As a result, a feasible visiting plan v leads to edges $e(r_i, v, \delta_{iv})\ \forall r_i \in \mathscr{R}$ and $e(v, k, \Delta_v)$, where δ_{iv} is the delay to pick up request r_i through sequence $\mathscr{S}$ and Δ_v is the total waiting time to realize $\mathscr{S}$.

We employ an exhaustive search over all feasible itineraries created from $\mathscr{R}$ to find the visiting plans v_g^k and v_c^k. However, it is worth noting that we are able to do so because we consider low-capacity vehicles with at most four seats. In order to consider high-capacitated vehicles, local improvement methods or exploration heuristics might be necessary to curb computation time.

Additionally, to ensure every vehicle will be assigned to a visiting plan, we add two extra edges. For parked vehicles, we add edge $e(k, v_{\text{idle}}^k, 0)$ to guarantee they can stay parked. Consequently, if v_{idle}^k is assigned to $k \in K_t^{\text{FAV}}$, k did not need to be hired.

For unloaded vehicles cruising to pick up users (i.e., $\mathscr{P}^k = \emptyset$ and $|\mathscr{R}^k| \geq 1$), we add edge $e(k, v_{\text{stop}}^k, 0)$, where $v_{\text{stop}}^k \in \mathscr{V}_R^k$ is a special rebalance case, where $\mathscr{R}^k$ is emptied and the vehicle stops at the closest parking place (the next break point in the shortest path to the current destination). Both v_{idle}^k and v_{stop}^k have zero-valued weights because we aim to avoid unnecessary trips, such that vehicles remain still whenever possible.

It is worth noting that vehicles can also continue to carry out their incumbent visiting plan, which is naturally added to the ERTV graph during the construction phase. Algorithm 3.2 presents all the construction steps of the ERTV graph.

Finally, upon adding edges that link requests i to visiting plans v, we populate the set $\mathcal{V}_i$ of plans where i can be serviced. Then, we define set $\mathcal{V}_i^{\text{SL}} \subseteq \mathcal{V}_i$ where $\delta_{iv} \leq w_{\text{pickup}}^{c_i}$, which consists of the set of plans where request i desired service levels are satisfied.

Figure 3.4 displays a portion of the ERTV graph based on the pairwise RV graph. It shows how requests $P_t = \{1,2,4,5\}$ from period t are associated to vehicles 1, 2, and 4 through visiting plans $\mathcal{V}^1 = \{v_{17}, v_{18}\}$, $\mathcal{V}^2 = \{v_0, v_1, \ldots, v_{14}\}$, and $\mathcal{V}^4 = \{v_{15}, v_{16}\}$, respectively. It is worth noting that vehicles 1, 2, and 4 include their incumbent visiting plans, namely, v_{18}, v_7, and v_{16}. This way, whenever appropriate, vehicles can continue carrying out these plans. Moreover, for vehicle 2, the plans v_{10}, v_{11}, and v_{12} stem from the same request set $\mathcal{R} = \{r_1, r_4, r_2\}$. Since $\mathcal{R}$ features two different SQ classes, v_{10} is the visiting plan achieved seeking to minimize the total delay whereas v_{11} and v_{12} are the visiting plans achieved when the goal is to minimize the delays of standard and low-cost classes. A similar outcome can be found for plans v_4, v_5, and v_6, which all stem from request set $\mathcal{R} = \{r_2, r_4\}$.

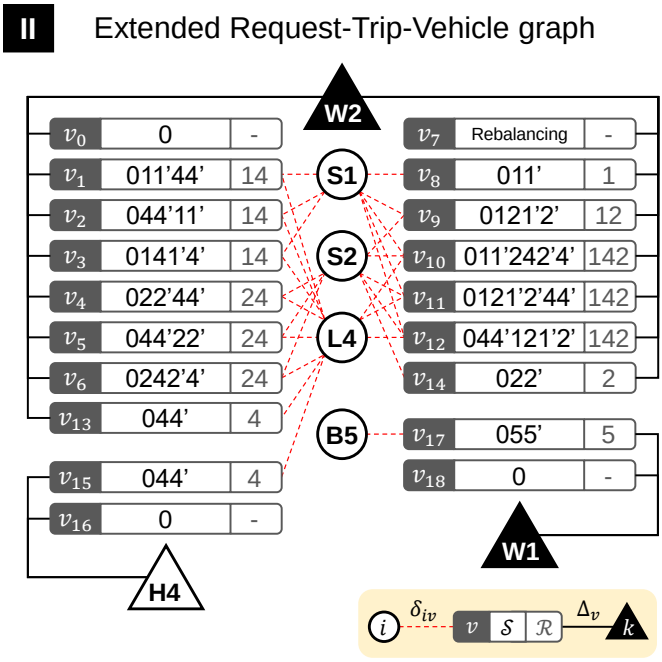


Figure 3.4: *ERTV graph created from the RV graph showed in Figure 3.3. For brevity, we omit the elements concerning vehicle H3. Edges connecting requests i to visiting plans v weight requests' pickup delays δ_{iv}, whereas edges connecting vehicles to visiting plans weight plans' total delays Δ_v (i.e., sum of all users' pickup and in-vehicle delays).*

Algorithm 3.2: ERTV graph construction

Input: RV graph from period t with request-request (RR) edges $e(r_i, r_j)$ with $i, j \in P_t$ and vehicle-request (RV) edges $e(r_i, k)$ with $i \in P_t$ and $k \in K$.
Output: ERTV graph with request and v-node edges $e(r_i, v, \delta_{iv})$ and v-node and vehicle edges $e(v, k, \Delta_v)$.

Function `getVisitingPlans` $(k, \mathscr{R})$
  $V = \emptyset$
  $V \leftarrow v_g^k$ `//Min. total delay` (3.27)
  for $c \in \{c_i \mid r_i \in \mathscr{R}\}$ **do**
    $V \leftarrow v_c^k$ `//Min. total delay for class c` (3.28)
  return V
Function `addVisitingPlansToERTV` (V)
  Add request and v-node edge $e(r_i, v, \delta_{iv})\ \forall r_i \in \mathscr{R}, \forall v \in V$
  Add v-node and vehicle edge $e(v, k, \Delta_v)\ \forall v \in V$
Function `addFeasiblePlansFromRequests` (R_q^k, k, R')
  for $\mathscr{R} \in R'$ **do**
    $V =$ `getVisitingPlans` $(k, \mathscr{R})$
    if $V \neq \emptyset$ **then**
      $R_q^k \leftarrow \mathscr{R}$
      `addVisitingPlansToERTV` (V)
begin
  for $k \in K$ **do**
    `addVisitingPlansToERTV` $(\{v_{\text{stop}}^k, v_{\text{idle}}^k\})$
    $R_q^k = \emptyset\ \forall q \in \{1, 2, \ldots, Q^k\}$
    $R' = \{\{r_i\} \mid \forall e(r_i, k) \in \text{RV graph}\}$ `//Candidate one-request trips`
    `addFeasiblePlansFromRequests` (R_1^k, k, R')
    $R' = \{\{r_i, r_j\} \mid \forall r_i, r_j \in R_1^k\}$ `//Candidate two-request trips`
    $R' = \{\mathscr{R} \mid \forall \mathscr{R} \in R' \wedge e(\mathscr{R}_1, \mathscr{R}_2) \in \text{RV graph}\}$ `//Filter unfeasible`
    `addFeasiblePlansFromRequests` (R_2^k, k, R')
    for $q \in \{3, \ldots, Q^k\}$ **do**
      $R' = \{\mathscr{R}_i \cup \mathscr{R}_j \mid \forall \mathscr{R}_i, \mathscr{R}_j \in R_{q-1}^k\}$
      $R' = \{\mathscr{R} \mid \forall \mathscr{R} \in R' \wedge |\mathscr{R}| = q\}$ `//Select candidate q-request trips`
      $R' = \{\mathscr{R} \mid \forall \mathscr{R} \in R' \wedge \forall r_i \in \mathscr{R}, \mathscr{R} \setminus r_i \in R_{q-1}^k\}$ `//Filter unfeasible`
      `addFeasiblePlansFromRequests` (R_q^k, k, R')

Table 3.4 presents the number of pickup and delivery permutations evaluated at each iteration of the Algorithm 3.2, considering vehicles can carry up to four passengers. Since some permutations might violate requests' service-level requirements, only a subset of them ends up becoming valid visiting plans.

Table 3.4: *Feasible request combinations for each vehicle throughout the iterations of Algorithm 3.2. Valid combinations are created based on the RV graph showed in Figure 3.3 considering four-seat vehicles, totaling 135 pickup and delivery combinations.*

	N. of requests (N. of possible visiting plans)				
Vehicle	1(1)	2(6)	3(90)	4(2,520)	Total
W1	B5		-	-	1
W2	S1	S1,S2	-	-	7
	S2	S2,L4	-	-	7
	L4	S1,L4	S1,S2,L4	-	97
H3	S2	S2,S3	-	-	7
	S3	S3,B5	-	-	4*
	L4	S2,L4	-	-	7
	B5	L4,B5	-	-	4*
H4	L4	L4	-	-	1

*No ridesharing in itineraries featuring business users

3.4.3 Visiting plan assignment formulation

For each period t, we use the ERTV graph elements to formulate an assignment problem, formalized using the following MILP model:

Minimize:

$$f(x_v, y_i, z_i, h^k)$$

Subject to:

$$\sum_{v \in \mathcal{V}^k} x_v = 1 \qquad \forall k \in K \tag{3.34}$$

$$\sum_{v \in \mathcal{V}_i} x_v + z_i = 1 \qquad \forall i \in B_t \tag{3.35}$$

$$\sum_{v \in \mathcal{V}_i} x_v = 1 \qquad \forall i \in P^{\mathrm{A}} \tag{3.36}$$

$$\sum_{v \in \mathcal{V}_S^k} x_v = h^k \qquad \forall k \in K_t^{\mathrm{FAV}} \tag{3.37}$$

$$\sum_{v \in \mathcal{V}_i^{\mathrm{SL}}} x_v = y_i \qquad \forall i \in P_t \tag{3.38}$$

$$\sum_{i \in P_t^c} y_i \geq \left\lceil \sigma^c \cdot |P_t^c| \right\rceil \qquad \forall c \in C \tag{3.39}$$

The optimization problem aims at minimizing the multi-objective function (3.33) in order to find the least number of hirings necessary to meet user service level expectations throughout SQ classes. Constraints (3.34) guarantee that each vehicle realizes a single visiting plan. In turn, constraints (3.35) ensure that every unassigned user is either assigned or rejected, whereas (3.36) guarantee that previously assigned requests continue so but allows the original plan to be changed (other vehicles can carry out the service). Constraints (3.37), guarantee the consistency of the hiring process, such that $h^k = 1$ whenever a visiting plan from $\mathcal{V}_S^k$ of vehicle $k \in K_t^{\text{FAV}}$ is chosen. Constraints (3.38) guarantee the consistency of each variable y_i, which is 1 only when the expected service level of request i is fulfilled. Finally, analogously to the static formulation, constraints (3.39) enforce that the service-level expectations of a minimum share of users from each class are met.

Figure 3.5 illustrates the assignment step for the input example presented in Figure 3.1(a). The feasible visiting plans featured in Figure 3.4 are optimally assigned to associated vehicles according with the multi-objective function (3.33). From the output it can be seen that all user expectations were met (i.e., no delays or rejections). Additionally, previously rebalancing vehicle 2 was assigned to pick up users 2 and 4 whereas vehicle 3 was hired to pick up requests 3 and 5.

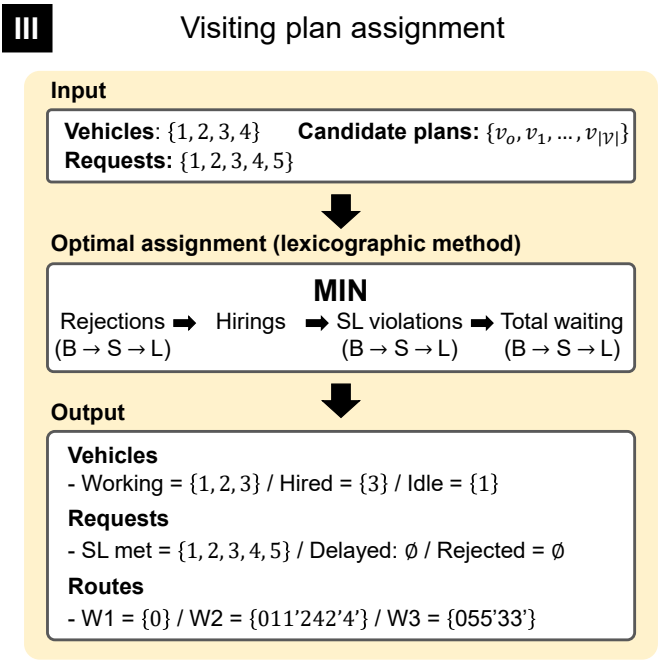


Figure 3.5: *At each period, feasible visiting plans are optimally assigned to available vehicles such that rejections, hirings, service level violations, and waiting times are hierarchically minimized. The order objectives are addressed following the lexicographic method is indicated by the arrows.*

Feasibility. Due to constraints (3.39), the MILP model is infeasible whenever the number of vehicles is insufficient to honor SQCs. Still, we can determine the best possible outcome by minimizing violations to a relaxed version:

$$\sum_{i \in P_t^c} y_i + a^c \geq \left\lceil \sigma^c \cdot |P_t^c| \right\rceil \qquad \forall c \in C, \tag{3.40}$$

where a^c represents the number of service level violations in class c, such that $0 \leq a^c \leq \left\lceil \sigma^c \cdot |P_t^c| \right\rceil$, $\forall c \in C$. Then, we can minimize a^c across classes using the ordered list of objectives

$$\{f^c_{\text{feasible}}(s) = a^c \mid \forall c \in C\}. \tag{3.41}$$

3.4.4 Idle vehicle rebalancing formulation

Due to the spatiotemporal characteristics of transportation requests, vehicles can end up stranded in low demand areas of a city, operating at subpar productivity rates. Hence, to address ongoing supply-demand imbalances, relocation strategies are commonly employed to route idle vehicles to regions where requests are more prone to appear. Following Alonso-Mora et al., we fix supply and demand imbalances by optimally rebalancing parked vehicles $K_t^{\text{P}} = \{k \mid \text{if } \nu^k = \nu^k_{\text{idle}} \forall k \in K\}$ to locations where the system failed to meet user needs. In the original formulation, vehicle rebalancing targets were the pickup points of rejected users. Since our hiring setup can avoid rejections altogether, we adopt two new rebalancing stimuli to move vehicles throughout the street network. First, since hiring indicates a lack of proper vehicle supply, we consider that the origins of hired vehicles $K_t^{\text{H}} = \{k \mid h^k = 1 \ \forall k \in K_t^{\text{FAV}}\}$ are also rebalancing targets. We refer to these origins as $O_t^* = \{o^k \mid \forall k \in K_t^{\text{H}}\}$, since they are a subset of regional centers Z^*. Second, we consider that service-level violations in a particular area also indicate insufficient vehicle supply. This way, besides the origins of rejected users, the origins of assigned but dissatisfied users also integrate the list of rebalancing targets. We refer to the origins of service-level violated requests as $P_t^{\text{U}} = \{i \mid y_i = 0,\ r_i \in P_t\}$, such that the list of rebalancing targets is $J_t = \{i \mid \forall\ i \in O_t^* \cup P_t^{\text{U}}\}$. Finally, we minimize the total travel times to reach the rebalancing targets J_t.

For each vehicle $k \in K_t^{\text{P}}$ we define a set of candidate visits $\mathcal{V}^k = \{\nu^k_{\text{idle}}\} \cup \mathcal{V}_R^k$, where visiting plans $\nu^k \in \mathcal{V}_R^k$ are based on sequences $\{S^k \mid S^k = \{S_0^k, i\}\ \forall i \in J_t\}$. Then, we use variables x_ν to optimally assign vehicles to targets as follows:

Minimize:

$$\sum_{k \in K_t^{\text{P}}} \sum_{\nu \in \mathcal{V}_R^k} \Delta_\nu x_\nu \tag{3.42}$$

Subject to:

$$\sum_{\nu \in \mathcal{V}^k} x_\nu = 1 \qquad \forall k \in K_t^{\text{P}} \tag{3.43}$$

$$\sum_{k \in K_t^{\mathrm{P}}} \sum_{v \in \mathcal{V}_R^k} x_v = \min\left(|J_t|, |K_t^{\mathrm{P}}|\right) \tag{3.44}$$

The objective function (3.42) aims at minimizing the total rebalancing delay. Equations (3.43) guarantee that all available idle vehicles K_t^{P} are used to reach targets in J_t (if any), and equations (3.44) ensure that each idle vehicle ends up either rebalancing or staying parked. Figure 3.6 illustrates how the rebalancing process takes place for the idle vehicles resulting from the assignment step illustrated in Figure 3.5.

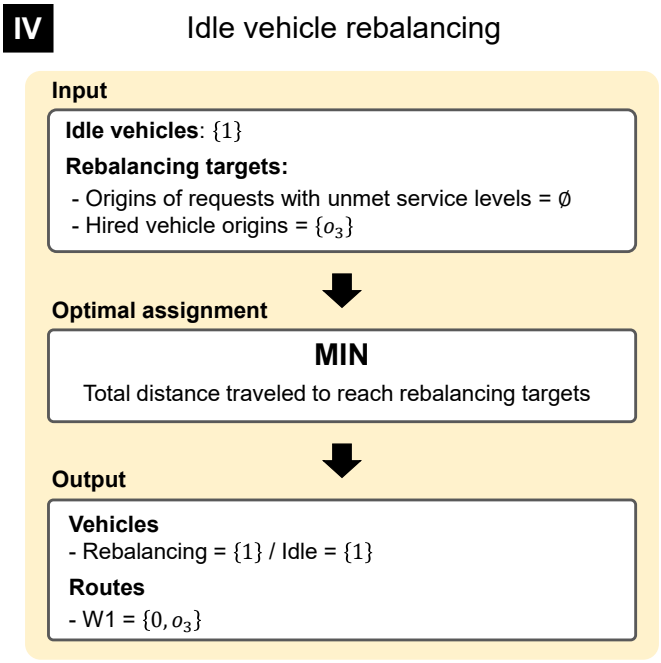


Figure 3.6: *Idle working vehicles are optimally rebalanced to underserved areas.*

3.5 Experimental study

In the following sections, we present the SQ parameters (Section 3.5.1), describe how simulation settings such as trip and travel data are set up (Section 3.5.2), and design twelve operational scenarios (Section 3.5.3) to study the influence of different user-base and FAV-supply configurations.

3.5.1 Service quality settings

The primary objective of our experiments is to assess the operational impact of actively controlling service quality for a heterogeneous user base. Since the interplay between service levels, fleet size, and service denial has already been thoroughly examined in the

literature (see, e.g., [67]), we focus our analysis on the operational adjustments required to service the request demand entirely, such that formerly inconvenienced users can also be successfully serviced under strict SQCs.

In order to design a heterogeneous user base, we first define the class SQC configurations adopted in this chapter. Next, to evaluate the outcome of meeting SQCs, we present three service-level enforcement rate scenarios, where we vary provider's leeway to violate the SQCs. Finally, to evaluate how class SQC configurations can influence fleet operations, we propose three user segmentation scenarios, where we vary the predominance of a certain class within the transportation demand.

Service quality contracts

Table 3.5 summarizes the SQC settings we adopt for each user class. Besides order of priority and sharing preferences, it shows the expected maximum pickup and total delays of each SQ class. The class-specific settings can represent, for instance, high-, intermediate-, and low-SQ requirements, possibly arising from high-, middle-, and low-income users, respectively. Besides income, the criteria to choose a suitable class can also be related to the nature of the appointment of a user. For example, trips to scheduled events (e.g., meetings, concerts, doctor appointments) are usually stricter, requiring higher service rates and levels. Alternatively, to make a parallel with current urban mobility options (regarding the expected quality of service), the low-cost class best represents former transit passengers, whereas standard and business classes best represent former ridesharing and carsharing/taxi passengers, respectively.

Table 3.5: *Service quality contract (SQC) for each SQ class.*

SQ class	Priority	Ride preference	Service level (min)	
			Expected max. pickup (w_{pickup})	Max. total delay ($w_{\text{tolerance}}$)
Business	1st	private	3	7
Standard	2nd	shared	5	7
Low-cost	3rd	shared	7	7

Service-level enforcement rate

We assess the outcome of different service-level enforcement rates σ^c through four scenarios where we assume the system addresses all SQ classes $c \in C$ using a common rate $\sigma \in \{0, 0.8, 0.9, 1\}$. The scenario where $\sigma = 0$ simulates the typical AMoD system in which service-level preferences are not enforced. The remainder scenarios allow us to investigate how increasingly enforcing the fulfillment of service-level expectations in 10% steps across user classes affects hiring, until the point at which service-level expectations are fully upheld, for $\sigma = 1$. Hence, scenarios $\sigma = 0$ and $\sigma = 1$ represent lower and upper service-level bounds, whereas $\sigma = 0.8$ and $\sigma = 0.9$ allow the system to violate service-level expectations of 20% and 10% of the requests, respectively. Although we do not consider costs in our model, the service-level enforcement rates are tunable parameters that allow providers to trade off vehicle hiring and user dissatisfaction costs.

User base segmentation

To investigate the interplay between requests of distinct SQ classes, we create three user base segmentation scenarios, namely, B+, S+, and L+, in which we vary the proportion of users belonging to each SQ class in the transportation demand (Table 3.6). We assume that the service quality demanded by each user base follows a normal distribution in which the predominant class of such base covers the average SQ requirements of most users (68%) whereas the other two classes can adequately service the rest.

The proposed scenarios aim to represent service quality requirements arising in different regions, on different occasions, or at different times. For example, user base L+ may better represent the vicinity of a campus area, where most users (presumably students) are willing to wait longer in exchange for cheaper rides. Conversely, the user base B+ fits affluent areas where privacy and high responsiveness are prone to play a more significant role. Throughout the simulation, we use the user base segmentation scenarios to determine the share of requests from the New York City taxi dataset that belongs to each SQ class.

Table 3.6: *User base segmentation scenarios define different distributions of SQ classes across the same transportation demand by varying, in turn, the proportion of requests in each class.*

User base	SQ class		
	Business	Standard	Low-cost
B+	68%	16%	16%
S+	16%	68%	16%
L+	16%	16%	68%

3.5.2 Simulation settings

Street network. We use a multidigraph from the street network of Manhattan, New York City comprised of 4,548 locations and 9,701 links. Travel times are drawn from the shortest distances between the street network locations, considering an average travel speed of 30km/h. For simplicity, we assume that congestion is an exogenous effect; neither the time of the day nor the number of vehicles traveling throughout the street network affects travel times.

Region center distribution. Figure 3.7 shows the distribution of 68 regional centers, optimally determined considering a maximal hiring delay $s = 150$s. This value allows that available hireable vehicles can fulfill the expectations of even business users who have placed their request at the beginning of a thirty-second round.

Transportation demand settings. We run our simulation on 42,702 real-world taxi requests from Manhattan occurring in the evening peak, from 18h to 19h, on the first day of February 2011. The raw dataset, containing detailed taxi information for the whole city, is processed in three phases. First, we filter out the demands occurring outside

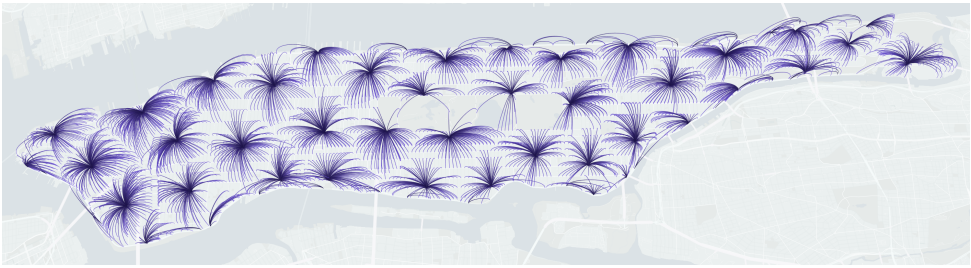


Figure 3.7: *Optimal distribution of regional centers with corresponding reachable locations throughout the street map of Manhattan, New York City considering a maximal hiring delay of 150s at 30km/h.*

Manhattan. To do so, we remove all requests whose origin/destination GPS coordinates cannot be matched (within a 50 meters range) to an intersection of such graph or whose travel distances are small (below 50 meters). After the matching process, we replace origin and destination GPS locations with their correspondent node IDs in the graph. Second, we select only the relevant trip data fields required to carry out our simulation, namely, pickup times (which are adapted to request times), origin/destination IDs, and number of passengers. Additionally, we filter out records whose number of passengers is higher than four (i.e., the maximum vehicle capacity we consider). After this step, the final shares of requests requiring one, two, three, and four seats are 77.00%, 16.54%, 4.53%, and 1.93%, respectively. Finally, we use the roulette wheel selection to randomly attribute SQ classes to users, following the proportions specified in the user base segmentation scenarios.

3.5.3 Case study configuration

We combine both service-level enforcement rates {0, 0.8, 0.9, 1} and user base segmentation scenarios {B+, S+, L+} presented in Section 3.5.1 to create a series of case studies for the static and dynamic formulations. Regardless of the case study considered, users expect to be serviced according to the SQC configuration presented in Table 3.5. Table 3.7 summarizes the settings used to create the instances for both formulations.

Static DARP-SQC

Each instance is a combination of a request batch of size $n \in \{15, 20\}$ and a user base segmentation $u \in \{\text{B+, S+, L+}\}$. To create such instances, we collect n requests from thirty-second batches of our trip data and randomly distribute SQ classes according to the proportions defined in the user base u. We maintain requests' original placement times as well as pickup and delivery locations but set all passenger counts to one to enable more ridesharing opportunities.

Since we assume a no-rejection policy, we adjust the fleet size and the disposition of the vehicles according to the settings of each instance. First, to guarantee all requests can be reached by a vehicle on time, we determine the minimum set of regional centers from which vehicles can access all nodes in less than three minutes (i.e., the shortest expected maximum pickup delay of all SQ classes). Then, for each request, we assume there will

be a four-seat vehicle, available throughout the entire planning horizon, stationed at a regional center that can adequately reach it, such that $|K| = n$. Although our model allows for vehicle heterogeneity, we do not investigate such a feature in this formulation to curb computational complexity. Creating a viable instance for a heterogeneous fleet would require stationing at least $Q_{\max} \times n$ vehicles at each regional center, assuming vehicle capacities can range from one to $Q_{\max} = \max\{Q^k \mid \forall k \in K\}$.

Dynamic DARP-SQC

We assume that the provider relies on an initial working fleet of 1,000 four-seat vehicles to service the transportation demand described in 3.5.2. If the policy allows, the provider can inflate the fleet size by occasionally hiring privately-owned vehicles (on a single-ride basis) to meet minimum class service levels requirements. By default, we assume the capacity of the hired vehicles is equal to the number of passengers of the request that first prompted the hiring.

Table 3.7: *Instance settings for both static and dynamic problem formulations.*

	Problem formulation	
Characteristic	Static	Dynamic
#Requests	{15, 20}	42,702
#Instances	5 request samples	-
Round duration (κ)	30s	30s
#Rounds	1	120
Company fleet size	-	1,000
Capacity of company vehicle	-	4 seats
Hireable fleet size	Equal to n. of requests	Unlimited
Capacity of hireable vehicle	4 seats	1, 2, 3, or 4 seats

3.5.4 Dynamic formulation benchmarking

We implement three policies by varying the objective function of our matching algorithm:

Min. waiting (MW): Standard formulation, with no service level constraints and no hiring, where the goal is to hierarchically minimize the rejection penalties and waiting times across classes $f^{\text{MW}} = \{f^c_{\text{reject}}\ \forall c \in C,\ f^c_{\text{wait}}\ \forall c \in C\}$:

$$\text{Min. } f^{\text{MW}}(x_v, z_i) \text{ s.t. (3.34), (3.35), (3.36).}$$

SL constraints (SL): No hiring formulation, enforcing service-level constraints, and aiming to hierarchically minimize $f^{\text{SL}} = \{f^c_{\text{feasible}}\ \forall c \in C,\ f^c_{\text{reject}}\ \forall c \in C, f^c_{\text{violate}}\ \forall c \in C,\ f^c_{\text{wait}}\ \forall c \in C\}$:

$$\text{Min. } f^{SL}(x_v, y_i, z_i, a^c) \text{ s.t. (3.34), (3.35), (3.36), (3.38), (3.39), (3.40).}$$

SL constraints + Hire (SLH): Proposed formulation, enforcing service-level constraints, allowing hiring, and aiming to hierarchically minimize $f^{\text{SLH}} = \{f^c_{\text{feasible}}\ \forall c \in C, f^c_{\text{reject}}\ \forall c \in C, f_{\text{hire}}, f^c_{\text{violate}}\ \forall c \in C,\ f^c_{\text{wait}}\ \forall c \in C\}$:

Min. $f^{\text{SLH}}(x_v, y_i, z_i, h^k, a^c)$ s.t. (3.34), (3.35), (3.36), (3.37), (3.38), (3.39), (3.40).

Although we assume providers are allowed to break service level expectations of up to $(1-\sigma^c)$ percent of users for each class $c \in C$, we consider that rejections are a more critical service-level violation than delays. Following this assumption, throughout all policies we first seek to minimize rejections. Subsequently, policy MW focuses on minimizing waiting times, whereas policies SL and SLH focus on minimizing service-level violations, which cover both rejections and pickup delays. Particularly in SLH, we use the hiring capabilities to prevent rejections from happening even further, once avoiding user rejection is more important than fleet size minimization. Ultimately, by contrasting MW, SL, and SLH, we can assess the effect of enforcing service-level constraints alone as well as how they influence vehicle hiring.

3.6 Results

The static and dynamic formulations were developed in Python and Java, respectively, and all MILPs were implemented using Gurobi 8.1.0. The experiments were performed on an AMD Opteron CPU running at 2.10 GHz and 128 GB RAM.

3.6.1 Static DARP-SQC

Table 3.8 presents the results for the static DARP-SQC model. Each instance is run for a maximum of five hours. The figures for each number of requests, user base, and service rate combination are obtained by averaging the results of the five different request distributions. In column "N. of hired," we present the average number of hired vehicles, considering that the fleet size equals the number of requests. Next, for each SQ class, we show the pickup and ride delays. Finally, column "N. of solved inst." presents the number of instances that could be solved optimally. We use these instances to calculate the averages.

The fleet sizes achieved under the no service level enforcement scenario (i.e., $\sigma = 0$) represent the minimum number of vehicles required to pick up all users. In contrast, the slightly higher fleet sizes achieved under the 80%, 90%, and 100% service rates highlight the compromise between the number of vehicles and the class delay tolerances enforced by the SQC constraints. From an AMoD platform's perspective, these results illustrate how exploring users' tolerance to extra delays effectively decreases fleet size while maintaining the strict service quality imposed by previously laid out agreements. By setting up balanced SQCs, platforms can avoid excessive hiring by exploiting the delay tolerance of specific classes while guaranteeing maximum performance for high-demanding user classes.

Table 3.8: *Number of vehicles hired, pickup delay, and ride delay for the static DARP-SQC instances across all case studies. Figures correspond to the average of the results achieved for instances that could be solved to optimality.*

N. of req.	User base	Service rate	N. of hired	Pickup delay (s)			Ride delay (s)			N. of solved inst.
				B	S	L	B	S	L	
15	B+	0%	10.4	138.3	175.6	203.5	0.0	22.9	12.9	5
		80%	11.8	90.9	134.2	221.6	0.0	48.4	12.9	5
		90%	12.2	90.5	109.1	220.5	0.0	48.4	12.9	5
		100%	12.2	90.5	109.1	220.5	0.0	48.4	12.9	5
	S+	0%	9.0	57.6	143.3	176.3	0.0	46.8	57.0	4
		80%	9.2	57.6	126.0	175.0	0.0	60.7	56.6	4
		90%	9.2	57.6	126.0	175.0	0.0	60.7	56.6	4
		100%	9.8	57.6	115.0	175.0	0.0	49.8	56.6	4
	L+	0%	9.0	68.8	152.3	165.1	0.0	0.0	67.3	4
		80%	9.2	59.2	147.9	163.6	0.0	0.0	67.3	4
		90%	9.5	59.2	124.6	163.6	0.0	0.0	60.8	4
		100%	9.5	59.2	124.6	162.4	0.0	0.0	61.9	4
20	B+	0%	14.0	124.2	227.8	205.6	0.0	28.5	8.1	5
		80%	16.8	73.1	162.0	223.5	0.0	28.5	12.3	5
		90%	17.2	73.1	134.4	199.2	0.0	6.0	11.5	5
		100%	17.2	73.1	134.4	199.2	0.0	6.0	11.5	5
	S+	0%	11.0	170.0	199.7	159.1	0.0	25.9	0.2	2
		80%	12.5	94.6	193.0	184.7	0.0	23.9	30.8	2
		90%	13.5	76.0	166.7	174.8	0.0	14.5	7.2	2
		100%	14.5	74.8	141.0	173.8	0.0	14.5	7.2	2
	L+	0%	10.0	116.0	201.7	181.7	0.0	42.3	49.7	1
		80%	10.0	116.0	103.3	197.7	0.0	151.3	39.4	1
		90%	10.0	116.0	103.3	197.7	0.0	151.3	39.4	1
		100%	11.0	116.0	69.2	197.7	0.0	133.2	35.2	1

From Table 3.8, we can also see that, even for a small twenty-request batch, optimal solutions could be found only for a few instances from user bases S+ and L+. Conversely, the predominance of business-class requests in the user base B+ allows reaching optimal solutions in every case. Having more standard and low-cost class users, who are willing to share a ride and wait longer, leads to a larger number of possible routes, which translates to additional complexity. Ultimately, the results indicate that addressing realistic instances for the DARP-SQC requires more computationally efficient algorithms.

3.6.2 Dynamic DARP-SQC

Throughout the following sections, we provide an exploratory analysis of the aggregate results regarding all user bases and service-level enforcement rates for all policies. We use the case study consisting of user base S+ and service-level enforcement rate $\sigma = 0.9$ to illustrate our results further and refer to it as the reference case study. In this case study, the most frequent SQ class (i.e., the standard class) imposes a balanced SQC when compared to its counterparts, requiring reasonably fast service levels and also allowing for ridesharing. On the other hand, the reference service-level enforcement rate allows us to thoroughly assess the flexibility achieved by violating the expectations of 10% of the users.

Service quality distribution

Table 3.9 summarizes the average service quality outcome across classes achieved by each policy for all case studies. The column "Serviced" shows the percentage of users picked up by the AMoD system. In turn, the column "Met SL" presents the percentage of satisfied users, serviced according to their expected service levels. Subsumed under the header "Violated SL", we present the two service level violations we consider in our approach. The column "Delayed" indicates the percentage of users whose desired service levels could not be fulfilled, whereas the column "Rejected" indicates the percentage of requests that could not be serviced. It is worth noting that each percentage under the "Serviced" column is the sum of corresponding percentages under "Met SL" and "Delayed" columns. The user base segmentation scenarios play a significant role in the number of users serviced when vehicle hire is not enabled. Once user base scenario B+ comprises more business users, which expect shorter delays and private rides, less sharing occurs, leading to about 10% fewer pickups than the other bases. However, by exploiting the user delay tolerances, our SL policy can achieve higher service rates than the MW policy across case studies. By delaying users who can wait longer, the SL-policy matching process can enable additional possibilities to combine overlapping routes. Although policy MW minimizes user class rejections and waiting times hierarchically, the lack of service-level enforcement constraints prevents the optimization process from adequately harnessing class delay tolerances.

Table 3.9: *Service quality achieved using each policy for all case studies.*

User base	Service rate	Policy	Serviced	Met SL	Violated SL	
					Delayed	Rejected
B+	0%	Min. waiting	84.99%	37.35%	47.64%	15.01%
	80%	SL constrs.	85.09%	55.32%	29.77%	14.91%
		SL constrs. + Hire	100.00%	91.98%	8.02%	-
	90%	SL constrs.	85.18%	56.01%	29.17%	14.82%
		SL constrs. + Hire	100.00%	95.91%	4.09%	-
	100%	SL constrs.	84.92%	56.84%	28.08%	15.08%
		SL constrs. + Hire	100.00%	100.00%	-	-
S+	0%	Min. waiting	96.14%	55.92%	40.22%	3.86%
	80%	SL constrs.	98.19%	85.58%	12.61%	1.81%
		SL constrs. + Hire	100.00%	91.05%	8.95%	-
	90%	SL constrs.	98.16%	85.73%	12.43%	1.84%
		SL constrs. + Hire	100.00%	95.58%	4.42%	-
	100%	SL constrs.	98.37%	86.69%	11.68%	1.63%
		SL constrs. + Hire	100.00%	100.00%	-	-
L+	0%	Min. waiting	96.53%	78.53%	18.00%	3.47%
	80%	SL constrs.	98.00%	93.52%	4.48%	2.00%
		SL constrs. + Hire	99.34%	96.14%	3.20%	0.66%
	90%	SL constrs.	97.79%	94.07%	3.72%	2.21%
		SL constrs. + Hire	99.63%	98.05%	1.59%	0.37%
	100%	SL constrs.	97.76%	94.41%	3.35%	2.24%
		SL constrs. + Hire	100.00%	100.00%	-	-

For example, for the reference case study, enforcing service-level constraints leads to a 29.81% percentage point increase over the 55.92% number of satisfied users (i.e., met service levels) from policy MW. At the same time, from Table 3.9, we can see that the number o serviced users increases moderately (from 96.14% to 98.16%). Ultimately, these results suggest that our service-level enforcement constraints were useful to reorganize pickups, resulting in a 53.3% increase in service quality across classes in the reference case study.

Figures 3.8 and 3.9, detail the service-level distribution for all requests, considering user base S+ and service-level enforcement rate $\sigma = 0.9$. In Figure 3.8, we can see that because policy MW does not enforce users' class service-level expectations, pickup waiting times of business users end up being the highest among the classes. Once we aim to first minimize rejections across classes in MW, this result suggests that by delaying the business users, the optimization process could pick up more users from the remaining classes. Figure 3.8 also shows that policy SL rejects thirty four business users (0.4% of the 6,831 business requests) but is able to achieve the expected three-minute maximum waiting time for around 75% of them.

Service-level enforcement rate versus vehicle hiring. Figure 3.9 offers an alternative perspective on how each policy makes use of the 10% service level violation allowed by adopting the service-level enforcement rate $\sigma = 0.9$. For SL, which features the relaxed version of the service-level enforcement constraints (3.40), we can verify how this relaxation took place across classes. For business and standard users, the share of SL violated requests are 11.24% and 17.73%, which correspond to 1.24 and 7.73 percentage points higher than the 10% violation limit. In contrast, for the SLH policy, the violations for business and standard users total 3.73% and 5.60%, figures significantly lower than the SL violation limit. This result could indicate that the optimization process is overhiring vehicles since there was still extra room for violating user expectations (up to 10%). However, by analyzing which user class hired vehicles are addressing the most, we can further understand why the service violation limit was underutilized.

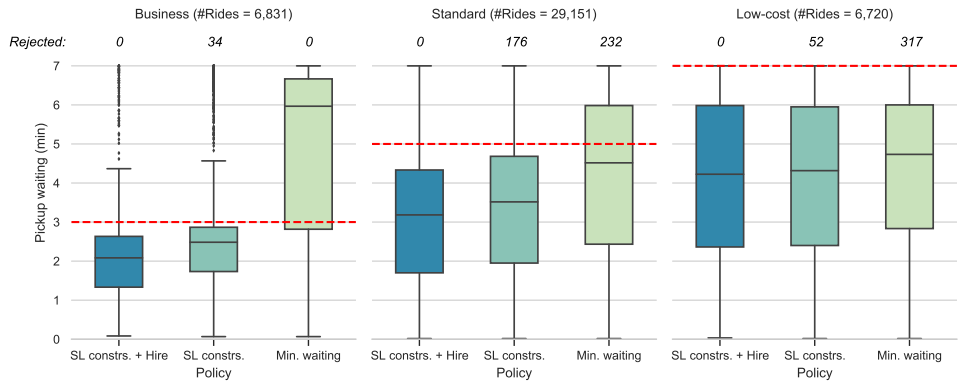


Figure 3.8: *Pickup waiting time distribution and number of rejected users for each SQ class across all policies for user base S+ and service-level enforcement rate $\sigma = 0.9$ for policies SL and SLH. The dashed lines mark the expected max. pickup delays of user classes.*

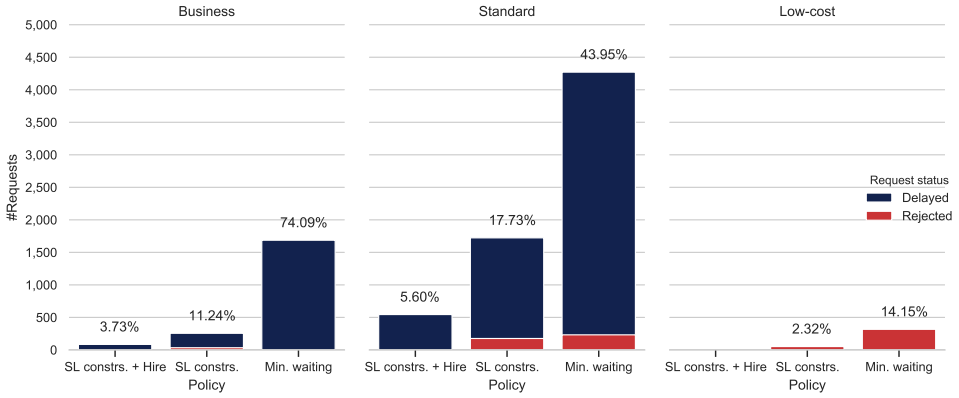


Figure 3.9: *Service-level violation breakdown into percentages of rejected and delayed users for each SQ class across all policies for user base S+ and service-level enforcement rate $\sigma = 0.9$ for policies SL and SLH.*

Table 3.10 expands the summary statistics presented in Table 3.9. For each user base segmentation scenario, it shows for every user SQ class, the percentage of users picked up by company or freelance vehicles as well as the percentage of users whose service-level expectations are violated. From Table 3.10, we can see that vehicle hiring occurs predominantly to fulfill business user requests. Therefore, these hirings end up freeing the four-seat company vehicles that would have been used to transport these requests privately. As a result, the ridesharing requests from standard and low-cost classes can enjoy a larger vehicle supply, which leads to fewer service-level violations. This extra vehicle supply, in turn, is reflected in the underutilization of the service-level violation limits.

Long-term fairness. Ultimately, it is worth noting that, although we enforce class service levels, individual users may experience service-level violations repeatedly, over a longer time horizon. To further minimize user dissatisfaction, a real-world service provider could strive to make up for repeating users who have had their service levels violated in previous rides. Provided that users $i \in P$ are associated with unique ids in the AMoD system, the variable of y_i can be set up ad-hoc to grant users the best service level. This way, σ^c rates could also be applied individually by keeping track of each traveler's service level over multiple rides.

The effect of rebalancing. Regarding the rebalancing approach, the inclusion of service-level violations as an additional stimulus to prompt rebalancing (besides service rejection) has been shown to increase fleet productivity. It can be seen from Figure 3.10 that in MW, idle vehicles can still be found until about 18:30, whereas in SL there are no idle vehicles from about 18:10 and onward. The number of idle vehicles drops sharply as soon as the system fails to fulfill user expectations across classes. As a result, by rebalancing vehicles earlier, policy SL contributes to achieving a two percentage point increase in the number of users picked up compared to MW. Moreover, when users cannot be rejected (i.e., policy SLH), the hiring operations have successfully replaced the user rejection stimuli to indicate where vehicles are needed.

Table 3.10: *Share of requests whose service-level expectations were met (through company or freelance vehicles) or violated, according to SQ class for each user base distribution, service-level enforcement rate, and policy.*

User base	Service rate	Policy	Business			Standard			Low-cost		
			Company	Freelance	Violated	Company	Freelance	Violated	Company	Freelance	Violated
B+	0%	Min. waiting	14.6%	-	85.4%	73.3%	-	26.7%	98.3%	-	1.7%
	80%	SL constrs.	38.2%	-	61.8%	87.1%	-	12.9%	96.0%	-	4.0%
		SL constrs. + Hire	60.4%	29.7%	9.9%	92.1%	-	7.9%	100.0%	-	-
	90%	SL constrs.	38.7%	-	61.3%	87.5%	-	12.5%	98.2%	-	1.8%
		SL constrs. + Hire	64.1%	31.2%	4.8%	94.8%	*	5.2%	100.0%	-	-
	100%	SL constrs.	39.3%	-	60.7%	89.4%	-	10.6%	98.7%	-	1.3%
		SL constrs. + Hire	66.8%	33.2%	-	99.4%	*	-	100.0%	-	-
S+	0%	Min. waiting	25.9%	-	74.1%	56.0%	-	44.0%	85.8%	-	14.2%
	80%	SL constrs.	86.0%	-	14.0%	82.9%	-	17.1%	96.7%	-	3.3%
		SL constrs. + Hire	75.4%	16.8%	7.8%	88.4%	*	11.3%	100.0%	-	-
	90%	SL constrs.	88.8%	-	11.2%	82.3%	-	17.7%	97.7%	-	2.3%
		SL constrs. + Hire	78.2%	18.1%	3.7%	92.8%	1.6%	5.6%	100.0%	-	-
	100%	SL constrs.	90.3%	-	9.7%	83.1%	-	16.9%	98.4%	-	1.6%
		SL constrs. + Hire	76.6%	23.4%	-	97.9%	2.1%	-	99.9%	*	-
L+	0%	Min. waiting	25.4%	-	74.6%	61.8%	-	38.2%	94.9%	-	5.1%
	80%	SL constrs.	83.8%	-	16.2%	87.3%	-	12.7%	97.2%	-	2.8%
		SL constrs. + Hire	82.6%	9.0%	8.4%	88.4%	-	11.6%	99.0%	-	*
	90%	SL constrs.	84.4%	-	15.6%	89.0%	-	11.0%	97.5%	-	2.5%
		SL constrs. + Hire	80.3%	15.9%	3.7%	92.6%	1.2%	6.2%	99.5%	-	*
	100%	SL constrs.	85.5%	-	14.5%	90.1%	-	9.9%	97.5%	-	2.5%
		SL constrs. + Hire	77.2%	22.8%	-	95.8%	4.2%	-	99.3%	*	-

Values lower than 0.01% are marked with "*" and absent data are marked with "-"

Fleet size inflation

From the overall fleet statistics presented in Table 3.11, we can see that the average, maximum, and median numbers of hired seats throughout all rounds are proportional to the service quality of each user base and service-level enforcement rate. For instance, we can see that the overrepresentation of business-class users is translated accordingly in the fleet of vehicles hired to service formerly dissatisfied users in policies MW and SL. Moreover, most hired vehicles end up being of low capacity. Once we assume that a hired vehicle's capacity equals the number of seats required by the request that prompted the hiring, this outcome is congruent with the characteristics of the user demand (77% of the requests demand only one seat). For the reference case study, Table 3.11 shows that at the demand peak, the total fleet capacity grows by 168 vehicles (see the hiring peak in Figure 3.10 between 18:15 and 18:30).

Table 3.11: *Average number of hired vehicles per capacity and summary statistics for the hired fleet in each round.*

User base	Service rate	Avg. hired/Vehicle capacity				Hired vehicles		
		1	2	3	4	Avg.	Median	Max.
B+	80%	76.39%	19.41%	2.74%	1.46%	223.88	309	493
	90%	79.67%	15.79%	3.21%	1.32%	243.75	336	497
	100%	79.92%	14.75%	4.16%	1.17%	254.74	355	520
S+	80%	76.41%	15.48%	4.82%	3.29%	32.79	22	102
	90%	73.45%	17.40%	5.45%	3.70%	47.41	43	168
	100%	74.29%	18.57%	4.94%	2.20%	60.32	65	167
L+	80%	66.49%	26.89%	3.36%	3.26%	15.64	8	72
	90%	77.33%	17.55%	3.50%	1.62%	30.85	22	104
	100%	76.71%	18.01%	2.61%	2.67%	56.07	51	191

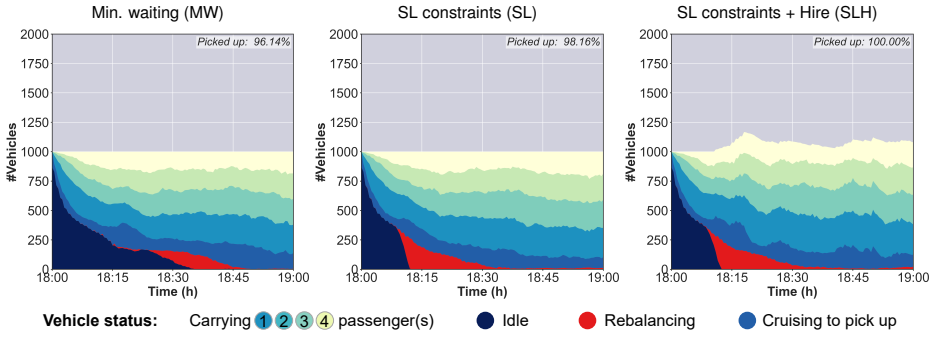


Figure 3.10: *Size of the working fleet every thirty seconds over the course of one hour for each policy, considering user base S+ and a 90% SL enforcement for SL and SLH. Colors help to identify the part-to-whole ratio of vehicles per status at each period.*

Whole day operation

In Figure 3.11, we show the fleet status for a full-day instance considering 210,269 requests configured according to the reference case study. Hirings occur predominantly in the evening, and the surplus number of vehicles over the 1000-vehicle fleet is lower than 200. From the figure, we can conclude that the fixed working fleet can meet the entire demand most of the time and still be sub-utilized at some periods (e.g., before 6:00). Our results stress how much vehicle downtime a provider can avoid by partially relying on the FAV fleet to completely fulfill the demand.

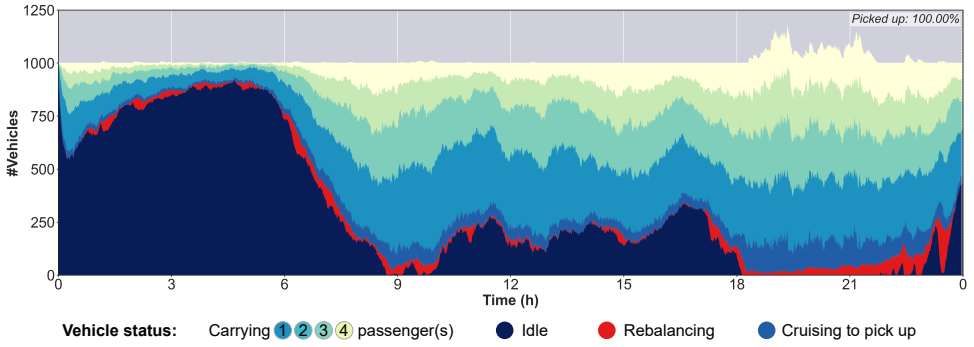


Figure 3.11: *Size of the working fleet every thirty seconds over the course of an entire day, considering user base S+ and a 90% SL enforcement for SLH, policy. Colors help to identify the part-to-whole ratio of vehicles per status at each period.*

Computational complexity

The same shortcomings found for the static version regarding the more flexible user-segmentation scenarios are also present in the dynamic version. In these scenarios, the lexicographical method cannot be completed on some rounds due to the maximum computation time we have imposed on the optimization process. Consequently, some assignments are sub-optimal, covering a subset of the objectives (high-priority first). This outcome is particularly present for scenario L+. From Table 3.11, we can see that the maximum number of hired vehicles for this user-segmentation for a 100% service rate is higher than the S+, where users have a lower waiting tolerance. Additionally, Table 3.9 shows that 0.37% of low-cost users are rejected while using the hiring policy SLH, where hired vehicles are made available to pick up all users. This indicates that, for some rounds, the hierarchical optimization stops prematurely, failing to minimize class rejections and vehicle hire.

3.6.3 Managerial insights

With widespread AV adoption, cities may implement policies that prioritize high occupancy vehicles or even charge empty-seat taxis. Such policies serve two main purposes. First, they discourage excessive low-occupancy AV rides, counterbalancing potential rises in overall *vehicle kilometers traveled* (VKT). Second, they contribute as alter-

native sources of income for cities, especially in the shadow of an expected revenue drop due to less parking space requirements [74]. Hence, keeping high occupancy rates and reducing total VKT are critical factors for future AMoD providers. Our multi-objective function covers both goals once it determines the minimum fleet size and mix to meet the demand expectations.

However, despite advantageous for the operator, single-ride contracts are likely to be unpopular among owners. Under such an agreement, there might be occasions where vehicles are made available for a whole day but make only a couple of trips. In this low-profit scenario, owners could, for instance, link their participation to a minimum compensation, such that their vehicles are used more frequently.

Alternatively, to guarantee frequent use, providers could establish long-duration contracts to actively manage and rebalance vehicles as if they belonged to the fixed working fleet. Under such contracts, however, operators may be required to bear all costs within the leasing period. These may include not only the costs to rebalance and service users but also the opportunity costs of hireable vehicles working for other competing operators. Therefore, once costs are considered, the rider's satisfaction must be adequately translated into user class fares to cover the provider's outlay concerning vehicle hire.

Moreover, when hiring, we assume there is an infinite number of FAVs readily available in each request's surroundings. This assumption entails that there exists an efficient hiring strategy in place, which hires vehicles of adequate capacities in advance and rebalances them to locations where the own fleet historically cannot fulfill the demand. Such an anticipatory hiring strategy could be the key to reaching a more reasonable compromise between FAV owners and providers' objectives in a real-world scenario. In this case, independent owners have an extra incentive to make their vehicles available to join the provider's fleet once they are compensated even before they are assigned to a request.

3.7 Conclusions

This chapter answers the research sub-question SQ2. We have introduced a new approach to actively control service quality in AMoD systems, increasing and decreasing the number of used vehicles in the short term to meet diversified user expectations. We have used these expectations to establish service quality contracts, allowing heterogeneous users to choose ride experiences that best match their preferences, especially regarding maximum waiting times and willingness to share a ride. Based on an experimental study using New York City taxi data, we have found that the developed approach allows us to significantly improve the service quality of all considered user categories. Enforcing the proposed service-level constraints, we can meet the expectations of 85.7% of the users across classes, a 53% average increase in comparison to conventional ridesharing systems. When hiring is enabled, we can meet the expectations of 95.6% of the user requests, at the expense of a mild fleet inflation (a maximum surplus of 168 FAVs is observed at the evening demand peak).

The proposed method allows providers to make a compromise between avoiding service-level violations and hiring extra vehicles, steering the vehicle supply to service the highest priority or most profitable customer segments, according to the particular

conditions of an operational environment (e.g., availability of idle vehicles, hiring costs, user dissatisfaction costs). Nevertheless, hiring occurs only to the extent that it preserves minimum service level requirements. Hence, we add to recent literature by providing an active means to control service quality on an operational level and distinguishing services classes in terms of user expectations.

Moreover, we have formalized the problem using a MILP formulation and proposed a matheuristic to deal with large-scale real-world instances. This matheuristic encompasses the construction of feasible ridesharing visiting plans and the optimal assignment of plans to available vehicles. Besides ride-matching, the assignment phase also includes a reactive rebalancing strategy that uses both service level violations and vehicle hire as stimuli to reposition vehicles. Using a lexicographic method, we have solved a multi-objective function where the primary goal is to minimize rejections (i.e., the most significant source of inconvenience) to subsequently minimize fleet size and service-level violations. We have also evaluated to what extent hiring extra vehicles affects overall service quality and fleet usage, by designing twelve scenarios where we vary the rate at which providers commit to fulfilling user service level expectations. In this way, our results help understand the operational impact of meeting user service quality expectations and the tradeoffs entailed by occasionally violating service quality expectations.

Although we have restricted the analysis to three user segments, the service quality classes we propose are general enough to serve as the basis for more diversified customer-centric services. For the classes considered, our results show that, regardless of the user base, most hired vehicles join the initial fleet to service one-passenger requests from the high-quality user segment (i.e., the business class). These findings indicate that providers may benefit from leveraging the strengths of two AMoD paradigms, namely, ridesharing and single-passenger personal mobility, on a common platform, without the necessity of owning the entire fleet.

One must notice that our commitment to service quality cannot be fully upheld in the face of (i) unexpected traffic disruptions or (ii) a shortage of hireable vehicles. To mitigate the SQ violations caused by (i), providers can carry out an analysis of the features of their specific problem instance to set up realistic SQ deals in the first place. This analysis can take into account, for example, the platform fleet size, the third-party fleet availability, the demand patterns, and the city infrastructure. They can even incorporate safety delay buffers in the service levels, such that the worst-case scenario is always accounted for.

In Chapter 4, we focus on alleviating the adverse effects caused by (ii). We strengthen our problem formulation by addressing the inherent uncertainty of the proposed scenario, in which both FAV supply and heterogeneous demand unfold throughout time. We assume a limited number of FAVs become available throughout the day, such that the AMoD system has to decide when and where it is best to hire third-party vehicles. Since vehicle supply is limited in this scenario, the system no longer can fully meet users' expectations. In order to make up for dissatisfied users, we assume SQ violations are compensated according to the inconvenience level (i.e., extra delay or service denial). From the service provider perspective, however, these compensations are penalties incurred on the company's revenue. To improve service quality and ultimately avoid such penalties, we propose a learning-based approach that leverages stochastic supply and demand information to prevent imbalances from happening.

Chapter 4

Learning to fulfill service level contracts

In the previous chapter, we demonstrated how providers could optimize fleet productivity by dynamically inflating the fleet size, up to the point that user service levels are met fully. Before, we considered that there was no prior knowledge of the class-specific user demand patterns or the availability of third-party owned vehicles, leading to a purely reactive method. Conversely, in this chapter, we show how the optimization process can benefit from exploiting such knowledge. In the face of uncertain demand and idle vehicle supply, we propose a learning-based optimization approach that uses the dual variables of the underlying assignment problem to iteratively approximate the marginal value of vehicles at each time and location under different availability settings. These approximations are used in the optimization problem's objective function to weigh the downstream impact of dispatching, rebalancing, and occasionally hiring idle third-party vehicles in a high-resolution transportation network of Manhattan, New York City. The results show that the proposed policy outperforms a reactive optimization approach in various vehicle availability scenarios while hiring fewer vehicles.

This chapter is structured as follows. We introduce Section 4.1 our literature review in Section 4.2, define the problem in Section 4.3, and formulate it using the language of dynamic resource management in Section 4.4. Section 4.5 presents our approximate dynamic algorithm, and Section 4.6 lays out the details of our experimental study and analyzes the performance of our method when dealing with several transportation scenarios. Finally, Section 4.7 concludes the work and presents an outlook for future research. Parts of this chapter have been submitted to a journal:

B. A. Beirigo, F. Schulte, and R. R. Negenborn. A learning-based optimization approach for autonomous ridesharing platforms with service level contracts and on-demand hiring of idle vehicles. *Transportation Science* (in press, 2021).

4.1 Introduction

Mobility-on-demand (MoD) platforms and *transportation network companies* (TNCs) such as Uber and Lyft have grown substantially and altered mobility behavior worldwide. Although envisioned to make vehicle ownership superfluous and, eventually, alleviate congestion, these ride-hailing platforms have primarily won customers from traditional public transport modes [15]. Ultimately, MoD solutions can only challenge vehicle ownership if service providers can offer high service levels consistently.

Sufficient vehicle supply is a critical factor when it comes to providing consistent service levels efficiently and sustainably. However, most existing models for ridesharing are not capable of responding quickly to significant demand changes. First, often fixed fleet sizes are assumed, which makes it hard to react to demand fluctuations on a tactical level, let alone in real-time. Second, when providers rely on third-party vehicles (i.e., independent drivers), they typically balance supply and demand using surge prices: fares at under-supplied areas dynamically increase both to attract more drivers and suppress excessive demand. Such a strategy, however, is highly controversial since it mainly benefits the platform at the expense of drivers and riders [118]. Regardless of the strategy, some customers end up being penalized with excessive delays, abusive prices, and rejections. With the emergence of *autonomous vehicles* (AVs), however, new possibilities to overcome the shortcomings caused by demand-supply imbalances arise. As soon as vehicle availability is detached from driver availability, ridesharing platforms can count on a larger pool of vehicles, which, currently non-automated, remain parked about 95% of the time [93].

In this chapter, we consider an *autonomous mobility-on-demand* (AMoD) system where a ridesharing platform can occasionally hire *freelance autonomous vehicles* (FAVs), that is, idle third-party-owned AVs, to support its own *platform-owned autonomous vehicles* (PAVs) fulfilling the demand adequately. Hence, in contrast with related literature, we model a highly diversified mobility system where AV ownership is disseminated among the platform and individuals, who simultaneously own and hire out their vehicles. We refer to this system as the AMoD-H. To guarantee service quality, the platform establishes strict *service level contracts* (SLCs) with its user base, such that contract violations (e.g., extra delays, rejections) incur penalties. Hence, by harnessing FAV availability, the platform can shorten the minimum size of the own fleet while addressing personalized demand fluctuations in real-time.

Modeling AMoD-H poses several challenges since requests have to be handled dynamically in the face of (i) irregular FAV availability and (ii) uncertain demand. First, while fleet availability is mostly taken for granted, we assume that the location, announcement time, and total service duration of freelance vehicles are uncertain. For example, FAV availability may resemble that of future AVs whose owners commute by car to work and decide to rent out their vehicles to an AMoD platform during designated intervals such that they have a chance to profit from otherwise unproductive parking times. Second, analogously to service offers in the aviation and rail industry, we segment users into first and second classes, such that the former is willing to pay a premium to enjoy higher service levels. We consider not only the stochastic trip distribution but also class membership distribution when designing anticipatory rebalancing strategies. Based on such details, platforms can improve decision making by taking into account

demand patterns arising within its user base, besides moving forward in the direction of a more personalized user experience. Since we consider a real-world transportation demand setting, determining an optimal policy would incur all "curses of dimensionality", and we are unable to enumerate all possible states and decisions, let alone the uncertainty associated with requests, and FAV hiring. We therefore develop an *approximate dynamic programming* (ADP,[81]) using *value function approximations* (VFAs). In the proposed approach, the dual variables of the underlying assignment problem, defined through a *mixed integer linear programming* (MILP) formulation, are iteratively used to approximate value functions representing the benefit of having an additional vehicle of either type at a certain location and time. Moreover, particularly for the freelance fleet, such approximations also indicate whether it is worthwhile to engage an FAV in further rebalancing or pickup actions, based on its remaining available time or how far it is from its owner's location. At the same time, VFAs are actively used in the objective function of the MILP formulation to weigh the outcome of present decisions (e.g., vehicle rebalancing, parking, and hiring). Eventually, after a number of iterations and value function updates, these learned approximations more accurately represent future states, such that solution quality improves over time. From a methodological perspective, the approach offers:

1. An ADP algorithm for a novel AMoD application that sustains contracted service levels of a heterogeneous user base by controlling vehicle supply on the operation level through on-demand hiring. Requests and third-party AVs arrive stochastically within the service area, such that the platform needs to determine a policy to fulfill the demand using either vehicle type (i.e., PAV or FAV).
2. A hierarchical aggregation structure that summarizes state features using both time and space dimensions. Spatial levels are comprised of increasingly larger clusters (regional centers) set up according to a minimum coverage set formulation on a high-resolution street network of Manhattan, New York City. In turn, temporal levels conform with the level of responsiveness demanded by modern mobility-on-demand applications, in which decisions (e.g., user-vehicle matching, vehicle dispatching, and rebalancing) need to be derived in short intervals.
3. An online discount function that dampens value function approximations arising from decisions involving multiperiod travel times (i.e., resource transformations that take more than one period). Besides leading to more robust estimations and simplifying the state representation, we show that such a discount function enables more complex rebalancing strategies since vehicles can consider varying distance ranges.

From a managerial viewpoint, we show that our policy addresses the requirements of all stakeholders:

- *Users* enjoy personalized service levels and are compensated when these are violated.
- *Cities* can impose strict street use regulations, such as the maximum number of cars per intersection, congestion pricing, and parking schemes. This level of control is enabled by our network representation, which is directly anchored to the real-world physical structure.

- *Independent AV owners* can profit from their cars' idleness by making them available to join a transportation platform during predefined time windows.
- *AMoD platforms* may keep the minimum number of cars necessary to maintain customers' service levels, or instead, rely entirely on the freelance fleet, while maximizing profits.

4.2 Related work

The goal of this literature review is threefold. First, we identify the underlying *dial-a-ride problem* (DARP) our model stems from (Section 4.2.1). Second, we survey studies on transportation platforms in which the demand (parcels or passengers) is fulfilled both by company- and/or third-party-owned vehicles (Section 4.2.2). Third, we analyze the mobility-on-demand literature that considers anticipation mechanisms (Section 4.2.3). We show that our work is the first to address two different sources of uncertainty, namely, third-party vehicle availability and user service levels.

4.2.1 The dynamic and stochastic dial-a-ride problem

In this chapter, we introduce a generalization of the classic *dynamic and stochastic dial-a-ride problem* (DSDARP) (see [45] for a comprehensive survey on DARP). Regarding the sources of uncertainty linked to our problem, as pointed out by Ho et al., stochasticity is generally on the side of demand, and rarely on the side of the supply. The lack of studies on supply stochasticity can also be seen across other transportation problems. For instance, on the *vehicle routing problem* (VRPs) literature, the bulk of stochastic models also focus on uncertain demand features (see [75] for a review on stochastic VRPs) with a few exceptions (e.g., vehicle breakdown). On the other hand, when the demand is a source of uncertainty, service level stochasticity was not explored in the literature yet. This type of stochasticity arises from a user base with heterogeneous customer profile segments whose transportation patterns, as well as their expectations regarding service quality, differ markedly.

4.2.2 On-demand and crowdsourced vehicles

In this section, we review studies in which a crowdsourcing platform matches the demand (partially or entirely) to third-party vehicles. Most studies in this category refer to ridesharing or crowdshipping scenarios, in which a platform seeks to fit riders or parcels into already planned driver routes. Moreover, in contrast with our work, these studies do not consider vehicle automation, such that the availability and preferences of the drivers (rather than the AV owners) have to be taken into account to design feasible routes. Since we only focus on dual-fleet models, the reader may refer to Furuhata et al. [37] and Le et al. [52] for comprehensive reviews on ridesharing and crowdshipping, respectively.

First, regarding the ridesharing scenario, Lee and Savelsbergh [53] assume dedicated drivers complement the ad-hoc fleet, satisfying rider requests that would otherwise remain unmatched. The authors argue that ensuring service levels is essential to retain more participants and investigate the cost-benefit of employing dedicated drivers. Their

findings suggest this cost-benefit depends on the number and time flexibility of the participants, as well as on the similarity between their travel patterns. Santos and Xavier [87] consider a setting where passengers are willing to share both taxis and rides, as long as sharing leads to lower costs than private trips. On the other hand, vehicle owners can reduce costs by servicing multiple passengers on the way to their destination. A *greedy randomized adaptive search procedure* (GRASP) heuristic is used to solve the dynamic version of the problem in a realistic scenario, with requests arriving at every minute and private vehicles at every hour, throughout a twelve-hour horizon. Although the underlying DARP variant they propose is general enough to accommodate our problem, leveraging passenger and vehicle stochasticity is out of the scope of their study. Moreover, following the ridesharing tradition, they assume drivers stop servicing customers as soon as they reach their destination. In contrast, our formulation is closer to the *general pickup and delivery problem* (GPDP) considered by Savelsbergh and Sol [88], where vehicles are stationed at a home depot, from where they can go back and forth within a designated time window.

Second, regarding the crowdshipping scenario, Archetti et al. [5] consider that a company can rely on *occasional drivers* (ODs) besides their own fleet to deliver goods. After arriving at the company's depot, each OD can make at most one delivery, provided that the extra travel distance required to do so, does not violate a flexibility threshold. Arslan et al. [6] build on Archetti et al.'s work by considering ODs can realize multiple pickup and/or drop-off tasks as long the extra time and number of stops does not inconvenience the drivers. Similarly to traditional ridesharing approaches, however, they assume that the delivery platform relies solely on third-party vehicles and consider tasks can be eventually handled by an emergency backup fleet to keep service levels high. Dahle et al. [27] also extend Archetti et al.'s by assuming ODs can perform multiple pickup and delivery operations within a time window. Later, Dahle et al. [28] focus on the design of compensation schemes that can fulfill ODs personal expectations, which are modeled through threshold constraints. They show that even sub-optimal compensation schemes, which do not attract as many ODs, can yield substantial cost savings.

Table 4.1 offers an alternative view on the pickup and delivery literature where requests can be fulfilled both by dedicated and third-party vehicles. In the first column, papers are subsumed under parcel and passenger categories. In the second column, we identify how authors refer to the provider's fleet and the third-party fleet. We also indicate whether the model considers a multi-trip (i.e., vehicles can return to their depot multiple times), or single-trip (i.e., vehicles stop the service as soon as they reach their depot or destination) setting. A checkmark in the "capacity" column indicates each vehicle can handle multiple requests at a time. To highlight how information regarding the third-party vehicles and the customer demand unfolds throughout time, we adopt the standard taxonomy used to classify transportation problems (e.g., VRPs, DARPs). Traditionally, these problems can fall into four categories, namely, static-deterministic (SD), dynamic-deterministic (DD), static-stochastic (SS), and dynamic-stochastic (DS). Dynamic or static classes indicate whether new information (e.g., demand, third-party vehicles) can modify existing plans. In turn, deterministic or stochastic classes indicate whether information about the uncertainty (e.g., demand and third-party vehicle distributions) is available at decision time. Finally, the last column shows the method used to solve each problem.

Table 4.1: *Transportation problems in which demand is (partially) fulfilled by third-party vehicles.*

Reference	Terminology	Third-party fleet supply	Single(S)/ Multi(M) trip	Capacity	Demand	Method
Parcel transportation						
Archetti et al. [5]	company vehicle occasional driver	SD	S		SD	TS
Arslan et al. [6]	back-up vehicle ad hoc driver	DD	S		DD	Heuristic
Dahle et al. [27]	company vehicle occasional driver	DD, DS	S	√	DD	LP, MILP
Dahle et al. [28]	company vehicle occasional driver	DD	S	√	SD	LP, MILP
Savelsbergh and Sol [88]	company vehicle independent driver	SD	M	√	DD	B&P
Passenger transportation						
Lee and Savelsbergh [53]	dedicated driver ad hoc driver	DD	S	√	DD	NS
Santos and Xavier [87]	taxi car owner	SD, DD	S	√	SD, DD	GRASP
This chapter	platform AV freelance AV	DS	M		DS	ADP

Supply & Demand: **SD** (static-deterministic), **SS** (static-stochastic), **DD** (dynamic-deterministic), **DS**(dynamic-stochastic).
Method: **TS** (Tabu Search), **LP** (Linear Programming), **MILP** (Mixed Integer Linear Programming), **B&P** (Branch and Price), **ADP** (Approximate Dynamic Programming), **GRASP** (Greedy Randomized Adaptive Search Procedure), **NS** (Neighborhood Search).

4.2.3 Stochastic mobility-on-demand problems

Assuming a fleet of centrally controlled (autonomous) vehicles, Alonso-Mora et al. [2], Vazifeh et al. [112], and Fagnant and Kockelman [34] have demonstrated that historical taxi demand could be almost entirely fulfilled with significantly fewer vehicles, especially when passengers are willing to share their rides. Studies have also shown that service levels can be substantially improved through anticipatory rebalancing strategies. For example, demand data has been already successively exploited using frequentist approaches (e.g., [3]), reinforcement learning (e.g., [41, 58, 115]), model predictive control (e.g., [48, 106, 121]), and approximate dynamic programming (e.g., [1]). Most of these approaches, however, dimension fleet size experimentally, by simulating configurations that can service the target demand under predefined minimum service level requirements. As pointed out by Vazifeh et al., fleet-size inflation can be required as a consequence of trip-demand bursts, occurring, for instance, after concerts or sports matches. Hyland and Mahmassani [47] refer to this ability to change the fleet size to flex with demand as "fleet size elasticity" and highlight that the benefits of increasing vehicle supply in the short term are likely significant. The authors also point out that, despite uncommon within the context of *shared autonomous vehicle* (SAV) fleet management research, current TNCs rely entirely on this feature, constantly manipulating prices to attract more drivers. Likewise, vehicle ownership may be highly disseminated in the future autonomous mobility market, with most AVs owned by individuals and small fleet operators rather than a single service provider [14]. Although some models are flexible enough to handle dynamic fleet inflation (e.g., [41, 63]), research on short-term SAV fleet size elasticity is still lacking [72].

4.3 Problem description

The AMoD-H emerges on AV-based transportation platforms aiming to fulfill a set of pickup and delivery requests P arising on an street network comprised of a set of nodes N (locations) and a set of directed edges E (streets). Requests arrive in batches P_t, where all requests $r \in P_t$ have arrived at continuous times in the interval $[t-1,\ t)$, for discrete time $t \in \mathcal{T} = \{0, 1, 2, 3, \ldots, T\}$. We consider that request arrival follows a known stochastic process $\mathcal{F}^{\mathrm{P}}$ concerned with two sources of uncertainty:

- *Request distribution:* The number of requests, arrival times, and origin-destination nodes depend on user demand patterns.
- *Request class:* Each request is associated with a service quality class $c \in C$ that identifies minimum service level requirements, particularly maximum pickup delays. Requests render the highest contributions when these requirements are respected, or, otherwise, incur in class-dependent waiting and rejection penalties.

To ensure these service quality requirements are met fully, the platform can hire additional vehicles online to address unexpected supply-demand mismatches. The fleet is comprised of a set of platform-owned autonomous vehicles K^{PAV} and a set of freelance autonomous vehicles K^{FAV}, such that the total fleet set is $K = K^{\mathrm{PAV}} \cup K^{\mathrm{FAV}}$. While PAVs can initiate service at any location $n \in N$, FAVs are distributed throughout a set of locations $O \subseteq N$ where they are typically parked. We refer to locations $o \in O$ as stations since FAVs are required to return to them upon finishing the service contract. Although O is known in advance, the platform deals with three sources of uncertainty (which follow a stochastic process $\mathcal{F}^{\mathrm{O}}$) when dealing with FAVs, namely:

- *Vehicle-station distribution:* Some parking locations can be more prone to accommodate FAVs. For example, vehicles can routinely park in the surroundings of their owner's locations (e.g., workplace, garage), or in more affordable parking places on the outskirts of the city.
- *Announcement time:* Vehicles are available to pick up users at stations at different times for a given day. For example, FAVs can become available downtown as soon as they drop their owners at work. Alternatively, some owners can make their FAVs available (possibly, from their garage) during the night or over the weekend. Regardless of the case, provided that one's itinerary is somewhat irregular due to external factors (e.g., weather, congestion) or particular preferences (e.g., appointments, company's culture), the announcement time can change. For example, a station that typically accommodates a hundred vehicles can have 20%, 50%, and 30% of them arriving in the intervals [7 AM, 8 AM), [8 AM, 9 AM), and [9, 10 AM), respectively.
- *Contract duration:* From the announcement time on, FAV owners make their assets available only during a predefined time interval. Consequently, vehicles must stop servicing users at the right moment, such that they have enough time to travel back to their respective stations before their owner's deadline. Analogously to the announcement times, the contract durations may depend on several factors related to an owner's schedule, leading to varying return deadlines. For example,

contracts can be short (e.g., shopping, doctor appointments), average (e.g., office hours, evening), and long (e.g., the whole weekend, vacation).

Finally, over the planning horizon $\mathcal{T}$, the platform aims to maximize the total contribution accrued by adequately servicing the requests while minimizing the operational costs associated with routing and hiring vehicles.

4.3.1 Example

In Figure 4.1, we illustrate the interplay between the elements of our model. For the sake of simplicity, we represent both vehicle and request discrete locations on a one-dimensional space for each time step such that $N = \{A, B, C, D, E, F, G, H, I, J, K\}$ and consider a time horizon $\mathcal{T} = \{1, 2, \ldots, 10\}$. We assume it takes a single period to travel between each location pair.

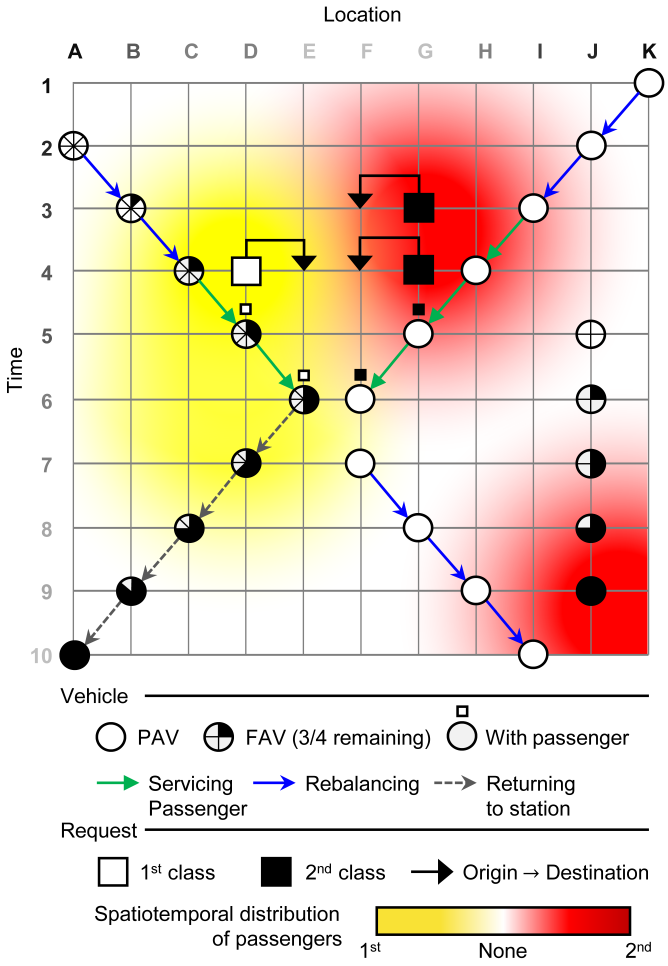


Figure 4.1: *Example of the AMoD-H.*

We represent the stochastic process $\mathscr{F}^{\mathrm{P}}$ in time and space by manipulating the transparency of yellow and red colors, corresponding to first- and second-class customers, respectively (see the color bar at the bottom). Regardless of the class, the more opaque is the color, the higher is the probability of finding a request. We assume first-class requests (i) generate higher profits, and (ii) demand higher service levels, such that failing to adhere to their performance requirements incur higher penalties. We illustrate such higher service levels by assuming first-class users require to be picked up within one period, whereas second-class users are willing to wait up to two periods. In turn, regarding the availability of the freelance fleet, we represent the stochastic process $\mathscr{F}^{\mathrm{O}}$ using the shades of gray on the axis tick labels. We assume that the darker is the shade, the higher is the chance of an FAV appearing. Additionally, we assume FAV contract durations last on average four time steps.

In the following, we describe the behavior of three vehicles in detail throughout ten periods. First, at time $t = 1$, the PAV at location K is faced with two decisions; namely, it can stay in its current location or move to a more promising location in anticipation of future demand. Promising areas consist of high-demand locations that typically generate the highest profits to the service provider. Pursuing such future profits, the PAV performs two rebalance movements, moving empty from K to J, and from J to I. Once it arrives at location I at time $t = 3$, the PAV is assigned to a second-class request (black square) demanding a trip from G to F (solid upper arrow). From this moment on, we consider the PAV is servicing the user, which covers both pick-up and delivery times. Once the PAV delivers the second-class user at F, it stays in F for one period, and then rebalances to location I, in anticipation of future passenger demand.

Second, at decision time $t = 2$, an FAV with an eight-period contract duration becomes available at location A and is immediately rebalanced to the high-demand area. Upon arriving at location C, it is matched to a first-class request demanding a trip from D to E. The FAV travels from C to D to pick up the user and finishes the service at location E and time $t = 6$. By this time, the FAV is available for four additional periods but spends this remaining time traveling back to its station at A to comply with the contract deadline.

Finally, at location J and time $t = 5$, a second FAV with a four-period contract duration appears. However, since it cannot reach high-demanding areas in the subsequent periods, this vehicle ends up not being hired by the platform, staying still until the end of its contract at time $t = 9$.

4.4 Problem formulation

We model the problem using the language of dynamic resource management (see [94]), where AVs (resources) are servicing subsequent trip request batches (tasks) occurring at a discrete-time $t \in \{0, 1, 2, \ldots, T\}$. Subsequently, we present the elements of our model: the system state (Section 4.4.1), information arrival (Section 4.4.2), decisions (Section 4.4.3), costs (Section 4.4.4), and objective function (Section 4.4.5).

4.4.1 System state

The state of a single resource is defined by an four-attribute vector a given by

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} \text{Vehicle type} \\ \text{Remaining servicing time} \\ \text{Station} \\ \text{Current location} \end{pmatrix} = \begin{pmatrix} a_{\text{type}} \\ a_{\text{remain}} \\ a_{\text{station}} \\ a_{\text{current}} \end{pmatrix}.$$

We refer to a single vehicle $k \in K$ with attribute a as k^a. First, the vehicle type attribute a_{type} helps distinguishing between third-party-owned, freelance vehicles (FAVs), and platform-owned vehicles (PAVs). This distinction is crucial because FAVs operate under a stricter availability and different cost plan (owners are entitled to a higher share of the profits).

The duration of such availability, in turn, is captured by the remaining servicing time attribute a_{remain}, which corresponds to the remaining time interval an FAV still can spend servicing orders. PAVs, on the other hand, are assumed to be always available. Thus, as time goes on, an FAV is increasingly unable to pick up new orders, especially those whose destinations are far away from the vehicle's station.

The station attribute a_{station} corresponds to the start and terminal location of each FAV, supposedly, the parking place where the owner expects the vehicle to return once the remaining servicing time has expired. It is worth noting that attributes a_{remain} and a_{station} are not taken into consideration for PAVs; we assume these vehicles are available indefinitely, besides not being obliged to depart from or return to a station.

Finally, similarly to a_{station}, the current location field a_{current} expresses where a vehicle is on the service area. The locations identified by attributes a_{station} and a_{current} integrate the node set N of the street network.

By including the temporal dimension, we have a_t, or the attribute vector of an AV at time t. Let $\mathscr{A}$ be the set of all possible vehicle attribute vectors. The state of all vehicles with the same state vector is modeled using

$$\begin{aligned} R_{ta} &= \text{Number of vehicles with attribute vector } a \text{ at time } t, \\ R_t = (R_{ta})_{a \in \mathscr{A}} &= \text{The resource state vector at time } t. \end{aligned}$$

Each request, in turn, is modeled using a three-attribute vector b, given by

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \text{Origin} \\ \text{Destination} \\ \text{Class} \end{pmatrix} = \begin{pmatrix} b_{\text{origin}} \\ b_{\text{dest}} \\ b_{\text{class}} \end{pmatrix}.$$

We refer to a single trip $r \in P$ with attribute vector b as r^b. Similarly to vehicle locations, origin (b_{origin}) and destination (b_{dest}) attributes correspond to nodes of the street network (i.e., b_{origin}, $b_{\text{dest}} \in N$), whereas the b_{class} attribute identifies the requested service quality $c \in C$.

Let $\mathscr{B}$ be the set of all possible request attribute vectors. The state of all rides with the same state vector occurring at time t is modeled using

$$\begin{aligned} D_{tb} &= \text{The number of trips with attribute vector } b \text{ at time } t, \\ D_t = (D_{tb})_{b \in \mathscr{B}} &= \text{The request state vector at time } t. \end{aligned}$$

With the resource and request state vectors, we defined our system state vector as

$$S_t = (R_t, D_t).$$

4.4.2 Exogenous information

Although the underlying system is known to evolve continuously over time, we measure states S_t before making any decisions at discrete periods $t \in \{0, 1, 2, 3, \ldots, T\}$. Between subsequent periods $t-1$ and t, we account for the exogenous information processes concerning both vehicle and demand attribute updates using variables

$$\begin{aligned} \hat{R}_{ta} &= \text{The change in the number of FAVs with attribute } a \text{ resulting from information arriving between } t-1 \text{ and } t, \\ \hat{D}_{tb} &= \text{The number of new requests with attribute } b \text{ placed between } t-1 \text{ and } t, \\ W_t = (\hat{R}_t, \hat{D}_t) &= \text{Exogenous information arriving between } t-1 \text{ and } t. \end{aligned}$$

For the complete stochastic process, we let $\omega \in \Omega$ represent the sample path W_1, $W_2, \ldots, W_T$, where Ω is the set of all sample paths.

In this chapter, we consider $\hat{R}_{ta}$ is concerned only with FAVs entering the system. However, it could also account for several alternative sources of uncertainty, such as travel delays, vehicle breakdowns, and early termination of FAV contracts. In any case, whenever an AV attribute changes randomly from a to a', we would have $\hat{R}_{ta} = -1$ and $\hat{R}_{ta'} = +1$.

4.4.3 Decisions

Regarding the types of decisions used to act on the fleet, we consider every vehicle can *service* a user (which is reachable within his maximum pickup delay), *stay parked* in its current location waiting to pick up users, *rebalance* to a more promising location, or, in the case of FAVs, *return* to its station before the contract deadline. Decisions are described using

$d^{\text{stay}} =$ Decision to stay parked in the current location,

$d^{\text{return}} =$ Decision to return to the station (FAV only),

$\mathscr{D}^{\text{R}} =$ Set of all decisions to rebalance (i.e., move empty) to a set of neighboring locations,

$\mathscr{D}^{\text{S}} =$ Set of all decisions to service a user, where an element $d \in \mathscr{D}^{\text{S}}$ represents the decision to cover a trip request of type $b_d \in \mathscr{B}$,

$\mathscr{D}^{\text{S}}_c =$ Subset of decisions in $\mathscr{D}^{\text{S}}$ associated with each service quality $c \in C$,

$\mathscr{D} =$ Set of all decisions $d \in \mathscr{D}^{\text{S}} \cup \mathscr{D}^{\text{R}} \cup \{d^{\text{stay}}\} \cup \{d^{\text{return}}\}$,

$x_{tad} =$ Number of times decision d is applied to a vehicle with attribute vector a at time t,

$x_t = (x_{tad})_{a \in \mathscr{A}, d \in \mathscr{D}} =$ Decision vector at time t.

Transition function

To model how decisions affect vehicle states, we consider a deterministic transition function a^M. Hence, before any new information arrives, applying a decision d to a vehicle with attribute vector a at time t, leads to a post-decision attribute vector

$$a' = a^M(a,d).$$

In turn, the new time of availability is given by

$$t' = \begin{cases} t+1, & \text{if } d = d^{\text{stay}}, \\ t+\tau(t,a,d), & \text{if } d \in \mathscr{D}^{\text{S}} \cup \mathscr{D}^{\text{R}} \cup \{d^{\text{return}}\}, \end{cases}$$

where $\tau(t,a,d)$ is the travel time spent to carry out a decision d to service a user, rebalance to another location, or return to the station. We consider that between t and t', vehicles are busy, such that the system cannot exert any control over them. Therefore, if, for instance, a decision d to cover a trip b is applied to a vehicle with attribute vector a at time t, this vehicle will end up in state a' (with $a'_{\text{current}} = b_{\text{dest}}$), and can only be used again at t'.

Abiding by the street network capacity

To avoid unrealistic vehicle distributions, we consider locations $j \in N$ can only accommodate up to $k_j^{\max}$ vehicles. In a real-world setting, different locations have different capacities, which may depend not only on the physical infrastructure (e.g., number of parking places), but also on city regulations. An artificial threshold may be imposed, for instance, to alleviate local congestion or improve accessibility to surrounding facilities. To implement this restriction, we keep track of the number of vehicles k_j^{inbound} ($0 \leq k_j^{\text{inbound}} \leq k_j^{\max}$) inbound to j and assure that at most $k_j^{\max} - k_j^{\text{inbound}}$ extra vehicles can enter j.

Further, we define the set of all decisions leading vehicles k^a to post-decision locations j as

$$\mathscr{D}_{a,j} = \{d \mid a' = a^M(a,d),\, a'_{\text{current}} = j,\, a'_{\text{current}} \neq a'_{\text{station}},\, \forall d \in \mathscr{D} \setminus \mathscr{D}^{\text{S}}\},$$

Using $\mathscr{D}_{a,j}$ and k_j^{inbound}, we can calculate the post-decision number of vehicles inbound to j (through either rebalance or stay decisions), and ensure the maximum capacity of j is not violated (see constraints (4.3)). One must notice that $\mathscr{D}_{a,j}$ does not cover FAVs inbound to their own stations (i.e., $a'_{\text{current}} = a'_{\text{station}} = j$). FAVs are assumed to have free access to their home stations at any time.

Fulfilling contract time windows

An FAV with attribute vector a can only be acted on using a decision d to stay, rebalance, or service users, when there is enough remaining servicing time to return to its station, that is

$$\tau(t,a,d) + \tau(t',a',d^{\text{return}}) \leq a_{\text{remain}}\ \forall a \in \mathscr{A},\, d \in \mathscr{D} - d^{\text{return}}.$$

Otherwise, the decision is deemed to be invalid, and the corresponding x_{tad} variable is preemptively discarded. At each period t, we define the set of vehicle attribute vectors associated to FAVs that must return to their station using

$$\mathscr{A}^{\text{return}} = \{a \mid \forall a \in \mathscr{A}^{\text{FAV}},\, \tau(t,a,d^{\text{return}}) = a_{\text{remain}}\}.$$

Although FAVs will eventually realize the return decision, we consider that they can always return to their station directly, even before their contract due time. Doing so adds flexibility to FAV operation since they can rebalance back and forth from their station, when suitable. This way, provided that FAV owners have already covered parking costs at their stations, the platform may find it worthwhile rebalancing FAVs back sometimes, to evade city parking costs.

Constraints

The decision variables x_{tad} must satisfy the following constraints:

$$\sum_{d \in \mathscr{D}} x_{tad} = R_{ta} \qquad \forall\, a \in \mathscr{A} \qquad (4.1)$$

$$\sum_{a \in \mathscr{A}} x_{tad} \leq D_{tb_d} \qquad \forall d \in \mathscr{D}^{\text{S}} \qquad (4.2)$$

$$\sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}_{a,j}} x_{tad} \leq k_j^{\max} - k_j^{\text{inbound}} \qquad \forall j \in N \qquad (4.3)$$

$$x_{tad^{\text{return}}} = R_{ta} \qquad \forall a \in \mathscr{A}^{\text{return}} \qquad (4.4)$$

$$x_{tad} \geq 0 \qquad \forall\, a \in \mathscr{A},\, \forall d \in \mathscr{D} \qquad (4.5)$$

Constraints (4.1) guarantee that all available vehicles (i.e., parked at the current period t) are assigned to a decision, whereas constraints (4.2) ensure that any trip request (identified by b_d) can be assigned to at most one vehicle. In turn, constraints (4.3) enforce that the number of vehicles inbound to a location j do not surpasses j's remaining capacity. Finally, constraints (4.4) ensure that every FAV whose returning trip delay $\Delta t(a_{\text{current}}, a_{\text{station}})$ is equal to its remaining contract duration a_{remain} is obliged to return to its station.

4.4.4 Cost function

Applying a decision d to a vehicle k^a at time t, takes the vehicle to state a' at time t' and generates a contribution c_{tad} given by

$$c_{tad} = \begin{cases} \textit{Service user} \\ \beta^k \cdot \left(p^c_{\text{base}} + p_{\text{time}} \cdot \Delta t_{\text{trip}} \; - \; c^k_{\text{time}} \cdot (\Delta t_{\text{pickup}} + \Delta t_{\text{trip}}) \; - \; c^c_{\text{delay}} \cdot w_{\text{delay}} \right), \\ \textit{Rebalance vehicle} \\ - \, c^k_{\text{time}} \cdot \Delta t_{\text{rebalance}}, \\ \textit{Return to station (only FAVs)} \\ - \, c^k_{\text{time}} \cdot \Delta t_{\text{return}}, \\ \textit{Stay parked} \\ c^{t,j}_{\text{stay}}. \end{cases}$$

Contributions c_{tad} are comprised of

$\beta^k =$ Platform profit margin when using vehicle k,
$p^c_{\text{base}} =$ Base fare of request $b = b_d$ of decision d from quality class $c = b_{\text{class}}$,
$p_{\text{time}} =$ Time-dependent fare,
$\Delta t_{\text{trip}} =$ Trip duration $\Delta t(b_{\text{origin}}, b_{\text{dest}})$ of request $b = b_d$,
$c^k_{\text{time}} =$ Time-dependent operational cost of vehicle k,
$\Delta t_{\text{pickup}} =$ Pickup duration $\Delta t(a_{\text{current}}, b_{\text{origin}})$ from current location to trip b origin,
$c^c_{\text{delay}} =$ Penalty due to the excess delay w_{delay},
$w_{\text{delay}} =$ Excess delay over the pickup delay w^c_{pickup} contracted by a user from class c,
$\Delta t_{\text{rebalance}} =$ Rebalance travel duration $\Delta t(a_{\text{current}}, r)$ to target location $r \in N$,
$\Delta t_{\text{return}} =$ Return travel duration $\Delta t(a_{\text{current}}, a_{\text{station}})$ of vehicle with $a_{\text{type}} = FAV$,
$c^{t,j}_{\text{stay}} =$ Cost of staying at location j at time t, such that $j = a_{\text{current}}$, and $j \in N$.

The profit margin β^k determines the percentage owed to the platform by assigning trips to a vehicle k^a of type $a_{\text{type}} \in \{\text{PAV, FAV}\}$. It allows us to adequately adjust, from the perspective of the platform, the incentive FAVs have to serve at their available times. Similarly to today's MoD applications, we assume most profits belong to the independent contractors, namely, FAV owners. In turn, constant c^k_{time} represents typical operational costs (e.g., tolls, fuel, wear and tear) for a vehicle k. We consider these costs are equal for

all vehicles, regardless of the type. The basis of a pay-per-use parking system is captured by the cost $c^{t,j}_{\text{stay}}$ of staying at the current location at time t, allowing city managers to create incentives for vehicles to avoid parking in congested areas. A user who requests a ride in class c expects to be picked up within w^c_{pickup} time units, but can tolerate up to $w^c_{\text{tolerance}}$ time units over w^c_{pickup} to be serviced, as long as he is compensated for the excess delay $w_{\text{delay}} = \max\{0,\ \Delta t_{\text{pickup}} - w^c_{\text{pickup}}\}$ and $w_{\text{delay}} \leq w^c_{\text{tolerance}}$. From the platform perspective, this compensation represents a delay penalty c^c_{delay} incurred for $w_{\text{delay}} > 0$, defined as

$$c^c_{\text{delay}} = p^c_{\text{base}} / w^c_{\text{tolerance}}.$$

This way, if $w_{\text{delay}} = w^c_{\text{tolerance}}$, the base fare is totally offset by the penalty, and the platform will only profit from the time-dependent fare. Further, we consider that when the platform fails to pick up a user from class c within the class maximum waiting time $w^c_{\text{pickup}} + w^c_{\text{tolerance}}$, the platform has to bear a rejection penalty

$$c^c_{\text{rejection}} = \rho \cdot p^c_{\text{base}},$$

where ρ is a penalty factor. Hence, when rejecting a request, the platform does not only fail to profit but may be required to compensate the inconvenienced users for a breach of contract, according to their service-level class. By setting up ρ, we can choose the extent to which rejections incur further losses, allowing us to experiment with different penalization schemes. Ultimately, for each period t, the total rejection penalty resulting from failing to service users from different classes is given by the function

$$P_t(S_t, x_t) = \sum_{c \in C} c^c_{\text{rejection}} \Big(\sum_{\substack{b \in \mathscr{B} \\ b_{\text{class}} = c}} D_{tb} - \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}^S_c} x_{tad} \Big). \tag{4.6}$$

We consider the service level violation penalties c^c_{delay} and $c^c_{\text{rejection}}$ are an essential element of SLCs, once they further back up the platform's commitment to service level fulfillment. By combining these two penalties, we guarantee that the platform is always better off servicing a user, regardless of the service level violation. Even when the base fare is totally offset by the delay penalty, the platform still can profit from the time-dependent fare whereas rejecting a user always leads to losses. Finally, the contribution function representing the profit a platform can accrue at each period t is given by

$$C_t(S_t, x_t) = \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} c_{tad} x_{tad} \ - \ P_t(S_t, x_t). \tag{4.7}$$

4.4.5 Objective

Let $X_t^\pi(S_t)$ be a decision function representing a policy $\pi \in \Pi$ that maps a state $S_t \in \mathcal{S}$ to a decision $x_t \in \mathcal{X}_t$, where $\mathcal{S}$ is the state space, $\mathcal{X}_t$ is the set of feasible decisions in state S_t, and Π is the set of potential decision functions. Starting from an initial state S_0, we aim to determine the optimal policy π^* that maximizes the expected cumulative contribution, discounted by a factor γ, over the planning horizon T:

$$F_0^*(S_0) = \max_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t=0}^{T} \gamma C_t\left(S_t, X_t^\pi(S_t)\right) \mid S_0 \right\}. \tag{4.8}$$

4.5 Algorithmic strategies

In principle, we can solve Equation (4.8) by means of classical dynamic programming, recursively computing (backward through time) Bellman's optimality equations

$$V_t(S_t) = \max_{x_t \in \mathcal{X}_t} \left(C_t(S_t, x_t) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1}) \mid S_t, x_t\}\right), \tag{4.9}$$

where $S_{t+1} = S^M(S_t, x_t, W_{t+1})$. The transition function $S^M(.)$ describes how the pre-decision state S_t evolves to the subsequent pre-decision state S_{t+1}, upon applying decisions x_t and receiving random information W_{t+1}. For each period, using the expected contributions V_{t+1} allows us to account for the downstream effect of decision making.

Solving (4.9), however, is computationally intractable for our problem setting. Doing so, would incur in all the three "curses of dimensionality" [81]. First, we are unable to enumerate all states S_t in state space $\mathcal{S}$, to evaluate value functions $V_t(S_t)$. Second, we are unable to find the optimal decision from the decision space $\mathcal{X}_t$ for all states in $\mathcal{S}$. Third, we are unable to determine the outcome space, whose dimensionality depends on the random information W_{t+1}, which for our problem comprises the uncertainty associated with the appearance of requests and FAVs.

4.5.1 An approximate dynamic programming algorithm

To estimate the value functions around each state in Equation (4.9), we develop an approximate value iteration algorithm (see, e.g., [82]). This ADP algorithm relies on the concept of post-decision state, which is a deterministic state immediately after implementing a decision and before any new information has arrived. Thus, applying the decision vector x_t to state S_t leads to a deterministic post-decision state

$$S_t^x = S^{M,x}(S_t, x_t),$$

where $S^{M,x}(.)$ is a transition function describing how the system evolves from S_t to S_t^x using decisions x_t. Then, from the post-decision state S_t^x, we can compute the subsequent pre-decision state

$$S_{t+1} = S^{M,W}\left(S_t^x, W_t(\omega)\right)$$

using $S^{M,W}(.)$, a transition function that models the arrival of new information $W_t(\omega)$. Through these functions, using a given policy π over the planning horizon T would produce a sequence $(S_0, x_0, S_0^x, W_1(\omega), S_1, x_1, S_1^x, W_2(\omega), S_2, \ldots, S_{T-1}, S_{T-1}^x, W_T(\omega), S_T)$. By breaking Equation (4.9) into two steps we have:

$$V_t(S_t) = \max_{x_t \in \mathscr{X}_t} \left(C(S_t, x_t) + \gamma V_t(S_t^x)\right),$$

$$V_t(S_t^x) = \mathbb{E}\left\{V_{t+1}(S_{t+1}) \mid S_t^x\right\}.$$

Since we cannot compute $V_t(S_t^x)$ exactly, we aim to find $\overline{V}_t(S_t^x)$, that is, a value function approximation around the deterministic post-decision state S_t^x. Once we already penalize rejections, we assume the unmet requests from the post-decision demand vector D_t^x are not carried over to future periods (i.e., we set $D_t^x = \emptyset$). In practice, this assumption implies that users will either walk away or re-enter the system in the next period through a new request upon being rejected. Hence, the post-decision state is equivalent to the post-decision resource vector, that is, $\overline{V}_t\left(S_t^x\right) = \overline{V}_t\left(R_t^x\right)$. Following the ADP algorithm, we estimate these approximations iteratively, such that at each iteration $n = 1, 2, \ldots, I$, a different sample path ω^n is considered, and we can take decisions using the value functions learned up to iteration $n-1$. Accordingly, to indicate the iterative nature of the algorithm, a superscript n is added to all variables.

Assuming $\overline{V}_t^n\left(R_t^{x,n}\right)$ is linear in R_{ta}^n, we have

$$\overline{V}_t^n(R_t^{x,n}) = \sum_{a' \in \mathscr{A}} \overline{v}_{t'a'}^n \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} \delta_{a'}(a,d) x_{tad}, \tag{4.10}$$

where

$\overline{v}^n{}_{t'a'} =$ Marginal value of a vehicle with post-decision attribute vector a' at arrival time t' at iteration n,

$\delta_{a'}(a,d) =$ Transition function equals to 1, if $a^M(a,d) = a'$, and 0, otherwise.

In our problem, the marginal values $\overline{v}_{ta}^n$ have slightly different interpretations, depending on vehicle type. For PAVs, these values approximate the overall contribution (i.e., until the end of the simulation horizon T) of assigning an incremental vehicle to a certain location at a certain time. For FAVs, however, a marginal value also reflects a vehicle's remaining contract duration and the station location. For example, FAVs with higher remaining service durations, operating in locations close to their stations, are expected to draw higher contributions. Conversely, FAVs far from their stations and with contracts about to expire, cannot render as high contributions since the last moments of their contract are reserved to a return trip to their station.

Although we assume $\overline{V}_t^n\left(S_t^{x,n}\right)$ is linear in R_t^n, we acknowledge this assumption is prone to result in an oversupply of vehicles in regions associated with high marginal values. Intuitively, the more vehicles rebalance to a certain region, the lower becomes their average contribution since only a few of them actually will service users. Instead of dampening these values as the number of vehicles increases (by using piecewise-linear approximations as in [102]), we limit the number of vehicles arriving at each network location. Our fine-grained spatiotemporal representation (featuring short pe-

riods and exact street coordinates), allow us to restrict the number of vehicles entering each location in constraints (4.3). Besides avoiding vehicles flooding certain areas, these constraints add a degree of realism to the model since they are based on the physical capacity of the actual infrastructure as well as city's traffic rules.

We do not restrict the number of vehicles dropping passengers at the same location because they are already bounded by the number of demands. Due to the characteristics of our problem setting, especially the adoption of short periods and the spatiotemporal distribution of the demand, it is unlikely that a high number of users are arriving at the same location at the same time.

Finally, the problem of finding the optimal decision function is

$$X_t^\pi\left(S_t^n\right) = \underset{x_t \in \mathscr{X}_t^n}{\arg\max} \left(\sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} c_{tad} x_{tad} + \gamma \sum_{a' \in \mathscr{A}} \overline{v}_{t'a'}^{n-1} \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} \delta_{a'}(a,d) x_{tad} \right) \tag{4.11}$$

$$= \underset{x_t \in \mathscr{X}_t^n}{\arg\max} \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} \left(c_{tad} + \gamma \sum_{a' \in \mathscr{A}} \overline{v}_{t'a'}^{n-1} \delta_{a'}(a,d) \right) x_{tad} \tag{4.12}$$

$$= \underset{x_t \in \mathscr{X}_t^n}{\arg\max} \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} \left(c_{tad} + \gamma \overline{v}_{t'a^M(a,d)}^{n-1} \right) x_{tad}. \tag{4.13}$$

4.5.2 A discount mechanism for multiperiod travel times

Mobility-on-demand users typically require quick response times from transportation platforms. For this reason, most studies on urban MoD applications either process requests as soon as they are placed or in batches, usually considering short time intervals. Following such practice in our ADP approach, however, prevents us from assuming that all decisions acting on the resources will be completed in the subsequent period. In fact, most pickup and rebalancing decisions can last longer than a single period. Although we work with a high-resolution street network, our locations still correspond to a restricted subset of all possible coordinates. The lower is the resolution of the underlying map, the fewer are the locations available, and the more multiperiod travel times can be expected between location pairs. Therefore, incorporating such a feature helps to create a more robust solution, independent of the length of the periods or the underlying map structures.

Such multiperiod resource-transformation times have a significant influence on the value function approximations. To avoid adding another attribute to our resource attribute vector to account for the arrival time at the destination location (see [102]), we implement an online discount mechanism to all value function approximations associated with post-decision states arising from rebalancing decisions. We dampen the value function of post-decision states $a' = a^M(a,d)$ by discounting the opportunity cost of staying still (i.e., $d = d^{\text{stay}}$) during the rebalancing periods $t'' \in \{t+1,\ t+2,\ \ldots,\ t'-1\}$ using

$$\overline{v}_{tt'a'}^n = \overline{v}_{t'a'}^n - \sum_{t''=t+1}^{t'-1} \overline{v}_{t''a^M(a,d^{\text{stay}})}^n \forall d \in \mathscr{D}^{\text{R}}. \tag{4.14}$$

In Equation (4.14), if the resource-transformation time takes a single period (i.e., $t' = t+1$) we have $\overline{v}^n_{tt'a'} = \overline{v}^n_{t'a'}$. Since we do not allow vehicles to interrupt a rebalance trip, this adaptation is crucial to avoid vehicles are too far-sighted, pursuing future rewards at long-distance locations while ignoring the requests that might occur (in the next periods) in the surroundings of their starting location after their departure. On the other hand, this adaptation also avoids that vehicles are stranded in low-demand areas, allowing them to move directly to farther high-demand areas instead of endlessly rebalancing to nearby low-demand neighbors. Therefore, at every decision time, an idle vehicle can also rebalance to farther, high-value function locations, as long as this decision is (i) at least as good as staying still for the whole rebalancing time, and (ii) competitive in relation to rebalancing to its closest neighbors.

4.5.3 Value function updates

At iteration n, we consider a sample path ω^n that determines $\hat{R}^n_t = \hat{R}_t(\omega^n)$ and $\hat{D}^n_t = \hat{D}_t(\omega^n)$, such that $W_t(\omega^n) = (\hat{R}^n_t, \hat{D}^n_t)$. Let $\overline{V}^{n-1}_t(S^{x,n}_t)$ be an approximation of the value of being in the post-decision state $S^{x,n}_t = S^{M,x}(S^n_t, x_t)$ considering the first $n-1$ iterations. Given the state $S^n_t = S^{M,W}(S^{x,n}_{t-1}, W_t(\omega^n))$, we can make decisions at time t by solving the optimization problem

$$F\left(S^n_t\right) = \underset{x_t \in \mathcal{X}^n_t}{\arg\max}\Big(C_t\left(S^n_t, x_t\right) + \gamma \overline{V}^{n-1}_t(S^{x,n}_t)\Big), \tag{4.15}$$

where we seek to determine the decision vector x_t in the feasible region $\mathcal{X}^n_t$ that maximizes the sum of the current contribution and the expected contribution (discounted by a γ factor).

In Algorithm 4.1, we present how our optimization problem is inserted into a classic ADP algorithm. First, all value function approximations are set to zero by default. Then, we start from an initial state $S^1_0 = (R^1_0, D^1_0)$, where R^1_0 comprises the state vectors of PAVs randomly distributed throughout the map, and D^1_0 is empty (i.e., no requests have arrived). We update value functions $\overline{v}^n_{ta}$ using the samples $\hat{v}^n_{ta}$ drawn from attribute vector a at time t and iteration n. New samples are smoothed using stepsizes α_n which are updated every iteration according to the McClain's rule (see [40]), such that

$$\alpha_n = \frac{\alpha_{n-1}}{1 + \alpha_{n-1} - \overline{\alpha}},$$

where $\overline{\alpha}$ is a constant that is approached as n advances. Initially, we set $\alpha_1 = 1$ such that value functions can start with the first sample value measured for each state.

4.5.4 Approximating the value function

Since we are unable to enumerate all the attributes in the state-space $\mathcal{A}$, we use hierarchical aggregation to create a sequence of state spaces. Aggregating on the space dimension helps to estimate the value function of states featuring locations that were not yet visited, by using the estimates of regions in hierarchically superior levels. We define regions by clustering nodes in our street network that can be accessed from central

Algorithm 4.1: An approximate dynamic programming algorithm to solve the AMoD-H assignment problem.

Choose an initial approximation $\overline{V}_t^0$, $\forall t \in \mathscr{T} = \{0,1,\ldots,T\}$.
Set the initial state to S_0^1.
for $n = 1,\ldots,I$ **do**
  Choose a sample path ω^n.
  for $t = 0,1,\ldots,T$ **do**
    Let x_t^n be the solution of the optimization problem (4.15).
    Let $\hat{v}_{ta}^n$ be the dual variable corresponding to the resource conservation constraint (4.1) for each $R_{ta}^n > 0$.
    Update the value function using:
      $\overline{v}_{ta}^n = (1-\alpha_n)\overline{v}_{ta}^{n-1} + \alpha_n \hat{v}_{ta}^n$.
    Compute the subsequent pre-decision state:
      $S_{t+1}^n = S^{M,W}(S_t^{x,n}, W_{t+1}(\omega^n))$ with $S_t^{x,n} = S^{M,x}(S_t^n, x_t^n)$.
    Update the total number of vehicles K_j inbound to each location $j \in N$.
Return the value functions: $\{\overline{v}_{ta}^n \mid \forall t \in \mathscr{T}, \forall a \in \mathscr{A}\}$.

locations within increasingly higher maximal delays. Besides aggregating across space, near periods can aggregate up to larger time intervals since the value function of a vehicle at a location (or region) is likely to carry some resemblance to the value functions of anterior/posterior periods. Such resemblance, therefore, allow us to approximate value functions across periods that belong to longer time intervals. Later, we present the final spatiotemporal hierarchical aggregation structure, achieved experimentally by assessing the performance of different aggregation structures on a single baseline scenario.

Hierarchical aggregation

In order to estimate the value function of attributes not yet observed, we use hierarchical aggregation coupled with the *weighting by inverse mean squared errors* (WIMSE) formula proposed by George et al. [39]. Our hierarchical aggregation structure lays out a sequence of state spaces $\{(\mathscr{T} \times \mathscr{A})^{(g)}, g = 1,2,\ldots,|\mathscr{G}|\}$ with successive fewer elements, where $(\mathscr{T} \times \mathscr{A})^{(g)}$ represents the g^{th} level of aggregation of the time-attribute space $\mathscr{T} \times \mathscr{A}$. Hence, each attribute $ta \in \mathscr{T} \times \mathscr{A}$ can be aggregated up to an attribute $ta^{(g)} = G^g(ta)$, where $G^g : \mathscr{T} \times \mathscr{A} \to (\mathscr{T} \times \mathscr{A})^{(g)}$. Doing so allows us to estimate the value $\overline{v}_{ta}^n$ associated with an attribute ta by combining the values $\overline{v}_{ta}^{(g,n)}$ from superior levels using

$$\overline{v}_{ta}^n = \sum_{g \in \mathscr{G}} w_{ta}^{(g,n)} \cdot \overline{v}_{ta}^{(g,n)}.$$

Weights $w_{ta}^{(g)}$ on the estimates of different aggregation levels are inversely proportional to the estimates of their mean squared deviations, according to the WIMSE formula:

$$w_{ta}^{(g,n)} \propto \frac{1}{\left(\overline{\sigma}_{ta}^2\right)^{(g,n)} + \left(\overline{\mu}_{ta}^{(g,n)}\right)^2},$$

where $\left(\overline{\sigma}_{ta}^2\right)^{(g,n)}$ is the variance of the estimate $\overline{v}_{ta}^{(g,n)}$, and $\left(\overline{\mu}_{ta}^{(g,n)}\right)^2$ is the aggregation bias, that is, the difference between the estimate $\overline{v}_{ta}^{(g,n)}$ at aggregate level g and the

estimate $\overline{v}_{ta}^{(0,n)}$ at the disaggregate level. Next, we normalize all weights by doing

$$w_{ta}^{(g,n)} = \frac{1}{\left(\overline{\sigma}_{ta}^{2}\right)^{(g,n)} + \left(\overline{\mu}_{ta}^{(g,n)}\right)^{2}} \left[\sum_{g' \in \mathcal{G}} \frac{1}{\left(\overline{\sigma}_{ta}^{2}\right)^{(g',n)} + \left(\overline{\mu}_{ta}^{(g',n)}\right)^{2}} \right]^{-1}. \tag{4.16}$$

The street network map

AMoD studies within the scope of reinforcement learning and ADP generally consider cars can rebalance to their immediate neighboring zones. Moreover, such rebalancing operations are expected to last at most a single period, such that, at decision time, all vehicles are either servicing customers or idle (potentially, after finishing rebalancing). Such zones, however, are defined using artificial grids (e.g., [1, 115]) or neighborhood borders (e.g., [41, 58]), which not necessarily translate into realistic drivable streets. In contrast, we work with a high-resolution transportation network of Manhattan comprised of 6,430 nodes and 11,581 edges. Therefore, pickup and rebalance decisions consist of movements between real-world street coordinates, discretized in a set of network nodes, which we guarantee to be no longer than thirty seconds away from one another (at an average speed of 20km/h). Such a high-granularity setup allows us to consider a more realistic demand matching scenario since real-world trip requests have a larger set of candidate nodes to which their GPS coordinates can be approximated.

Hierarchical regional centers

In order to define hierarchical regions in our street-network map, we implement a variant of the facility location problem proposed by Toregas et al. [103], which is concerned with the time that separates a location from its closest facility. The goal of this problem is to determine the minimum set of facilities in the street network that together can cover (reach) all others within s time units. Let

$x_j =$ 1, if a facility is located at $j \in N$, 0 otherwise,
$t_{i,j} =$ Travel time between nodes $i, j \in N$,
$s =$ The maximal service delay of a vehicle departing from a facility,
$N_{p,s} =$ Subset of locations able to reach location $p \in N$ within s time units (i.e., $N_{p,s} = \{j \mid t_{jp} \leq s,\ \forall j \in N\}$).

The minimum set covering problem is defined as follows:

Minimize:

$$\sum_{j \in N} x_j \tag{4.17}$$

Subject to:

$$\sum_{j \in N_{p,s}} x_j \geq 1 \qquad \forall\ p \in N \tag{4.18}$$

$$x_i \in \{0,1\} \qquad \forall\ \mathrm{i} \in N \tag{4.19}$$

An optimal solution to this set covering problem would give us the location of the minimum set of facilities $J_s \subseteq N$ that would be required to service all locations p and still ensure a maximal service time of s units for the entire system. We assume these facilities are regional centers $j \in J_s$, and consider that each node $p \in N$ integrates the region of its closest center $j = \arg\min_{i \in J_s} t_{ip}$.

4.5.5 Benchmark policy

We benchmark our method against a myopic policy π_{myopic} comprised of two phases. In the first phase we determine the optimal vehicle-request assignment at period t by maximizing the contribution function given by Equation (4.7). Next, in the second phase, idle vehicles are optimally rebalanced to under-supplied locations using the linear program proposed by Alonso-Mora et al. [2]. This program aims to minimize the total travel distance of reaching the pickup locations of unassigned requests while guaranteeing that either all vehicles or all requests are assigned. The original formulation is adapted such that it abides by the contractual deadlines of freelance vehicles. We preemptively discard FAVs that, although idle, cannot reach any rebalancing targets within their remaining service time.

To increase the matching rate, we assume rebalancing decisions can be revoked at every decision time. Hence, in the first phase of our policy, both rebalancing and idle vehicles can be assigned to new requests. Thanks to our high-resolution network representation, we can calculate the current location of all rebalancing vehicles at each period t. Therefore, rebalancing vehicles can be matched to any request occurring in the surroundings of their ongoing route (i.e., the shortest path to their destination).

4.6 Experimental study

We implemented our approach using Python 3.6 and Gurobi 8.1. Test cases were executed on a 2.60 GHz Intel Core i7 with 32 GB RAM.

4.6.1 Training and testing datasets

We create our dataset by randomly sampling 10% of the 2011 Manhattan, New York City taxi demand. Value functions are created using requests sampled from the 15^{th} Tuesday, and their effectiveness is assessed on testing instances created using samples from the remaining 51 Tuesdays of 2011. This setup allows that we investigate how the policy learned from a single weekday performs throughout the whole year. To assess the quality of our VFA policy, we compare the average profit and service level across the 51 samples against the averages provided by the benchmark policy. Figure 4.2 shows the total request count for each day. For the sake of fairness, the random processes associated with trip sampling, service class assignment, and fleet distribution are a function of the iteration number (i.e., the seed). This way, regardless of the configuration, we guarantee that the same information will be considered across all training iterations and testing instances.

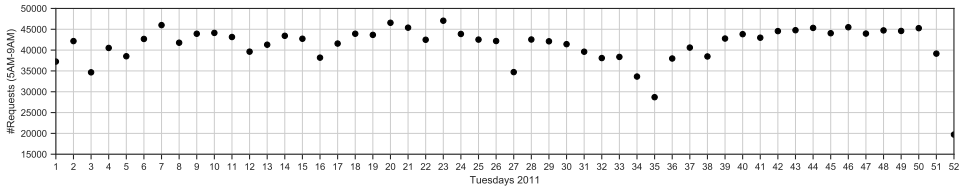


Figure 4.2: *Request count between 5 AM and 9 AM throughout all 52 Tuesdays of 2011 of the Manhattan taxi demand dataset. VFAs are determined using only samples from the 15th Tuesday.*

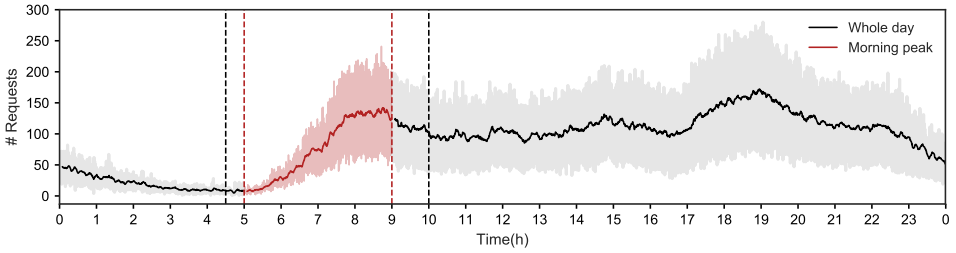


Figure 4.3: *Demand pattern of Manhattan taxi trips on a typical Tuesday, 2011. At every ADP iteration, our simulation draws samples from the morning peak (the interval in red from 5 AM to 9 AM). The dashed lines at 4:30 AM and 10:00 AM mark the full length of the experiment. The interval [4:30, 5:00) is a rebalancing offset, whereas the interval [9:00, 10:00) is a termination offset. The former is laid out to provide extra time for vehicles to rebalance before any requests arrive, and the latter allows enough time to deliver all requests picked up during the sampling interval.*

Figure 4.3 offers a close-up on the transportation demand of the 15$^{\text{th}}$ Tuesday, highlighting the morning peak from which we draw samples. The dashed lines at 4:30 AM and 10:00 AM delineate the full extent of our experiment. During the interval [4:30, 5:00), the fleet has a thirty-minute offset (30 periods) to rebalance in order to serve the future demand. Request batches arrive every other minute in the interval [5:00, 9:00) (240 periods), and vehicles have a termination offset [9:00, 10:00) (60 periods) to make sure all requests picked up around the end of the trip sampling threshold can be delivered. The rebalance offset and the lack of requests at the end of the trip sampling interval allow us to better assess the performance of our anticipatory rebalancing method. Regarding the computation time, training and testing algorithms take on average five and two minutes, respectively, to process a single ADP iteration.

4.6.2 Model tuning

In this section, we motivate our algorithmic choices by showing their effectiveness experimentally. First, we describe the baseline scenario we use throughout the tuning process. Next, we present the spatiotemporal hierarchical aggregation structure we use to approximate value functions. Then, we highlight the effectiveness of our discount function when dealing with multiperiod travel times, show how we set up our rebalancing strategy, and describe the importance of setting a limit on the number of vehicles allowed in each node of the street network. Finally, we offer a sensitivity analysis on the maximum pickup times and base fares values.

Regarding the tuning of the ADP parameters, we have found that letting the stepsize $\overline{\alpha} = 0.1$ and the discount factor $\gamma = 1$ has led to superior performance experimentally for $I = 500$ iterations. Hence, we adopted these values across all considered scenarios.

Baseline scenario

Before we study the impact of FAV hiring and service classes, we tune our model using a baseline scenario that emulates a traditional MoD application with a fixed fleet size, homogeneous users (i.e., no service quality classes), and no service level penalties. This scenario features a fleet of 300 PAVs, which are randomly distributed throughout the street network at the beginning of each iteration. Every minute, we sample the correspondent request batch such that 10% of the requests are selected, totaling about 4,300 requests over all periods. We set up the PAV fleet size experimentally, aiming to service the sampled demand partially. We assume such an undersupplied scenario to guarantee that there are always unmet requests left to be addressed, eventually, by the freelance fleet. Additionally, following constraints (4.3), we assume there can be only five vehicles inbound to each location. Table 4.2 summarizes the baseline scenario parameters.

Table 4.2: *Parameters for the baseline scenario, featuring a fixed PAV-fleet, homogeneous users, and no service-level penalties.*

Problem characteristic	Attribute Value(s)
Fleet size ($\|K\|$)	300 PAVs
Max. #vehicles/location ($k_j^{\max}$)	5 (for all locations $j \in N$)
Base fare (p_{base})	€2.4
Distance fare (p_{time})	€1.0/km
Driving costs (c_{time})	€0.1/km
Pickup delay (w_{pickup})	10 min
Number of locations ($\|N\|$)	6,430
Period length	1 minute
Simulation length (T)	330 periods (morning peak) - 30: rebalancing offset (30 min) - 240: trip sampling (5 AM to 9 AM) - 60: finalize delivery offset (1 h)
Demand stochastic process ($\mathscr{F}^{\text{P}}$)	10% of the real-world Manhattan taxi demand on the 15$^{\text{th}}$ Tuesday of 2011 (randomly sampled)

Hierarchical aggregation levels

We determine our aggregation levels experimentally by analyzing the quality of the solutions provided by different schemes that combine both space and time. Figure 4.4 illustrates the underlying structure of our spatial aggregation configuration, showing to which regional center each location in N aggregate up.

In Table 4.3, we show the decline in the attribute space size for each aggregation level. At the most disaggregated level (i.e., $g = 0$), we consider that both FAV and PAV value functions are indexed by time and location. FAVs, in particular, are also indexed by

N (6,430)
Manhattan street
network node set

RC5 (50)
Five-minute
regional centers

RC10 (21)
Ten-minute
regional centers

Figure 4.4: *Regional center distribution on the Manhattan street network graph. Labels RC5 and RC10 identify the regional centers determined using the facility location problem formulation considering maximal service delays 5 and 10 minutes, respectively. Each location in N is connected to its respective regional center (white circle) by a red line.*

their remaining contract durations and station locations. Since considering fine-grained values for the FAV-only attributes could lead to an excessively large attribute space, we replace them with coarser substitutes. First, for the remaining contract durations, we assume values are discretized in hours. Assuming FAVs arrive in the system during the 240 one-minute trip sampling intervals (see Table 4.2), the longest contract can last four hours. Therefore, contract durations in intervals 1-60, 61-120, 121-180, and 180-240, aggregate up to one, two, three, and four remaining hours, respectively. Second, we assume the station locations aggregate up to one of the 21 ten-minute regional centers.

At aggregation level 1, we aggregate time up to three-minute intervals and locations to the closest five-minute regional center. Additionally, we stop considering FAV-related attributes, therefore using only the spatiotemporal information to index them. Finally,

Table 4.3: *Hierarchical aggregation levels. The symbol "-" indicates that the attribute is not considered.*

g	#Period	#Location	#Contract*	#Stations*	$\lvert\mathscr{T}\rvert\times\lvert\mathscr{A}^{\text{FAV}}\rvert$	$\lvert\mathscr{T}\rvert\times\lvert\mathscr{A}^{\text{PAV}}\rvert$
0	330 (t = 1 min)	6,430 (N)	4 (4h/60 min)	21 (RC10)	178,239,600	2,121,900
1	110 (t = 3 min)	50 (RC5)	-	-	5,500	5,500
2	110 (t = 3 min)	21 (RC10)	-	-	2,310	2,310

*Only considered for FAVs.

at aggregation level 2, we continue to aggregate time in three-minute intervals and aggregate locations up to the coarser ten-minute regional centers. At this level, we are left out with 2,310 possible attributes for each fleet type, substantially improving our ability to estimate values $\hat{v}_{ta}$ of states not yet visited.

We separate states by car type to emphasize that PAVs and FAVs are not interchangeable: FAVs are expected to work harmonically with PAVs as a backup fleet. The marginal value of an FAV at a certain time and location differs from its PAV counterpart, not only because it depends on the contract duration and station location attributes, but mainly because FAV operations entail a lower profit margin to the platform which consequently leads to lower value functions.

Effectiveness of VFA discount function

We assess whether our discount function is able to produce high-quality value function approximations by disabling it and allowing vehicles to rebalance to increasingly farther distances. We assume vehicles can rebalance to eight regional centers determined using one-, five-, and ten-minute maximal service delays. Accordingly, we label these experiments as 8xRC1, 8xRC5, and 8xRC10, and add an extra label [P] to indicate the cases where we apply the discount function. Therefore, in all test cases featuring the label [P], rebalancing leads to penalties proportional to the trip duration. We benchmark these results against a simple rebalancing procedure where vehicles can only move to their adjacent neighbors. Since traveling to these neighbors in the street network graph is guaranteed to take less than thirty seconds, no multiperiod travel times are incurred. Figure 4.5 shows that for all three rebalancing strategies considered, applying the discount function leads to superior results, with the 8xRC1[P] rebalancing configuration having the highest profits and percentage of serviced users by the 500$^{\text{th}}$ iteration.

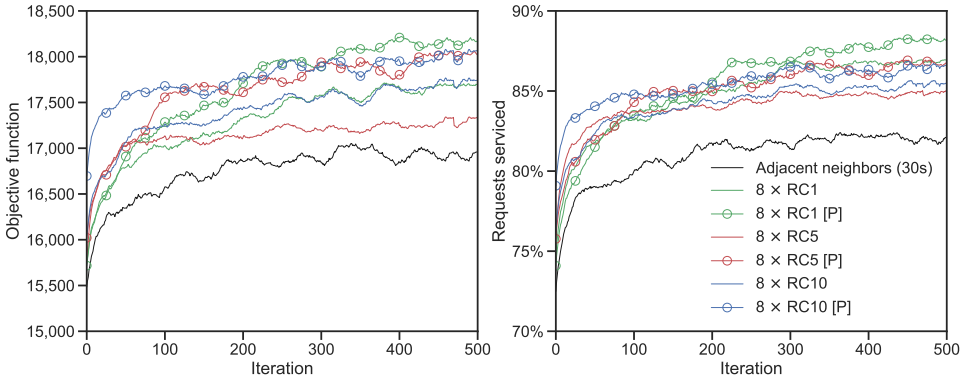


Figure 4.5: *Performance comparison of rebalancing strategies when using the discount function (represented by a label [P]).*

Rebalancing configurations

We study rebalancing configurations in which vehicles can move to a subset of increasing distant regional centers besides its adjacent neighboring locations. Notably, in Figure 4.5, the 8xRC10[P] configuration allows high-quality results at the beginning of the simulation (first 200 iterations) but is later surpassed by the 8xRC1[P] configuration. The performance of the long-distance configuration (8 × RC10[P]) is inferior to its counterparts because rebalancing to farther ten-minute region centers prevents vehicles from consistently measuring a greater range of states, although it allows them to escape from low-demand areas faster initially. Therefore, rebalancing to one-minute region centers offers a more balanced trade-off between exploration and exploitation, since vehicles can visit more locations ($|RC1| = 758 \approx 12\%$ of node set N) and bypass the intricate complexity of the real-world street networks.

In order to assess whether we could benefit from combining the short-distance 8xRC1 configuration with medium- and long-distance rebalancing movements, we created the following configurations:

- *Short + Medium (8 × RC1 + 4 × RC5)* – Rebalance to eight one-minute region centers and four five-minute region centers.
- *Short + Long (8 × RC1 + 4 × RC10)* – Rebalance to eight one-minute region centers and four ten-minute region centers.
- *Short + Medium + Long (8 × RC1 + 4 × RC5 + 2 × RC10)* – Rebalance to eight one-minute region centers, four five-minute region centers, and two ten-minute region centers.

Figure 4.6 shows that configuration 8 × RC1 + 4 × RC5 (i.e., adding four five-minute region centers to the rebalancing pool of eight one-minute region centers) results in higher performance than the 8xRC1 configuration initially while having comparable convergence behavior after the 400th iteration. However, since the rebalancing configurations perform similarly at the end of the training iterations, we select 8 × RC1 as the default rebalancing configuration. We do so, mainly because this configuration requires less processing time than its counterparts, once fewer targets are considered.

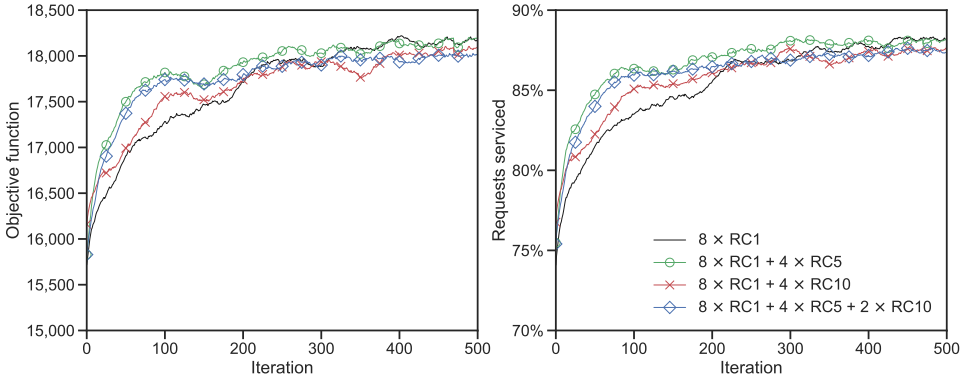


Figure 4.6: *Performance comparison of rebalancing strategies combining short-, medium-, and long-distance targets.*

Setting the maximum number of vehicles per location

Our baseline scenario considers that no more than five vehicles can be inbound to any location, according to constraints (4.3). To investigate how much these constraints contribute to generating high-quality value function approximations, we run the test cases 8xRC1 and 8xRC1+4xRC5, allowing that an unlimited number of vehicles move to each location. Figure 4.7 shows that besides reaching a subpar performance initially, disabling the maximum number of vehicles/location constraints is a great source of instability as the experiment progresses. Vehicles end up rebalancing in troves to locations associated with high-value function approximations, producing a rather unrealistic scenario in our problem setting, where locations correspond to GPS coordinates in a Manhattan street segment. Figure 4.8 shows that allowing that up to five vehicles are inbound to each location achieves the best performance for our baseline scenario and rebalance configuration 8xRC1+4xRC5.

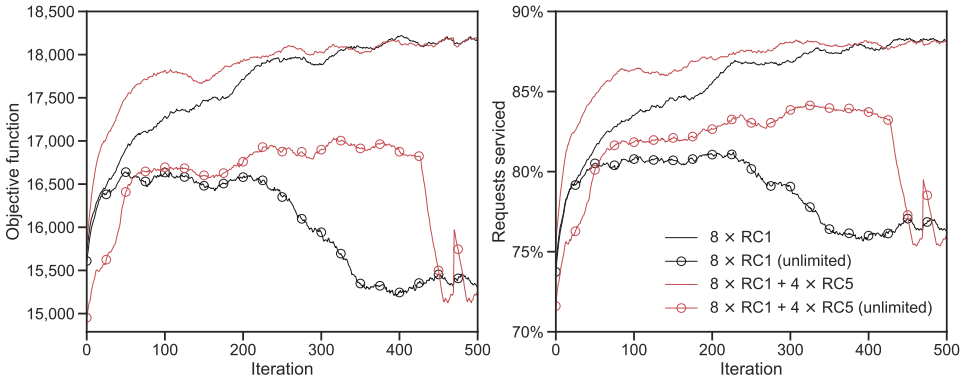


Figure 4.7: *Effect of allowing an unlimited number of vehicles at each node for rebalancing strategies 8xRC1 and 8xRC1+4xRC5.*

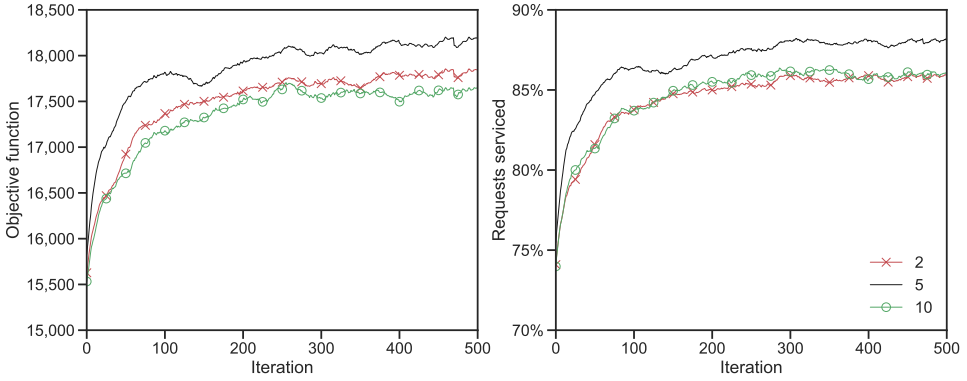


Figure 4.8: *Performance of rebalancing configuration 8xRC1+4xRC5 when allowing that at most two, five, and ten vehicles are inbound to each location.*

Base fare values and service levels

We also analyze the impact of base fare values and pickup delays on the overall performance. While base fare values directly influence the scale of value functions, the maximum pickup delays limit the matching radius of vehicles. Table 4.4 presents the average results of our testing instances considering our baseline scenario under nine different combinations of maximum pickup delays and base fare values. Apart from the average number of requests serviced, pickup delay, and the objective function, it also shows the average trip distance of both serviced and rejected users as well as the share of the fleet total time spent parked, rebalancing, picking up, and carrying users. Since we consider 330 periods and a 300-vehicle fleet, this total fleet time corresponds to 990,000 periods (300×330).

Increasing base fares makes the contribution accumulated via distance rates more and more irrelevant, as indicated by the increment in the average trip distance of rejected requests. Therefore, adopting high base fares create a bias towards short-duration trips as indicated by the decrease of both the share of the time picking up users and their average trip distances. Accordingly, vehicle rebalancing also raises, once vehicles tend to return more frequently to high demand areas. As for the influence of higher maximum pickup delays, increasing delays from five to ten minutes can result in about a 10% increase in the number of requests serviced. Such an increase, however, is moderate when we contrast ten- and fifteen-minute delays (about 2 percentage points). This result suggests that, for the fleet size we have set, it is unlikely that increasing pickup delays even further will lead to more pickups. Since we consider the decision to pick up or reject a user is taken within a single period, eventually, there are not enough vehicles to fulfill the demand, regardless of how long users are willing to wait.

4.6.3 Platform fleet management

In this section, we illustrate the behavior of our π_{VFA} policy. Besides the baseline parameters described in Table 4.2, we consider the best tuning settings achieved in Section 4.6.2, namely, the three hierarchical aggregation levels presented in Table 4.3, the five-vehicle limit per location, and the rebalancing strategy 8 × RC1. Figure 4.9 and Figure 4.10 compare the performance of the proposed VFA policy against the myopic policy on a single testing instance. Since the myopic policy reacts to request rejection, from Figure 4.9, we can see that the fleet can fulfill the demand entirely until about 6:45 AM, when the first rebalancing movement appears. In contrast, under our VFA policy, most vehicles are rebalancing before 6:30 AM. As can be seen in Figure 4.10, the π_{myopic} rejects fewer users than π_{VFA} until about 7:30 AM, but from this time on, the π_{VFA} outperforms the myopic approach, ultimately resulting in about 14% more users serviced. The difference between the policies is further highlighted in the busiest period (from 8 AM to 9 AM). The myopic policy reacts immediately to the demand peak, picking up as many users as possible, disconsidering the future outcome of these decisions. In contrast, under the VFA policy, caring about post-decision outcomes makes many vehicles to stay parked or rebalance, which may result in some rejections initially, but leads to higher service rates in the long-run.

Table 4.4: *Impact of maximum pickup delays and base fare values on the solution quality considering the baseline scenario. Each value corresponds to an average of the results achieved for the 51 testing instances.*

Max. delay (min)	Base fare (€)	Requests serviced	Pickup delay (min)	Objective function (€)	Trip distance (km)		Fleet total time / Status			
					Serviced	Rejected	Rebalancing	Picking up	Carrying	Parked
5	2.4	76.88%	2.49	15,305	2.95	3.32	10.36%	7.92%	28.11%	53.60%
	4.8	75.49%	2.50	22,200	2.88	3.51	11.73%	7.82%	26.94%	53.51%
	9.6	78.19%	2.49	38,190	2.81	3.83	12.91%	8.07%	27.29%	51.73%
10	2.4	85.74%	3.72	16,834	2.89	3.94	8.64%	13.17%	30.73%	47.47%
	4.8	87.77%	3.72	25,540	2.80	4.77	10.31%	13.52%	30.48%	45.69%
	9.6	89.34%	3.86	43,531	2.77	5.34	8.83%	14.29%	30.69%	46.19%
15	2.4	87.77%	4.35	17,335	2.93	3.80	7.98%	15.75%	31.94%	44.33%
	4.8	89.35%	4.40	26,095	2.83	4.81	8.43%	16.24%	31.40%	43.94%
	9.6	89.71%	4.47	43,594	2.77	5.46	9.88%	16.57%	30.79%	42.76%

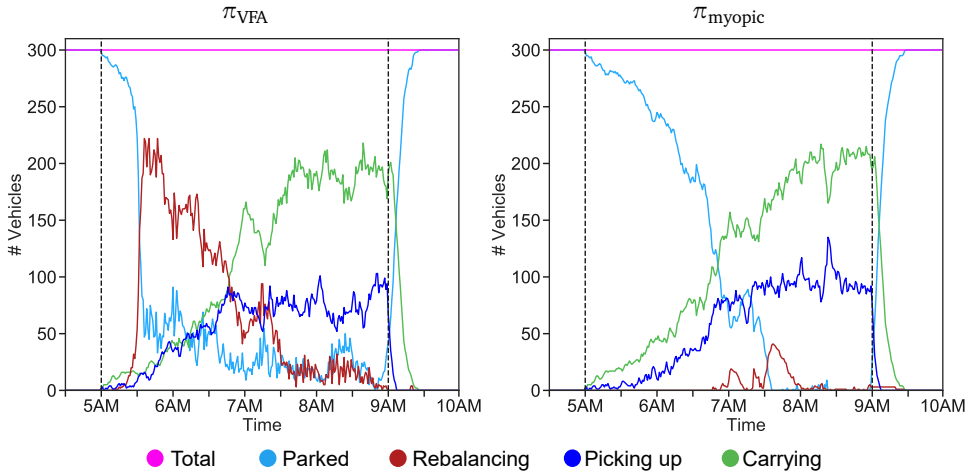


Figure 4.9: *Comparison of the number of PAVs by state (parked, rebalancing, picking up, and carrying passengers) for each policy on a single testing instance.*

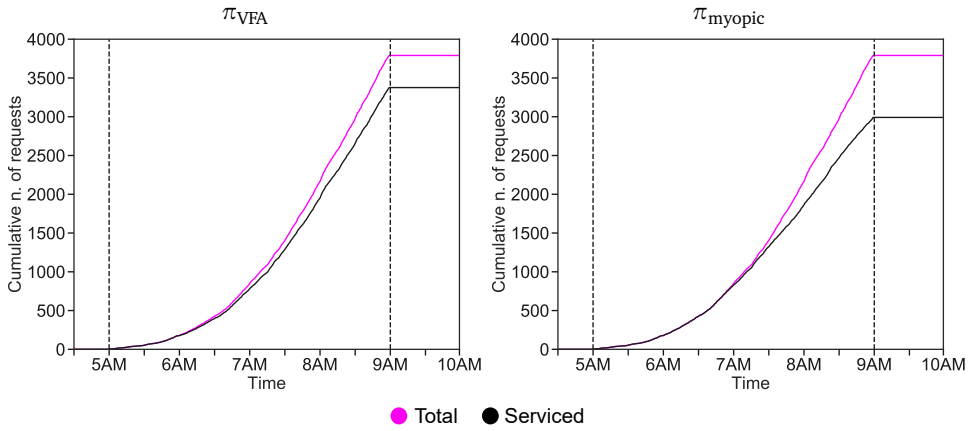


Figure 4.10: *Comparison of the cumulative number of requests serviced throughout all time steps on a single testing instance. The VFA policy leads to an 89.18% service rate, whereas the myopic policy can reach a 78.17% service rate.*

Figure 4.11 further illustrates where vehicles are likely to move to, based on the dimension of the value functions exploited by π_{VFA}. For each location in N, we average the estimates across thirty-minute intervals from 6 AM to 9 AM. As can be seen from Figure 4.11, having more vehicles in the middle section of Manhattan between 6:30 AM and 7:30 AM is prone to lead to higher contributions. This period is consistent with the predominance of rebalancing operations shown in Figure 4.9. After 7:30 AM, VFAs get lower and lower, although demand is the highest. As demonstrated by Al-Kanj et al. [1] for a similar AMoD setting, value functions monotonically decrease with time, since they reflect the expected revenue vehicles can accrue until the end of the time horizon. Hence, as time goes on, vehicles have less time to pick up users and make profits.

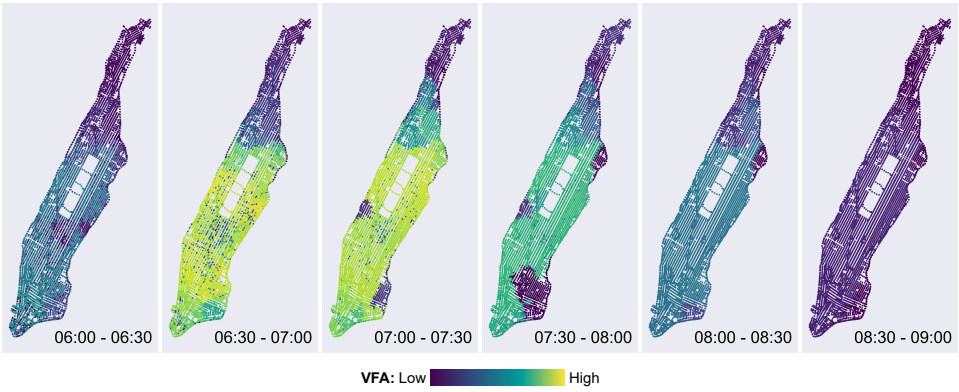


Figure 4.11: *Average value function approximations (VFAs) for each location in the street network graph across sequential thirty-minute intervals. Value functions are the highest in the middle section of Manhattan from 6 AM to 7:30 AM, shortly before the demand reaches its peak.*

4.6.4 Enforcing service level contracts

In this section, we build upon the baseline scenario such that service level violation penalties are taken into consideration. We show how the penalization mechanisms, namely, the tolerance delays and rejection penalties, can lead to a higher service rate while compensating users who have had their service levels violated. First, regarding the service level preferences, we assume first-class users (SQ1) can wait at most five minutes to be picked up, whereas second-class users (SQ2) can wait at most ten minutes. Proportionally, we assume that the base fare of SQ1 users is twice the SQ2 base fare, such that $p_{\text{base}}^{SQ1} = 4.8$ and $p_{\text{base}}^{SQ2} = 2.4$. One should notice that the parameters defined for SQ2 users coincide with those used in the tuning.

As for the penalty parameters, we consider five-minute tolerance delays for both classes and rejection penalties $\rho \in \{0,1,2\}$. For $\rho = 0$, we have a scheme where only delay penalties are incurred, whereas, for $\rho \in \{1,2\}$, rejection penalties are equivalent to one and twofold the user base fares. Finally, we also analyze the impact of these penalties when servicing users from SQ1 and SQ2, both separately (scenarios A1 and A2) and combined (scenario A3). In A3, the service class distribution follows a stochastic process where first-class user locations and request times coincide with the 20% most generous tippers (among tipping users) of the Manhattan demand occurring between 5 AM and 9 AM. To create this distribution, we first aggregate all requests from the taxi demand considered (all 2011 Tuesdays) according to their location and placement time (within five-minute bins). Next, we assign first-class labels to all requests whose tip/fare ratio ranks over the 80^{th} percentile, which is around 0.26. Then, we determine for each location and time bin pair the ratio of first-class requests, which we consider as the probability of them appearing. Table 4.5 summarizes the parameters that we use to build upon the PAV baseline scenario to assess the impact of enforcing service level contracts.

Table 4.5: *Summary of the parameters used to enforce service level contracts on different user bases.*

Problem characteristic	Attribute Value(s)
Classes (C)	{SQ1, SQ2}
Max. pickup delay class c (w^c_{pickup})	SQ1 = 5 min / SQ2 = 10 min
Waiting tolerance class c ($w^c_{\text{tolerance}}$)	SQ1 = 5 min / SQ2 = 5 min
Penalty factor (ρ)	{0, 1, 2}
Base fare (p^c_{base})	SQ1 = €2.4 / SQ2 = €4.8
User base	*Scenarios:* [*A*1] Only SQ1 users [*A*2] Only SQ2 users [*A*3] The 20% highest tippers are SQ1

Sensitivity analysis of penalization schemes

In Table 4.6, we show for the homogeneous user base scenarios A1 and A2 to what extent manipulating the penalization scheme alters the average performance of the fleet from both user and platform perspectives. For comparison, in the top row of each user base, we place the results for instances similar to those presented in Table 4.4, in which neither delay nor rejection penalties are applied.

Although in practice, the same maximum delays are considered (i.e., ten and fifteen minutes), applying tolerance delays alone (i.e., $\rho = 0$) leads to faster pickups for both SQ1 and SQ2 classes. Since any time spent within the tolerance delay offsets the base fare values, the π_{VFA} policy ends up incorporating a greater sense of urgency. From the perspective of the provider, such a penalty mechanism enables improved user service levels at the expense of slightly lower total contributions. This tradeoff is more prominent for the user base A1, in which pickup delays decreased in 41 seconds while increasing in 0.25 percentage points the number of serviced requests, at the expense of €1,287 fewer profits. Moreover, a close analysis of the fleet total time indicates that the tolerance delays remarkably impact the fleet management strategy to service A2, since vehicles spend more time rebalancing and less time parking. These relations suggest that tolerance delays help to achieve more accurate VFAs, which adequately and quickly drive vehicles to the most promising areas.

However, sole adopting tolerance delays only improves the ride experience of serviced users, compensating them according to the inconvenience inflicted. A true commitment to SLCs have to also adequately compensate those who have been through the greatest possible inconvenience, namely, service rejection. By making up for rejections, platforms can improve customer loyalty, once users can trust the transportation provider genuinely strives to keep consistent service quality, to the point of having "skin in the game" (i.e., risking company profits). Our results show that, besides providing such a guarantee, the application of rejection penalties can also increase the number of requests serviced, with vehicles spending more time rebalancing and less time parked. High penalty factors, however, creates a rejection bias against long distance requests (see the increase in the mean trip distance associated with rejections). Conversely, the trip distance of serviced requests decreases, indicating that the fleet management strategy consists of fulfilling short trips and quickly rebalancing back to high-demand areas.

Table 4.6: *Sensitivity analysis on penalization schemes. Top lines feature results for comparable configurations where no penalties are applied (see Table 4.4). Performance markers consist of the average results achieved by applying our π_{VFA} policy on the 51 testing instances.*

Base fare (€)	Max. delay (min)		Rej. pen. (ρ)	Requests serviced	Pickup delay (min)	Objective function (€)	Trip distance (km)		Fleet total time / Status			
	Pickup	Tolerance					Serviced	Rejected	Rebalancing	Picking up	Carrying	Parked
No penalties												
4.8	10	-	-	87.77%	3.72	25,540	2.80	4.77	10.31%	13.52%	30.48%	45.69%
Delay and rejection penalties (user base A1)												
4.8	5	5	0	88.02%	3.31	24,253	2.85	4.39	6.31%	12.05%	31.17%	50.47%
			1	88.94%	3.48	21,584	2.79	5.05	6.43%	12.83%	30.81%	49.93%
			2	88.91%	3.56	19,009	2.77	5.23	8.10%	13.11%	30.53%	48.26%
No penalties												
2.4	15	-	-	87.77%	4.35	17,335	2.93	3.80	7.98%	15.75%	31.94%	44.33%
Delay and rejection penalties (user base A2)												
2.4	10	5	0	87.76%	4.32	16,956	2.94	3.77	9.07%	15.65%	31.96%	43.32%
			1	89.13%	4.40	15,776	2.83	4.74	8.61%	16.19%	31.34%	43.85%
			2	89.25%	4.50	14,489	2.80	5.10	10.28%	16.57%	30.95%	42.20%

Ultimately, our findings suggest that both measures are effective to improve service quality, such that we incorporate them in our standard setup. Hence, we adopt the five-minute tolerance delays and rejection penalties equivalent to the base fare (i.e., $\rho = 1$), since these offer a more balanced trade-off regarding users' trips distances. To illustrate how this scheme works in the current transportation setting, in the following, we exemplify how the service provision unfolds for a regular SQ1 user. First, in case the request cannot be fulfilled, the platform warns the user (within one minute) and compensates him immediately a rejection penalty equivalent to the base fare. Otherwise, when the user can be serviced, a vehicle takes in average 3.48 min to pick up him. When the waiting time surpasses the five minute threshold, a fraction of the base fare is discounted from the user's total trip cost, proportional to the waiting time in the tolerance interval.

4.6.5 Vehicle productivity and fleet size

Although we have demonstrated that our penalization scheme can improve PAV-fleet productivity and user service levels, Table 4.6 shows that the platform still cannot service about 10% of the users. Our results indicate that this inability to cover the demand entirely is due to insufficient vehicle supply. As can be seen in Figure 4.9, under our VFA policy, most vehicles are busy (i.e., rebalancing, picking up, or carrying users) during the demand peak. When rejections start to accumulate from about 6:30 and on (see Figure 10), we can see that the number of parked vehicles drop dramatically, especially in the myopic policy. In such a scarcity scenario, vehicles tend to reject users whose trips are not economically efficient. Typically, a vehicle is better off parking in high-demand areas than traveling to pick up users in low-demand areas, associated with unpromising future returns. This fleet management strategy can also be seen during the busiest period in Figure 4.9, which features two "idleness peaks" (at around 7:15 and 8:30) where about fifty AVs are parked, waiting for future requests.

4.6.6 Freelance fleet management

In this section, we show how a third-party-owned fleet of FAVs can complement the PAV-fleet to improve user service levels. First, we describe how we model the uncertainty associated with the freelance fleet availability and then we assess the outcome of hiring FAVs.

Modeling FAV availability

We assume both announcement times and contract durations are drawn from a truncated normal distribution $\psi(\overline{\mu}, \overline{\sigma}, a, b; x)$, where $\overline{\mu}$ and $\overline{\sigma}$ are the mean and variance of the normal distribution, whereas a and b specify the truncation interval. Since our study draws on Manhattan's demand, we also harness the daily commuting patterns of the island to establish realistic announcement times. We consider FAVs arrive between 5 AM and 9 AM, reaching an arrival peak at 8 AM. This arrival pattern is adapted from the time workers leave home to go to work in Manhattan (see Table 4.7), where most departures (54.60%) occur between 7 AM to 9 AM.

Table 4.7: *Time leaving home to go to work in Manhattan [110].*

Time leaving home	Workers
12:00 AM to 4:59 AM	1.10%
5:00 AM to 5:29 AM	1.40%
5:30 AM to 5:59 AM	1.10%
6:00 AM to 6:29 AM	4.10%
6:30 AM to 6:59 AM	4.60%
7:00 AM to 7:29 AM	9.70%
7:30 AM to 7:59 AM	10.20%
8:00 AM to 8:29 AM	20.10%
8:30 AM to 8:59 AM	14.60%
9:00 AM to 11:59 AM	33.00%

Regarding the contract durations, we investigated two scenarios. First, in scenario D1, vehicles are available until the end of the trip sampling interval at 9 AM. Second, in scenario D2, contracts can last from 1h to 4h (viz., trip sampling interval) and most FAVs are made available for 2h, resulting in the distribution $\psi(2\text{h}, 1\text{h}, 1\text{h}, 4\text{h}; x)$. We generate these contract durations in tandem with announcement times, adjusting durations that surpass the maximum simulation time when added to their announcement times. For this reason, contracts in the range [1h, 1.5h] become more common since FAVs arriving after 8:30 AM have maximum contract durations of 1.5 hours.

Regarding the spatial distribution of these vehicles over the map, we investigate two deployment scenarios with increasingly higher numbers of stations $O \subseteq N$:

- *Clustered* [C] – Stations are drawn from 1% distinct randomly chosen locations ($|O| \leq 64$). In this scenario, AVs cruise to park in a small set of parking lots (e.g., due to incentives, city regulations).
- *Scattered* [S] – Stations are drawn from all available locations ($|O| \leq 6{,}430$). This scenario simulates the behavior of AVs which park nearby their owners' locations.

We assume that across all iterations the station location set O remains stable for all deployment scenarios. Thus, under scenario C, for instance, FAVs always start from the same set of 64 nodes.

Table 4.8 summarizes the parameters governing an operational scenario in which the fleet is comprised of PAVs and FAVs. This scenario extends our baseline scenario by allowing extra 200 FAVs into the platform, distributed according to the availability settings mentioned earlier.

Improving service quality with on-demand hiring

In this section, we offer different perspectives on the results achieved when FAVs, which are available according to the parameters described in Table 4.8, join the PAV fleet to uphold user SLCs. Table 4.9 and Table 4.10 present an average performance comparison between the VFA and myopic policies on the testing data set for user base A3. Table 4.9 shows the influence of each FAV availability scenario (i.e., contract duration and station distribution combination) on the mean objective function, percentage of requests serviced, and pick up delay. Table 4.10 presents the fleet utilization breakdown, that is, the percentage of the total fleet time spent in each vehicle status.

Table 4.8: *Summary of the parameters for on-demand hiring.*

Problem characteristic	Attribute Value(s)
Profit margin (β)	100% (PAVs) and 30%(FAVs)
Fleet size ($\|K\|$)	300 PAVs + 200 FAVs
Number of stations (O)	*Distribution scenarios:* [C] Clustered - 64 (0.01*N) [S] Scattered - 6,430 (1.00*N)
FAV hiring stochastic process ($\mathscr{F}^{O}$)	*Station:* chosen at random from O *#Vehicles/station:* random *Announcement time:* ψ(8AM, 1h, 5AM, 9AM; x) *Contract duration scenarios:* [$D1$] from announcement time until 9 AM [$D2$] ψ(2h, 1h, 1h, 4h; x)

Table 4.9: *Comparison of the average objective function, number of requests serviced, and pickup delays between VFA and myopic policies on all FAV availability scenarios.*

Policy	Contract duration	Station distr.	Obj. func. (€)	Requests serviced	Delay (min) SQ1	Delay (min) SQ2
Myopic	D1	C	18,267	96.73%	3.0	4.6
		S	18,306	96.63%	3.0	4.7
	D2	C	17,628	92.66%	3.1	5.0
		S	17,587	92.16%	3.1	5.0
		No hiring	15,273	75.32%	3.2	5.0
VFA	D1	C	18,869	98.80%	3.1	4.7
		S	18,986	98.90%	3.0	4.7
	D2	C	18,811	98.45%	3.1	4.7
		S	18,809	98.43%	3.0	4.6
		No hiring	17,442	89.25%	3.1	4.4

For the sake of comparison, the bottom line of each policy in both tables presents the results achieved when hiring is not considered. As can be seen from Table 4.9, in the no-hiring scenario the VFA policy can service about 18% more requests than the myopic policy, besides providing lower pickup delays, especially for the SQ2 class.

Once hiring is enabled, over 90% of requests are picked up regardless of the policy across all scenarios. However, substantial differences can be seen between the policies when different contract durations are considered. On average, we have found that D1 contracts allow a surplus of about 6,000 more minutes of total fleet time than D2. This extra time reflects positively on the platform profits and in the number of requests serviced. While the average difference across station distribution between D1 to D2 contract durations is about 4 percentage points in the myopic policy, this difference is less than 0.5 percentage points in the VFA policy. The same pattern can be seen on the difference between SQ2 user pickup delays, which differ much dramatically across the contract durations scenarios under the myopic policy. Hence, by better managing both vehicle types, the VFA policy can sustain high service levels for all user bases, even under a more restrict FAV availability. From a different perspective, FAV owners wanting to im-

prove the odds of renting out their vehicles, have to set up service availability adequately, such that the platform has enough time to rebalance and return these vehicles.

Moreover, as confirmed by the total fleet time breakdown in Table 4.10, FAVs tend to stay idler under the proposed VFA policy. Although not highlighted by the objective functions due to our low-cost setup, this characteristic is crucial for providers, especially in the light of vehicle automation, when induced demand due to ease of use may play a significant role. City managers are increasingly concerned about traffic, and proposals for imposing congestion charges abound. Therefore, a platform owner is generally better off using fewer vehicles, especially FAVs, which need to spend extra time returning to their origin station. Ultimately, the proposed VFA policy can find a compromise between service levels and vehicle activity, prioritizing the own fleet over outside hire to address requests.

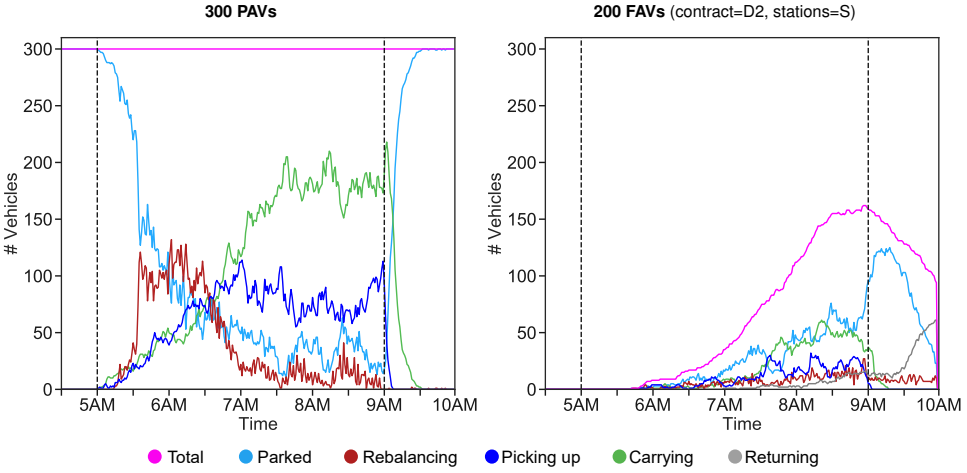


Figure 4.12: *Number of vehicles per status (parked, rebalancing, servicing passengers, and returning to station) by one-minute step separated by fleet type for a single testing instance. The total number of PAVs is constant throughout the whole time horizon, whereas the number of FAVs varies according to a stochastic process.*

Figure 4.12 further illustrates the impact of including 200 FAVs to service user base A3 on a single testing instance. FAVs arrive according to the stochastic process $\mathscr{F}^{\mathrm{O}}$ assuming contract duration scenario D2 and station distribution scenario S. It can be seen that the vehicle/status distribution still resembles the results achieved by a PAV-only fleet (see Figure 4.9), showing that the inclusion of FAVs does not disrupt the PAV-fleet operation significantly. Since we assume 70% of the profits accrued by FAVs belong to their owners, using FAVs returns fewer profits to the platform while inflicting similar operational costs. That is the main reason why the service level improvement by hiring vehicles (about 10% for the VFA policy) does not translate proportionally into the profits. However, maintaining high service levels result in increased customer satisfaction, which may improve the platform's reputation and generate a higher turnover in the long run.

Table 4.10: *Comparison of the average fleet total time per status across all FAV availability scenarios considering the VFA and myopic policies.*

Policy	Contract duration	Station distr.	Rebalancing		Picking up		Carrying		Parked		Returning	
			PAV	FAV	PAV	FAV	PAV	FAV	PAV	FAV	PAV	FAV
Myopic	D1	C	0.49%	0.65%	13.56%	14.79%	31.39%	22.36%	54.57%	54.24%	-	7.96%
		S	0.52%	0.88%	13.60%	14.76%	31.46%	21.88%	54.42%	54.23%	-	8.26%
	D2	C	0.54%	1.10%	13.64%	21.89%	31.43%	26.06%	54.39%	38.91%	-	12.05%
		S	0.57%	1.65%	13.71%	21.69%	31.49%	25.38%	54.24%	38.29%	-	12.98%
		No hiring	0.75%	-	14.17%	-	31.82%	-	53.25%	-	-	-
VFA	D1	C	8.14%	1.30%	15.08%	10.23%	31.53%	20.01%	45.25%	60.38%	-	8.08%
		S	7.34%	1.23%	14.90%	10.29%	31.61%	19.98%	46.15%	59.73%	-	8.76%
	D2	C	8.08%	1.31%	14.98%	13.10%	31.50%	24.08%	45.43%	51.15%	-	10.35%
		S	8.09%	1.12%	14.82%	12.48%	31.40%	24.23%	45.68%	50.88%	-	11.29%
		No hiring	8.57%	-	15.25%	-	31.28%	-	44.90%	-	-	-

4.7 Conclusions

This chapter answers research sub-question SQ3. We propose a solution to control service quality on an operational level using a learned-based optimization approach. We introduce a model for a dynamic and stochastic dial-a-ride problem arising on an AMoD platform that hires idle AVs to maintain consistent user service levels. Developing an approximate dynamic programming algorithm, we iteratively improve a policy to dispatch and rebalance both platform- and third-party vehicles on a real-world street network of Manhattan. The proposed policy deals with two seldomly considered sources of uncertainty, namely, (i) the spatiotemporal distribution of user service level preferences and (ii) the availability of third-party vehicles. While (i) allows providers to better address heterogeneous user expectations by rebalancing more vehicles to areas featuring high demanding users, (ii) enables the learning of routing policies that take into account when, where, and how many third-party vehicles are expected to appear throughout the planning horizon.

The proposed approach improves service quality for the ridesharing platform customers in multiple ways. First, penalizing both excessive delays and rejections following SLCs is shown to be an effective measure to increase the number of requests serviced. Second, the policy learned by sampling the demand from a particular weekday was shown to be generic enough to adequately address the demand patterns of all similar weekdays throughout a whole year. Without any hiring, such a policy consistently outperforms a reactive optimization policy, servicing on average about 18% more requests. Moreover, although both policies manage to service most requests when hiring is considered, the proposed policy has been shown to do it more efficiently, using fewer FAVs, and providing better service levels.

We conduct experiments on the historical Manhattan taxi demand considering a variety of fleet and demand configuration scenarios. Using a baseline scenario featuring only PAVs and homogeneous users, we define a hierarchical aggregation structure to approximate value functions of unvisited states. Besides time and space, the proposed layers also consider FAV-specific characteristics, such as contract duration and home station location. In particular, the spatial hierarchical aggregation structure improves existing configurations in which locations aggregate up to ad-hoc regions. We propose a minimum set covering formulation to optimally determine regions whose nodes can be accessed from a regional center within a maximal time limit. This formulation offers a more robust and versatile approach to hierarchical spatial aggregation since it automatically captures the peculiarities of any transportation network.

Optimal regional centers are also used to set up several rebalancing strategies, in which vehicles can move to a subset of neighboring centers, determined through different maximal time limits. The obtained results show that rebalancing to short-range regional centers allows vehicles to incrementally scape from perpetually low-demand areas, besides offering a good compromise regarding computational time. Since we adopt short intervals, these rebalancing movements occasionally result in multiperiod travel times (i.e., at decision time, vehicles are still acting on decisions from previous periods). We show that by actively lowering VFAs of post-decision states of farther rebalancing targets improves the performance of our solution for test cases with increasingly higher rebalancing distances.

Moreover, we develop a high-resolution state representation in which the spatial attributes correspond to discretized GPS coordinates (rather than grids, zones, or areas) and periods are no longer than one minute to comply with the demanding expectations of current MoD users. Such characteristics prevent our policy from incurring into infeasible (concerning the infrastructure capacity) or illegal (concerning the city regulations) decisions altogether in real-time. Ultimately, making use of the underlying street network allows us not only to comply with real-world constraints but also improves the solution quality. Our experiments demonstrate that constraining the maximum number of vehicles inbound to each intersection is crucial to achieving stable VFAs, since these constraints exempt us from modeling the behavior of nonlinear approximations.

This research can be extended in many promising directions. First, one could focus on designing an inverse formulation to determine the minimum number of company-owned vehicles necessary to complement third-party fleets available according to varying stochastic distributions. Second, the requirements of independent owners could take into consideration alternative parameters. For example, they could establish minimum profit margins or compensations to join ridesharing platforms. As a result, platforms would have to consider these parameters to achieve balanced solutions, weighing customer dissatisfaction and outsourcing costs. Additionally, by considering time travel uncertainty, service quality contracts would have to be further adapted to compensate users beyond the violations previously described. Ultimately, this uncertainty could lead to service time window violations on the supply side, such that platforms could also set up contracts prescribing compensations for inconvenienced FAV owners. Lastly, one could also consider the impact of cities' policies (e.g., congestion pricing, empty-vehicle fees, parking costs, ride subsidization) on platform operations, such that vehicles are steered to fulfilling overarching mobility optimization goals. This is the direction we take in Chapter 4, where we investigate how subsidization and penalization policies can steer vehicles to targeted areas, overriding natural rebalancing patterns arising from the transportation demand.

Chapter 5

Overcoming mobility poverty

In chapters 3 and 4, we show how providers can optimize fleet utilization and vehicle hiring in order to guarantee user service level requirements. Particularly in Chapter 4, we demonstrate that regional service levels can be improved by applying anticipatory relocation strategies that take into consideration when and where requests are more likely to appear. The nature of transportation demand, however, invariably creates learning biases towards servicing cities' most affluent and densely populated areas, where alternative mobility choices already abound. As a result, current disadvantaged regions may end up perpetually underserviced, therefore preventing all city residents from enjoying the benefits of *autonomous mobility-on-demand* (AMoD) systems equally. In this chapter, we extend the method presented in Chapter 4 to investigate how to overcome the demand patterns naturally incorporated into value functions to improve service levels of disadvantaged areas. We show for a case study on Rotterdam, the Netherlands, that the proposed method can better cater to users departing from the city's targeted areas, substantially outperforming myopic and reactive benchmark policies.

This chapter is organized as follows. In Section 5.1 we describe the shortcomings of current AMoD systems regarding service level distribution and in Section 5.2, we review the most common service quality metrics. Next, we formulate the problem in Section 5.3 and present a solving strategy in Section 5.4. Then, Section 5.5 presents our experimental study, and Section 5.6 the results regarding the effectiveness of both subsidization and penalization policies in rebalancing vehicles to targeted areas. Finally, Section 5.7 concludes the work and offers an outlook on the role of city managers in the deployment of AMoD systems. Parts of this chapter have been published in [9]:

B. A. Beirigo, F. Schulte, R. R. Negenborn. Overcoming mobility poverty with shared autonomous vehicles: A learning-based optimization approach for Rotterdam Zuid. In *Proceedings of the 11th International Conference on Computational Logistics*, pages 492-506, Enschede, the Netherlands, 2020.

5.1 Introduction

Service levels of residents from different areas of a city can vary significantly due to an uneven distribution of transport resources. Peripheral or low-income regions are typically more prone to suffer from accessibility poverty, that is, the difficulty of reaching certain key activities (e.g., employment, education, healthcare) due to mobility poverty, which is concerned with the systemic lack of transportation and mobility options [61]. Since low-income and mobility poverty are strongly correlated, offering sufficient transportation choices to disadvantaged areas can ultimately improve social equity.

Shared autonomous vehicles (SAVs) and, more generally, *autonomous mobility-on-demand* (AMoD) systems, offer an opportunity to overcome mobility poverty. Sharing services reduce the cost of personal mobility once all expenses of purchasing, maintaining, and insuring vehicles are distributed across a large user-base [98]. Recent research has demonstrated that efficient SAV fleet management can help AMoD providers fulfilling today's transportation demand using much fewer vehicles. However, typical performance measures fail to account for differences in demographics appropriately, lacking nuanced equity implications [20].

Due to natural demand patterns or deliberate profit-seeking policies, SAVs can end up re-enforcing existing inequalities by frequently moving to regions that are more prone to generate higher profits. Such a preference for affluent regions can already be identified in the current transportation landscape, where mobility options (e.g., ride-hailing, micro-mobility, ride-pooling, and transit) abound in cities' central areas.

In this chapter, we propose an anticipatory relocation strategy to rebalance a fleet of AVs to improve the mobility of targeted disadvantaged areas. Analogously to Chapter 4, this strategy consists of an *approximate dynamic programming* (ADP) algorithm that uses demand data throughout an iterative process to derive *value function approximations* (VFAs). These lookahead approximations are then considered in the optimization process to weigh the downstream impact of current decisions.

We illustrate our method using the case of Rotterdam, the Netherlands, where the northern region (Rotterdam Noord) encompasses the entire city center, outperforming the southern (Rotterdam Zuid) in a range of socio-geographical factors, such as income and transport connectivity. To improve the mobility of the residents in the Zuid region and ultimately their access to key activities, we investigate to which extent ride subsidization and rejection penalties can contribute to overall fairness, adequately driving vehicles to underserviced areas.

We consider a first-mile case study in which users request vehicles from a private AMoD provider to access the closest train station (see Figure 5.1). This setup is particularly relevant for the deployment of *mobility-as-a-service* (MaaS) solutions, which are based on the integration of different transport services. We show that a proper cost scheme setup can overcompensate the rebalancing bias towards densely populated and high-income areas improving mobility choices in underserviced areas. Ultimately, our results help city managers to understand the cost of laying out equitable transportation policies that balance providers' profitability and the service levels of underserviced users.

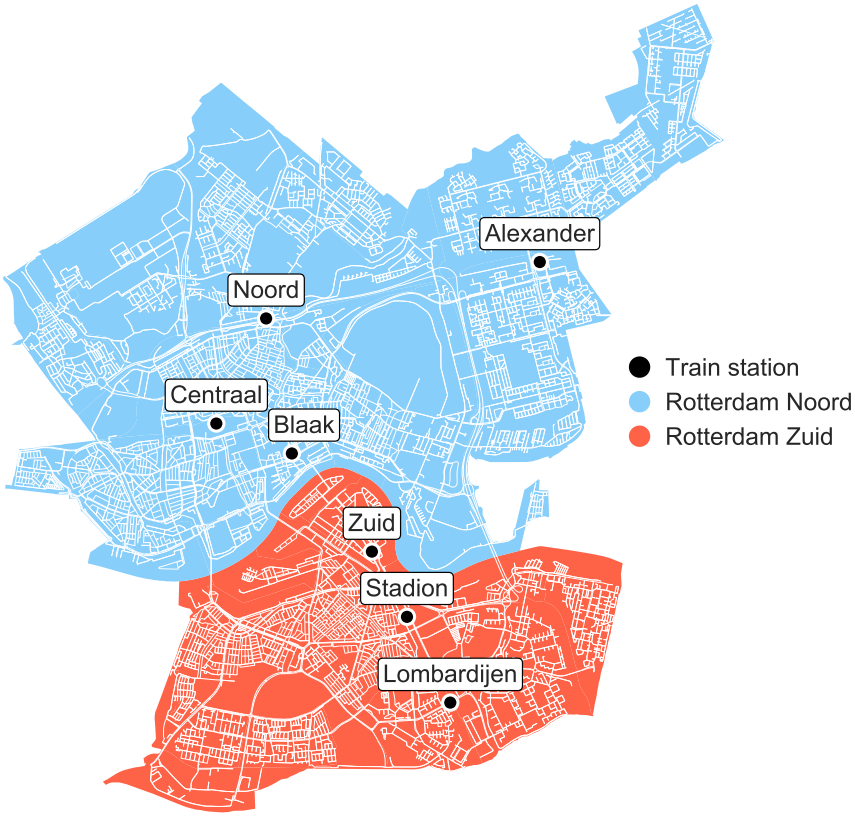


Figure 5.1: *Rotterdam regions (Noord and Zuid) and the seven train stations that are the destinations of all first-mile trips.*

5.2 Related work

AMoD systems rely on rebalancing strategies to find a reasonable compromise between asset utilization and user satisfaction. Supply and demand mismatches are typically addressed using ongoing imbalance cues (e.g., request rejections, idle vehicles) and predicted demand information (based on historical data). For example, Pavone et al. [79] propose an optimal transport problem in which locations with a surplus of idle vehicles continuously send empty vehicles to locations with a shortage of idle vehicles. Similarly, Alonso-Mora et al. [2] present a reactive rebalancing approach that sends idle vehicles to undersupplied areas, which are identified by the occurrence of unsatisfied requests. Through a linear program, vehicles are assigned to the departure locations of these requests, aiming to minimize the total sum of travel times. Later, Alonso-Mora et al. [3] use past historical data to compute a probability distribution over future demand and proposes an assignment algorithm to match vehicles to future requests. Fagnant and Kockelman [34] relies on a rule to overcome supply-demand imbalances using a block-based division operational map. For each block, they compare the supply of idle stationary vehicles versus the share of currently waiting travelers plus soon expected travelers

in the near future (according to the block historical trip rate). Similarly, Chen et al. [18] use a price-based strategy, encouraging trips originating in a cell with a surplus of vehicles and penalizing trips originating in a cell with a deficit of vehicles.

Learning-based methods have also been successfully employed to enable anticipatory rebalancing. Through a *reinforcement learning* (RL) framework, Wen et al. [115] train a *deep Q-network* (DQN) using rewards based on waiting time savings of users picked up due to rebalancing movements. Conversely, penalties are applied when vehicles remain idle during the rebalancing period. Considering a grid map, they model states using grid-wise idle-vehicle distribution, in-service vehicles, and predicted demand (based on a Poisson process) in the surroundings. Gueriau and Dusparic [41] propose a decentralized RL approach based on Q-learning. They show that agents (i.e., vehicles) can contribute to global performance by learning how to optimize their own individual performance with local information only. Lin et al. [58] also takes advantage of the RL framework through a *contextual multi-agent actor-critic* (cA2C) algorithm. Their design stands out due to two main features, (i) the adoption of centralized value functions (shared by all agents), and (ii) context embedding that establishes explicit coordination among agents. Iglesias et al. [48] design a *model predictive control* (MPC) algorithm that leverages customer demand forecasts to rebalance vehicles. The forecasting model is based on a *long short-term memory* (LSTM) neural network. Al-Kanj et al. [1] use an ADP formulation that allows for anticipatory rebalancing and recharging of electric vehicles. Their approach maximizes vehicle contribution over time, using value function approximations to estimate the impact of each decision in the future.

Similarly to Al-Kanj et al. [1], we enable anticipatory rebalancing by using value functions to steer vehicles towards high-demand areas. In contrast with all proposed methods, however, we add nuance to service levels, acknowledging that users from different regions face distinct accessibility barriers to the transport system. Based on Lucas et al. [62], we consider transport accessibility primarily in terms of availability and affordability. Additionally, following Cohen and Cabansagan [20], our AMoD rebalancing policy aims to complement public transit and redistribute transport resources towards disadvantaged areas.

Table 5.1 summarizes the rebalancing literature aforementioned. References under the first column are classified according to the rebalancing method, prediction mechanism, and performance metrics specially concerned with rider satisfaction. For the sake of conciseness, alternative metrics (e.g., costs) as well as the typical service level achievement objective are left out of our comparison. Ultimately, from the user experience perspective, related literature focuses on improving service levels (i.e., decreasing pickup and/or in-vehicle delays), whereas we focus on increasing service rate (i.e., the ratio of serviced requests under minimum service level requirements) throughout different regions.

5.3 Problem formulation

We model the problem using the language of dynamic resource management (see [1], [94]), where AVs (resources) service a sequence of trip request batches (tasks) dynamically revealed at discrete-time $t \in \{0, 1, 2, \ldots, T\}$.

Table 5.1: *Rebalancing methods, prediction mechanisms, and service quality metrics of MoD applications ("-" = absent).*

Reference	Rebalancing	Prediction	Performance metric(s)
Pavone et al. [79]	ILP	-	PD
Alonso-Mora et al. [2]	LP	-	ID, PD
Chen et al. [18]	Rule	-	PD
Fagnant and Kockelman [34]	Rule	historical trip rate	PD
Alonso-Mora et al. [3]	LP	probability distribution	ID, PD
Wen et al. [115]	RL	Q-function, poisson process	PD
Gueriau and Dusparic [41]	RL	Q-function	ID, PD
Lin et al. [58]	RL	value function	-
Iglesias et al. [48]	MPC	demand forecast	PD
Al-Kanj et al. [1]	ADP	value function	-
this chapter	**ADP**	**value function**	**RSL**

Rebalancing: **RL** (Reinforcement Learning), **MPC** (Model Predictive Control), **LP** (Linear Programming), **ADP** (Approximate Dynamic Programming)
Performance metric(s): **ID** (In-Vehicle Delay), **PD** (Pickup Delay), **RSL** (Regional Service Level)

We assume all requests arrive in batch intervals of five minutes, occurring within the earliest time $t_e = 6{:}00$ and the latest time $t_l = 12{:}00$. To ensure the system has enough time to rebalance vehicles and deliver all users, we add thirty-minute offsets before t_e and after t_l, such that the total horizon $T = 84$ (i.e., 420/5).

The state of a single resource is defined by the attribute a representing the vehicle's location in the node set N of $G = (N, E)$, a strongly connected graph drawn from a section of Rotterdam, the Netherlands. The city comprises six districts and 45 neighborhoods from which we select 40 to exclude the peripheries, such that node and edge sets have sizes $|N| = 10{,}364$ and $|E| = 23{,}048$, respectively (see Figure 5.1).

By including the temporal dimension to location a, we have a_t, or the location of an SAV at time t. Let $\mathscr{A}$ be the set of all possible vehicle attributes. The state of all vehicles with the same state attribute is modeled using

$$\begin{aligned} R_{ta} &= \text{ Number of vehicles with attribute } a \text{ at time } t, \\ R_t = (R_{ta})_{a \in \mathscr{A}} &= \text{ The resource state vector at time } t. \end{aligned}$$

Each request, in turn, is modeled using an attribute vector b comprised of origin and destination attributes $b_{\text{origin}}, b_{\text{destination}} \in N$. Let $\mathscr{B}$ be the set of all possible request attribute vectors. The state of all rides with the same state vector occurring at time t is modeled using

$$\begin{aligned} D_{tb} &= \text{ The number of requests with attribute vector } b \text{ at time } t, \\ D_t = (D_{tb})_{b \in \mathscr{B}} &= \text{ The request state vector at time } t. \end{aligned}$$

With the resource and request state vectors, we defined our system state vector as $S_t = (R_t,\ D_t)$. States S_t are measured before making decisions at each epoch $t \in \{0, 1, 2, 3, \ldots, T\}$. In this chapter, we consider each vehicle can realize three different types of decisions, namely, *service* a single user at a time, *stay parked* in its current location waiting to pick up users, and *rebalance* to a more promising location. Decisions are described using

$d^{\text{stay}} =$ Decision to stay parked in the current location,
$\mathscr{D}^{\text{R}} =$ Set of all decisions d to rebalance (i.e., move empty) to a set of neighboring locations,
$\mathscr{D}^{\text{S}} =$ Set of all decisions d to service a user,
$b_d =$ Trip $b \in \mathscr{B}$ covered by decision $d \in \mathscr{D}^{\text{S}}$,
$\mathscr{D} = \mathscr{D}^{\text{S}} \cup \mathscr{D}^{\text{R}} \cup \{d^{\text{stay}}\}$,
$x_{tad} =$ Number of times decision d is applied to a vehicle with attribute a at time t,
$x_t = (x_{tad})_{a \in \mathscr{A}, d \in \mathscr{D}}$.

The decision variables x_{tad} must satisfy the following constraints:

$$\sum_{d \in \mathscr{D}} x_{tad} = R_{ta} \qquad \forall\, a \in \mathscr{A} \tag{5.1}$$

$$\sum_{a \in \mathscr{A}} x_{tad} \leq D_{tb_d} \qquad \forall d \in \mathscr{D}^{\text{S}} \tag{5.2}$$

Constraints (5.1) and (5.2) guarantee flow conservation of vehicles and requests, respectively. Once we aim to distinguish regions in N, we define $U(a) : N \to L$ as a function that maps each location $a \in N$ to a discrete geographical area in L, with $L = \{\text{Noord, Zuid}\}$. Applying a decision d to a resource with attribute a at time t generates a contribution c_{tad}, such that

$$c_{tad} = \begin{cases} \textit{Service request} \\ p^a_{\text{base}} + p_{\text{time}} \cdot \Delta t_{\text{trip}} - c_{\text{time}} \cdot \left(\Delta t_{\text{pickup}} + \Delta t_{\text{trip}}\right), \\ \textit{Rebalance to neighboring areas} \\ -\, c_{\text{time}} \cdot \Delta t_{\text{rebalance}}, \\ \textit{Stay parked} \\ 0. \end{cases}$$

Contributions c_{tad} of service, rebalance, and stay decisions are comprised of

$p^a_{\text{base}} =$ Base fare of trips departing from region $U(a) \in L$,
$p_{\text{time}} =$ Time-dependent fare,
$c_{\text{time}} =$ Vehicle time-dependent costs,
$c^a_{\text{penalty}} =$ Penalty for rejecting users from region $U(a) \in L$,
$\Delta t_{\text{trip}} =$ Trip duration $\Delta t(b_{\text{origin}}, b_{\text{dest}})$ of request $b = b_d$.
$\Delta t_{\text{pickup}} =$ Pickup duration $\Delta t(a, b_{\text{origin}})$ from current location to trip b origin.
$\Delta t_{\text{rebalance}} =$ Rebalance travel duration $\Delta t(a, r)$ to neighboring location $r \in N$.

Assuming contributions are linear, the contribution function for period t discounted by the total rejection penalty is given by

$$C_t(S_t, x_t) = \sum_{a \in \mathscr{A}} \sum_{d \in \mathscr{D}} c_{tad} x_{tad} - P(S_t, x_t), \tag{5.3}$$

where

$$P_t(S_t, x_t) = \sum_{b \in \mathscr{B}} c_{\text{penalty}}^{b_{\text{origin}}} (D_{tb} - \sum_{a \in \mathscr{A}} \sum_{\substack{d \in \mathscr{D}^S \\ b_d = b}} x_{tad}). \tag{5.4}$$

Let $X_t^\pi(S_t)$ be a decision function that represents a policy $\pi \in \Pi$, which maps a state S_t to a decision x_t at time t. We aim to determine the optimal policy π^* that, starting from an initial state S_0, maximizes the expected cumulative contribution, discounted by a factor γ, over all the time periods:

$$F_0^*(S_0) = \max_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t=0}^{T} \gamma C_t \left(S_t, X_t^\pi(S_t) \right) | S_0 \right\}. \tag{5.5}$$

5.4 Algorithmic strategies

We address Equation (5.5) using the same algorithmic treatment presented in Chapter 4 (see Section 4.5). Therefore, in this section, we specifically describe the modeling of the elements concerned with the problem at hand. In Section 5.4.1, we show how we reduce the state space through hierarchical aggregation and in Section 5.4.2 we describe the elements of our rebalancing strategy.

5.4.1 Hierarchical aggregation for value function estimation

As in Chapter 4, we aggregate the state space hierarchically using the method proposed in [94]. We define three hierarchical levels experimentally, namely, 1, 2, and 3, that aggregate states both in space and time. Spatially, the node set is aggregated in hexagon bins of 0.17 km^2, 0.46 km^2, and 5.16 km^2, resulting in 1,016, 198, and 38 bins, which integrate the location sets N^1, N^2, and N^3, respectively (see Figure 5.2). Valid bins cover at least one node of Rotterdam's street network and are identified by the closest node to their geographical center. As such, we assume that the travel time between two bins is based on the shortest path between their corresponding center nodes at 20 km/h speed. We aggregate temporally by increasing the length of the periods. Given the seven-hour horizon, we assume level 1 comprises the 84 five-minute periods, whereas levels 2 and 3 include 42 and 38 periods lasting ten and fifteen minutes, respectively. Ultimately, the state-space size for each aggregation level declines from 85,344 to 8,316, and then to 1,444.

5.4.2 Rebalancing strategies

Slicing the area of network G using a hierarchy of geometric shapes allows us to infer a relation of proximity between nodes within the same region. We exploit this relation by assuming vehicles are allowed to rebalance to the center of the surrounding hexagon neighbors across all three hierarchical levels, totaling up to eighteen rebalancing options. This way, vehicles can explore increasingly farther neighborhoods, insofar as rebalancing targets become hexagon centers up in the spatial hierarchy.

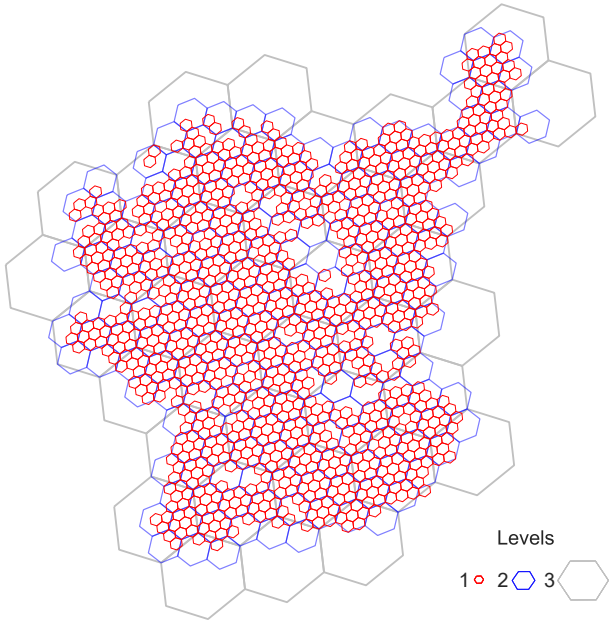


Figure 5.2: *The three spatial aggregation levels set up for the Rotterdam area encompassing the street network G. Starting from level 1, hexagon bins cover an increasingly higher number of locations of the node set N.*

Similarly to Chapter 4, we prevent vehicles from flooding high demand locations by bounding the number of vehicles they can accommodate. We consider that rebalancing trips can take place only when the cumulative number of vehicles expected to arrive or staying at node $i \in N^1$ at time t, does not surpass a limit $k^{\max}$.

5.5 Experimental study

The following sections describe the characteristics of our first-mile requests (Section 5.5.1), how we determine fleet size and distribute vehicles (Section 5.5.2), the cost schemes we evaluate to drive vehicles to Zuid (Section 5.5.3), and present two alternative policies to benchmark our method (Section 5.5.4).

5.5.1 Demand configuration

Upon receiving a five-minute request batch, the system sets up available trip decisions taking into account a pickup radius $w_{\text{pickup}} = 10$ minutes. If users cannot be accessed by any vehicle within w_{pickup}, they are immediately rejected, having to resort to another transportation means. Since we assume the AMoD system integrates a broader MaaS ecosystem, a rejection means that the users will have to rely on alternative modes to fulfill their trips. Hence, in the worst-case scenario, serviced users wait at most fifteen minutes to be picked up. In the following, we describe the process we used to create a series of spatiotemporal demand distribution scenarios.

Spatial demand patterns

To generate r request departure points, we use a weighted random process to select an origin neighborhood according to its relative population density using the Dutch census [16]. Next, we use a regular random process to select a street node within this origin neighborhood. We use this process to select 3,000 request origins, from which about one-third end up within the Zuid region. Figure 5.3 presents the probability distribution of selecting request origins across Rotterdam neighborhoods. Since we investigate first-mile trips, destination points correspond to the closest train stations of each origin point.

Temporal demand patters

Regarding the time the requests arrive at the system, we propose five scenarios in which we vary the demand patterns of residents departing from Noord and Zuid regions. We consider that the number of requests always peaks at 8:00 for the residents of Rotterdam Noord, whereas the number of requests originated in Rotterdam Zuid peak at 6:00, 7:00, 8:00, 9:00, and 10:00, leading to five request arrival scenarios labeled N8Z6, N8Z7, N8Z8, N8Z9, and N8Z10. By varying the demand peaks in Zuid, we can investigate how well the algorithm can distribute vehicles in the light of different levels of competition with the Noord demand. Moreover, the relative position between Noord and Zuid peaks allows us to assess the efficacy to which vehicles can move between regions in anticipation to demand. For instance, the earlier the demand peak in Zuid, the more time vehicles will have to move from Zuid to Noord. Conversely, the later the demand peak, the more time vehicles will have to move from Noord to Zuid. All these scenarios are modeled using a normal distribution truncated by t_e and t_l, with a standard deviation of one hour and means equal to the demand peaks entailed by each region.

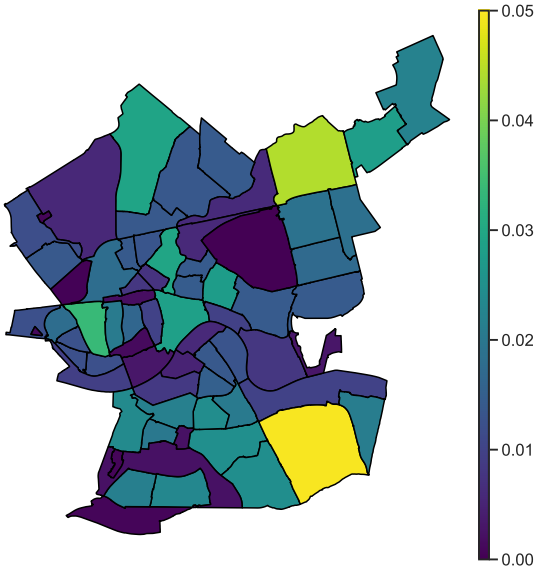


Figure 5.3: *Probabilities of choosing a departure location within each Rotterdam neighborhood based on the part-to-whole ratio of the number or residents.*

5.5.2 Fleet configuration

We determine the fleet size following the *model predictive control* (MPC) algorithm proposed by Iglesias et al. [48]. Their approach assumes perfect information throughout the whole horizon and does not allow for delays, such that, at the end of each time step, there is a sufficient number of vehicles to pick up all requests at each location. To decrease computation times, we assume that all trip origins and destinations are associated with their respective third-level hexagon centers (38 in total). We have found that the average fleet sizes achieved using the optimal MPC formulation across ten demand realizations for each demand pattern scenario range from 382.8 (N8Z10) to 499.9 (N8Z8). The results also indicated that the closer the demand peaks of Noord and Zuid regions are, the higher is the fleet size required. To deliberately create a scarcity scenario that splits the vehicle workforce between the two regions, we carry out all experiments using a 300-SAV fleet. At the beginning of each ADP iteration, we randomly distribute these vehicles throughout level-one hexagon locations. Consequently, since Noord is broader and more populated than Zuid, a service bias towards Noord will naturally emerge.

5.5.3 Cost schemes

We investigate the influence of six cost schemes on Zuid riders' service rates by manipulating both base fares and penalties. We refer to these schemes using the labels B1R0, B4R0, B8R0, B1R1, B1R4, B1R8, where B and R correspond to the p_{base} and c_{penalty} constants and the digits represent scaling factors. For instance, B1R0, B4R0, and B8R0 represent cost schemes in which the base fare p_{base} of Zuid users is one, four, and eight times higher. Regarding the values adopted, we consider $p_{\text{base}} = c_{\text{penalty}} = €2.5$, time-dependent fare $p_{\text{time}} = 1€/\text{km}$, and time-dependent operational costs $c_{\text{time}} = 0.1€/\text{km}$. Finally, we assume B1R0 is our reference cost scheme and use it for all Noord users.

5.5.4 Benchmark policies

We benchmark our ADP π_{VFA} policy against two alternative policies, namely, π_{myopic} and π_{reactive}, in which no information about the future is available. Both policies aim to maximize the cost function in Equation (5.3), but while π_{myopic} seeks only to determine the optimal vehicle-request assignment represented by Equations (5.1) and (5.2), the π_{reactive} policy relies on an additional vehicle rebalancing phase. The rebalancing is based on the state-of-the-art algorithm proposed by Alonso-Mora et al. [2], which consists of a linear program where idle vehicles are optimally rebalanced to under-supplied locations. Ultimately, this program aims to minimize the total travel distance of reaching the pickup locations of unassigned requests while guaranteeing that either all vehicles or all requests are assigned.

5.6 Results

We implemented our approach using Python 3.6 and Gurobi 8.1. Test cases were executed on a 2.60 GHz Intel Core i7 with 32 GB RAM. For all case studies, we run our ADP algorithm throughout 1,000 iterations, considering stepsizes $\overline{\alpha} = 0.1$, and maximum vehicle count $k^{\max} = 5$, which have been found to show good performance experimentally. Since we aim to improve mobility for Zuid users, our analysis focuses mainly on the service rate achieved for each region. Still, provided that users can be picked up timely, delays are already bound to w_{pickup}.

Table 5.2 presents the average service rates (across ten demand realizations) for each policy π and cost scheme configuration. We separate the service rates between users departing from each region to evaluate the effect of the proposed cost schemes on driving vehicles towards Zuid. The results are subsumed under three categories, namely, "Baseline", "Base fare," and "Rejection penalty." The "Baseline" category comprises the averages achieved using the myopic and reactive policies, which we use to benchmark the performance of our learning-based method. As hypothesized, when equity concerns are disregarded, service rates differ markedly between regions, being consistently higher in Noord.

It can be seen from Table 5.2 that π_{reactive} can already significantly improve the service rate in Zuid, servicing about 10% more requests than π_{myopic} in all demand scenarios. Since the reactive rebalancing policy relies on trip rejection stimuli to move vehicles to undersupplied areas, the fleet is disproportionately driven to Zuid. However, π_{reactive} still leads to a moderate service bias towards the Noord region, regardless of the demand scenario.

We separate the results of our proposed π_{VFA} policy according to the main feature entailed by each cost scheme. Hence, the figures subsumed under the "Base fare" and "Rejection penalty" categories highlight the effect of scaling up fares and penalties, respectively. It is worth noting that our π_{VFA} policy performs better than the baseline policies, even for cost scheme B1R0, in which no scaling is considered.

Table 5.2: *Average ratio of serviced users departing from Noord and Zuid regions for each policy, demand scenario, and cost scheme. Figures correspond to the mean average of ten demand distributions over 3,000 requests.*

Policy(π)	Cost scheme	Origin in Noord					Origin in Zuid				
		N8Z6	N8Z7	N8Z8	N8Z9	N8Z10	N8Z6	N8Z7	N8Z8	N8Z9	N8Z10
Baseline (no penalties, default base fare)											
myopic	B1R0	.783	.788	.795	.790	.795	.637	.668	.688	.694	.682
reactive	B1R0	.807	.846	.850	.852	.853	.778	.775	.783	.790	.773
Base fare (subsidization schemes)											
VFA	B1R0	.926	.905	.908	.939	.963	.812	.876	.890	.904	.949
	B4R0	.892	.870	.871	.903	.933	.888	.923	.944	.956	.967
	B8R0	.883	.858	.828	.847	.925	.905	.946	.969	.976	.980
Rejection penalty (penalization schemes)											
VFA	B1R1	.903	.885	.886	.908	.949	.848	.919	.921	.935	.940
	B1R4	.903	.848	.863	.892	.919	.881	.957	.948	.966	.968
	B1R8	.874	.842	.812	.825	.899	.903	.961	.972	.956	.980

The performance improvement is especially remarkable when demand peaks from Noord and Zuid are far apart, for example, in scenarios N8Z6 and N8Z10. These scenarios provide enough time for vehicles to rebalance in anticipation from one region to the other instead of reacting to imbalances on short notice. Moreover, during the thirty-minute rebalancing offset previous to the requests' arrival, vehicles also can harness the value functions to reach areas where users are more prone to appear. This explains how even the competitive scenario N8Z8 could benefit from using the π_{VFA} policy.

To investigate the trade-off between profits and Zuid service rate, we average the results of our VFA policy for each cost scheme over all the demand scenarios. Then, we compare the schemes' averages against the averages obtained for the reference cost scheme B1R0. Average profits are determined in terms of base fare accumulation, considering the number of requests departing from each region and the ratios of serviced users in Table 5.2. Cost schemes B4R0 and B8R0 lead to 104.8% and 249.6% higher profits to service about 5.6% and 7.8% more Zuid users over B1R0. In contrast, cost schemes B1R1, B1R4, and B1R8 incur 3.8%, 9.3%, and 16.7% losses compared to B1R0 average profit, to service about 3.0%, 6.5%, and 7.7% more Zuid users. For both cost scheme categories, the results indicate that further scaling up incentives or penalties is prone to diminishing returns once less and less service rate gains in the Zuid region can be seen.

Throughout all instances considered, scenario N8Z6 has consistently presented the lowest service rates. Since Noord starts with more vehicles than Zuid, and the demand peak in Zuid occurs earlier, this scenario necessarily demands that vehicles move from Noord to Zuid. With most requests happening at 6:00, most rebalancing operations have to be performed within the rebalancing offset. Considering that vehicles can start from remote parts of Noord, the results indicate that the thirty-minute offset is insufficient to rebalance all necessary vehicles to Zuid adequately. As the demand peak in Zuid is pushed forward (e.g., N8Z7), vehicles have more time to access Zuid user origins, and service rates increase.

5.7 Conclusions

This chapter addresses research sub-question SQ4. In contrast with Chapter 2, where we were concerned with guaranteeing that vehicles could physically access the whole transportation network, in this chapter, we aim to improve people's ability to access key life opportunities through mobility systems. We presented a learning-based fleet rebalancing method that determines a compromise between company revenues and social equity between two distinct regions of Rotterdam, the Netherlands. Our anticipatory rebalancing strategy caters to the needs of targeted regions, compensates for biases towards more affluent and densely populated regions, and mitigates mobility poverty in disadvantaged areas. Based on a range of cost schemes, we show the tipping point at which cost manipulation can affect value function approximations enough to influence the rebalancing process. Ultimately, our approximate dynamic programming algorithm achieves superior service rates compared to myopic and reactive strategies, regardless of cost scheme and demand scenarios considered.

Our results illustrate how the public sector could work in tandem with private providers to guarantee that new mobility solutions consider the patterns of disadvantaged populations. In this way, as mobility technologies develop, private market innovation can be steered to achieve social equity goals, such as preventing mobility poverty. It is worth noting, however, that adequately fulfilling such goals depends on further policy making. Since the private sector is at the forefront of the development of new mobility systems, the public sector could care for their deployment and has to invest in incentives for the use of these systems in the broader scheme of city transportation. For instance, new services could integrate existing MaaS frameworks, complementing other transportation options (e.g., transit, walking, and cycling).

Future research will focus on designing equity-aware rebalancing strategies to increase service rates based on alternative census tract information (e.g., age, gender, income) rather than the departure location alone. Additionally, to improve the effectiveness of location-based equity policies, regional transport accessibility can be further investigated, for example, in terms of transit infrastructure, delays, and availability.

Up to this chapter, we focused on optimizing mobility service for passenger transportation only. As a result, regardless of the optimization strategy used to improve fleet productivity, vehicles will inevitably be idle at low-demand periods, most notably during the early morning. Ultimately, fitting transportation services to a single type of demand bounds providers' profitability, making them inescapably dependent on this demand's particular patterns. However, once we consider the broader ecosystem AMoD systems are expected to work in, alternative opportunities for deploying more flexible transportation systems arise. As cars become commodities and transportation systems increasingly focus on offering high-quality rides, providers can explore more lucrative vehicle setups that can accommodate different demand types. In Chapter 6, we model an AMoD system where vehicles can handle both people and freight demands interchangeably. We show that providers can improve profits by exploiting the intrinsic service quality requirement differences between the two demand types.

Chapter 6

Integrating people and freight transportation

Throughout the last chapters, we have demonstrated how AVs can fulfill user service level requirements. So far, these requirements have been concerned with the needs of a single type of commodity, namely, passengers. Their demand patterns alone dictate the fleet productivity, invariably leading to suboptimal fleet utilization. In Chapters 3 and 4, we already proposed strategies to remedy low fleet productivity. Instead of owning the entire fleet, we showed that providers could rely partially or entirely on micro-operators, that is, private AV owners. This chapter offers an alternative strategy to guarantee high productivity, consisting of consolidating passenger and parcel flows. We focus on modeling a variation of the people and freight integrated transportation problem in which these commodities are pooled in mixed-purpose compartmentalized *shared autonomous vehicles* (SAVs). Such vehicles are supposed to combine freight and passenger overlapping journeys on the shared mobility infrastructure network. Once parcels are more amenable to waiting, new opportunities to optimize fleet usage arise.

This chapter is organized as follows. Section 6.1 further motivates the consolidation of different commodity flows and Section 6.2 formally describes the proposed *share-a-ride with parcel lockers problem* (SARPLP). Next, Section 6.3 offers a mathematical model for the problem and Section 6.4, presents an experimental study aiming to compare the performance of single-purpose and mixed-purpose fleets. Then, Section 6.5 presents the results of our exact implementation and Section 6.6 provides concluding remarks on the future of shared autonomous transportation of passengers and parcels. Parts of this chapter have been published in [8]:

B. A. Beirigo, F. Schulte, and R. R. Negenborn. Integrating people and freight transportation using shared autonomous vehicles with compartments. In *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems*, pages 392–397, Savona, Italy, 2018.

6.1 Introduction

Improving fleet productivity has been an overarching concern of this thesis. Such concern has led us to implement strategies to curb vehicle idleness, using both rebalancing and fleet size inflation algorithms. Nevertheless, since passenger transportation demand greatly varies throughout the day, any *shared autonomous vehicle* (SAV) fleet, hired or not, would be inevitably idle during low-demand hours. In this chapter, we propose an alternative strategy to make vehicle productivity less dependent on passenger demand fluctuation. This alternative consists of a *people and freight integrated* (PFIT) system, where transportation providers can use the same vehicles to fulfill both passenger and parcel requests interchangeably.

Due to the growth of the e-commerce proportion in distribution channels, urban goods' demand tends to become more fragmented, unveiling a necessity for new logistic approaches to deal with last-mile parcel deliveries [35]. For instance, Ford automaker envisages an "autolivery" future in which self-driving vans are used to quickly transport goods within a city but partnered with drones to realize the final leg of the journey [71]. Although some long-haul modes (e.g., aircrafts, ferries) already integrate passenger and freight demands, short-haul integration is hardly observed in practice [89]. To the best of our knowledge, integration on a ride-hailing setting was only explored in [55], [56], and [57]. The authors describe the *share-a-ride problem* (SARP), a variation of the well-known *dial-a-ride problem* (DARP), in which people and parcels can share the same taxi. However, ridesharing is limited in their approach once each passenger request can only share a ride with a single parcel request.

In contrast, we model the *share-a-ride with parcel lockers problem* (SARPLP), in which different commodities are transported simultaneously by compartmentalized mixed-purpose SAVs. We assume passenger compartments are private cabins tailored for human transportation, whereas freight compartments can be parcel lockers of different sizes. Differently from SARP, we consider all possible ridesharing scenarios of people and freight requests. Hence, each vehicle is allowed to carry one or more passengers, carry various sized parcels, and carry passengers and parcels. Finally, to assess the performance of such mixed-purpose fleets, we compare them with equivalent single-purpose fleets in which people and freight requests are handled separately.

6.2 Problem description

The SARPLP considers a fleet of mixed-purpose SAVs with compartments (i.e., shared vehicles equipped with people seats and parcel lockers). In the following, we identify potential compartment types and exemplify commodities they could accommodate:

- S – Documents, small objects (e.g., jewelry, electronics);
- M – Average sized objects (e.g., bags, purses);
- L – Large objects (e.g., suitcases, groceries);
- X – Extra-large objects (e.g., household appliances);
- A – Adult seat;
- C – Children seat (above 3 years of age);
- B – Baby seat (under 3 years of age);
- W – Wheel chair space.

The set of human compartments is $H = \{A, C, B, W\}$ and the set of freight compartments is $F = \{S, M, L, X\}$. We consider that passengers demand to be serviced on short notice, and parcels have more flexible pickup and delivery times (i.e., they do not have to be immediately addressed). For instance, an online store might determine a 24h delivery policy, whereas a restaurant might require a much shorter span. In this chapter, we model a static formulation, where the details of both request types (e.g., number of compartments, pickup/delivery coordinates, and time windows) are known in advance. Still, pickup windows and travel delays are assumed to be much shorter for passenger requests.

Regarding the fares of the transportation service, we consider that, besides time-dependent fares, commodities are charged according to the number and type of compartments requested. For freight transportation, for example, the cost can be proportional to the dimensions of the compartments. Ultimately, to properly determine a service fare, a request must include (i) the pickup and delivery coordinates and (ii) the number of units required for a determined type of compartment. This information is essential during the scheduling phase. Only vehicles whose available number of compartments match the order specifications are suited to service a potential commodity transportation demand.

Theoretically, the problem can be modeled as a variant of the classic *pickup and delivery problem* (PDP), in which transportation requests consist of point-to-point transports (i.e., movements of people or cargo between *origins and destinations* (ODs) [10, 104]). According to Berbeglia et al. [10], depending on the way vehicles move between points, such problems can be categorized as (i) many-to-many, (ii) one-to-many-to-one, and (iii) one-to-one. In (i), any point can serve as a source or destination for any commodity, whereas in (ii), commodities might be transported from the depot to the customers and from the customers to the depot. Finally, in (iii), each commodity has a given origin and a given destination, such as the door-to-door system presented in this chapter. Figure 6.1-a) highlights the differences from Li et al. [55] implementation, making explicit the concept of compartmentalized requests.

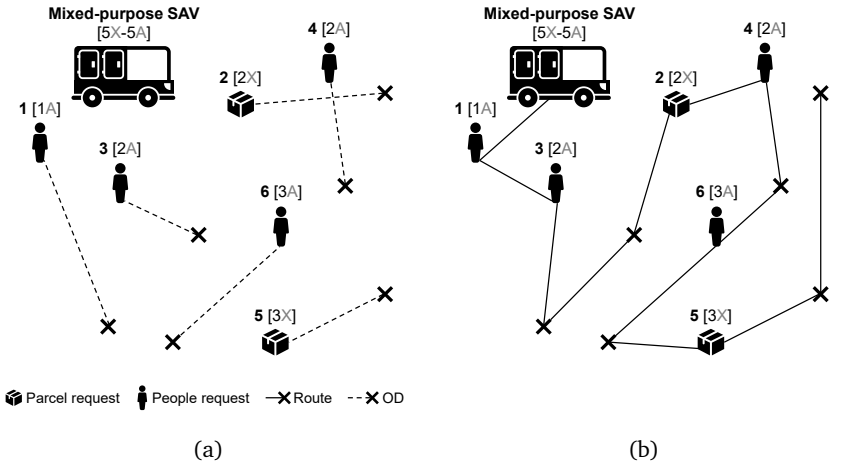


Figure 6.1: *Example of a PFIT system. (a) A mixed-purpose capacitated SAV must fulfill both people and parcel requests. (b) Both request types are integrated into the same vehicle throughout the route.*

A mixed-purpose SAV comprised of five compartments of type "A" and five compartments of type "X" (represented as 5X-5A) is supposed to find the best route to service a set of six transportation requests (two parcels and four people). Besides the identification number, each request also includes, between brackets, the number of compartments of each type demanded. Figure 6.1-b) shows the resulting route from the problem presented in Figure 6.1-a). The occupancy status of the vehicle after visiting each point (in terms of available compartments) is given by the sequence {1: 5X-4A, 3: 5X-1A, 1': 5X-4A, 3': 5X-5A, 2: 3X-5A, 4: 3X-3A, 4': 3X-5A, 6: 3X-2A, 6': 3X-5A, 5: 0X-5A, 5': 3X-5A, 2': 5X-5A}. Assuming request IDs also indicate their order of appearance, one should notice that the ridesharing route presented in Figure 6.1-b) privileges people demands, occasionally postponing parcel demands.

6.3 Problem definition

We extend the MILP model presented by Li et al. [55] to define a pickup and delivery problem that can handle constraints involving our performance demands, vehicles' specifications, and requests' heterogeneity. Table 6.1 presents all the problem variables. The SARPLP is modeled on a directed graph $G = (N, E)$. The node set N is partitioned into $\{P, D, O\}$, where $P = \{1, 2, \ldots, n\}$ is the set of pickup nodes and request indices, and $D = \{n+1, n+2, \ldots, 2n\}$ is the set of delivery nodes. The set O comprises the origin nodes o^k of vehicles $k \in K$. We consider that the time spent by a vehicle k to go from node i to node j, with $i, j \in N$, is $t_{i,j}$.

Each vehicle k is equipped with a set of compartment types $c \in C$, where C is the set of all compartment types across the whole fleet. Transporting a compartment c from i to j, generates a base fare p^c_{base} and a time-dependent fare p^c_{time}, while incurring in time-dependent operational cost c^c_{time}. For each compartment type $c \in C^k$, a vehicle k can have a capacity $Q^{c,k} \geq 0$. Since passenger and freight requests typically demand different service levels, we further partition C into $\{F, H\}$, where F and H correspond to freight and human compartment sets, respectively. Loading and unloading a compartment $c \in C$ is associated to delays d^c_{load} and d^c_{unload}.

Requests consist of transportation demands of commodities between origin destination pairs $(i, n+i)$, with $i \in P$ and $n+i \in D$. These commodities can be accommodated by compartment types $c \in C$, such that $q^c_i \geq 0$ represents the requested number of compartment types c. To guarantee an adequate flow of commodities, compartment demands are associated with all nodes in N. We let $q^c_i = -q^c_{i-n}$, $\forall i \in D$, and $q^c_{o^k} = 0 \; \forall k \in K, \forall c \in C$. A delay d_i is spent to load/unload cargo at node $i \in N$. If $i \in P$, $d_i = \sum_{c \in C} q^c_i \cdot d^c_{\text{load}}$ and if $i \in D$, then $d_i = \sum_{c \in C} q^c_i \cdot d^c_{\text{unload}}$.

All request nodes $i \in P \cup D$ have time windows defined by earliest arrival time e_i and latest departure time l_i. For $i \in P$, e_i is the time the request appears in the system whereas $l_i = e_i + \delta_i$, where δ_i is the maximum pickup delay. For $j \in D$, $e_j = e_{j-n} + d_{j-n} + t_{j-n,j}$ and $l_j = l_{j-n} + d_{j-n} + t_{j-n,j} + \Delta_j$, where Δ_j represents the maximum ride delay.

The decision variable $x_{i,j}^k$ is 1 when vehicle $k \in K$ traverses arc $(i,j) \in E$ and the variable $\omega_i^{c,k}$ is the load of compartment c from vehicle k upon leaving node $i \in N$. The ride delay of request $i \in P$ in vehicle k is defined by variable Δ_i^k and the time at which vehicle k arrives at node $i \in N$ is given by the variable τ_i^k.

Table 6.1: *Sets, parameters, and variables of the SARPLP.*

Sets	
K	Vehicles
H	Human compartments
F	Freight compartments
C	$= H \cup F$
P	Pickup nodes and request indices
D	Delivery nodes
O	Vehicle origin nodes
N	$= P \cup D \cup O$
Parameters	
Compartments	
p_{base}^c	Base fare of compartment c
p_{time}^c	Time-dependent fare of compartment c
d_{load}^c	Loading delay of compartment c
d_{unload}^c	Unloading delay of compartment c
Requests	
e_i	Earliest time at node i
l_i	Latest time at node i
δ_i	Maximum pick-up delay of request i
Δ_i	Maximum in-vehicle delay of request i
d_i	Cumulative load/unload delay at node i
q_i^c	Number of compartments c requested by i
Vehicles	
o^k	Origin node of vehicle k
$Q^{c,k}$	Number of compartments of type $c \in C^k$ of vehicle $k \in K$
c_{time}^k	Average operational cost/s (fuel, tolls, etc.) of vehicle k
Distances	
$t_{i,j}$	Travel time between nodes i and j, $\forall i,j \in N$
Variables	
$x_{i,j}^k$	(Binary) 1 if vehicle k traverses arc (i,j), 0 otherwise
τ_i^k	Arrival time of vehicle k at node i
Δ_i^k	Ride time of request i in vehicle k
$\omega_i^{c,k}$	Load of compartment $c \in C^k$ of vehicle k after visiting node $i \in N$

To streamline the solution space, we define the set of valid vehicle-node combinations

$$\mathcal{Q} = \{(k,i) \mid k \in K,\, i \in N,\, Q^{c,k} \geq |q_i^c|, \forall c \in C\},$$

where each tuple (k,i) indicates that a vehicle k can completely accommodate all compartment demands associated to node i. In turn, the set of valid movements, where each vehicle k can traverse arc (i,j), is given by

$$\mathcal{X} = \{(k,i,j) \mid k \in K,\, i,j \in N,\, (k,i) \in \mathcal{Q},\, (k,j) \in \mathcal{Q}\}.$$

The formulation of the SARPLP is as follows:

Maximize:

$$\sum_{k \in K} \sum_{i \in P} \sum_{c \in C} (p^c_{\text{base}} + p^c_{\text{time}} \cdot t_{i,n+i}) \cdot q_i^c \cdot x^k_{i,n+i} \quad - \sum_{(k,i,j) \in \mathcal{X}} c^k_{\text{time}} \cdot t_{i,j} \cdot x^k_{i,j} \tag{6.1}$$

Subject to:

$$\sum_{(k,i,j) \in \mathcal{X}} x^k_{i,j} \leq 1 \qquad \forall i \in P \tag{6.2}$$

$$\sum_{(k,i,j) \in \mathcal{X}} x^k_{i,j} = \sum_{(k,i,n+j) \in \mathcal{X}} x^k_{i,n+j} \qquad \forall k \in K,\, \forall j \in P \tag{6.3}$$

$$\sum_{(k,j,i) \in \mathcal{X}} x^k_{j,i} \leq \sum_{(k,i,j) \in \mathcal{X}} x^k_{i,j} \qquad \forall k \in K,\, \forall j \in D \tag{6.4}$$

$$\tau^k_j \geq (\tau^k_i + t_{i,j} + d_i) x^k_{i,j} \qquad \forall (k,i,j) \in \mathcal{X} \tag{6.5}$$

$$e_i \leq \tau^k_i \leq l_i \qquad \forall i \in N \tag{6.6}$$

$$\Delta^k_i = \tau^k_{n+i} - (\tau^k_i + d_i) \qquad \forall i \in P \tag{6.7}$$

$$t_{i,n+i} \leq \Delta^k_i \leq t_{i,n+i} + \Delta_i \qquad \forall i \in P \tag{6.8}$$

$$\omega^{c,k}_j \geq (\omega^{c,k}_i + q_j) x^k_{i,j} \qquad \forall (k,i,j) \in \mathcal{X},\, \forall c \in C \tag{6.9}$$

$$\omega^{c,k}_i \geq \max\{0,\, q^c_i\} \qquad \forall (k,i) \in \mathcal{Q}, \forall c \in C \tag{6.10}$$

$$\omega^{c,k}_i \leq \min\{Q^{c,k},\, Q^{c,k} + q^c_i\} \qquad \forall (k,i) \in \mathcal{Q}, \forall c \in C \tag{6.11}$$

$$x^k_{i,j} \in \{0,1\} \qquad \forall (k,i,j) \in \mathcal{X} \tag{6.12}$$

$$\omega^{c,k}_i \in \mathbb{N} \qquad \forall (k,i) \in \mathcal{Q}, \forall c \in C \tag{6.13}$$

$$\tau^k_i, \Delta^k_i \in \mathbb{N} \qquad \forall (k,i) \in \mathcal{Q} \tag{6.14}$$

The objective function (6.1) maximizes the total profit obtained from the commodity delivery revenue minus the operational cost of the active vehicles. Constraints (6.2) guarantee there is at most one arc leaving every pickup point (i.e., service denial is allowed in this formulation). Next, constraints (6.3) guarantee that vehicles visiting request pickup nodes must also visit their associated delivery nodes. Constraints (6.4) impose that every vehicle arriving at a delivery node can either stay or leave from it. The minimum arrival time of each vehicle k at a node j is imposed by constraints (6.5). This time corresponds to the sum of the arrival time of the previously visited node i, the cumulative loading time at i, and the travel time from i to j. In turn, constraints (6.6) enforce pickup time windows. Constraints (6.7) defines the ride time a commodity spends inside a vehicle, whereas constraints (6.8) ensure in-vehicle delay tolerances are respected. Next, Constraints (6.9), (6.10) and (6.11) ensure consistency of vehicle's compartment loading. Finally, model variables are defined in constraints (6.12), (6.13) and (6.14).

6.4 Experimental study

We conduct a numerical study considering various experimental settings and instances to determine the benefits of different fleet compositions. In particular, we focus on the performance assessment of fleets comprised of mixed-purpose vehicles, whose internal space is divided among freight and people compartments, and fleets composed of single-purpose vehicle compartments dedicated to a specific class of commodity.

When creating the instance scenarios, our ultimate goal was to provide insights on how distinct factors concerning particular characteristics of vehicles and requests may influence the model's outcome. We focus especially on (i) the number of vehicles used to address the requests, (ii) the overall profit gleaned during the fleet's operation, and (iii) the overall occupancy level. Regarding (iii), we assume the occupancy level of a single vehicle is proportional to the share of time and number of loaded compartments occupied throughout the entire operational route (i.e., from the dispatching moment until the delivery of the last customer). Hence, for a particular instance, the overall occupancy level consists of the average occupancy levels of all vehicles actually involved in the solution (i.e., dispatched from their origins).

SARPLP general operational settings

Table 6.2 presents the compartment-related parameters, which are shared by across all instances. We consider the transportation of "A" and "X" compartments, assuming they have similar dimensions and fares. Further, we consider that each vehicle can be equipped with ten of these compartments and service costs $c^k_{\text{time}} = 0.005\,€/\text{s}$, $\forall k \in K$. Regarding the pickup and in-vehicle delays, we consider that passengers demand to be picked up within three minutes and tolerate up to a ten-minute in-vehicle delay. In contrast, freight requests are more flexible, allowing a one-hour time window to be picked up and a five-hour in-vehicle delay. Additionally, we set the delays for embarking and disembarking passengers to one minute and loading and unloading parcels to five

Table 6.2: *Compartment parameters settings.*

Passengers		Parcels	
Parameter	Value	Parameter	Value
H	{A}	F	{X}
δ^{A}	3 min	δ^{X}	1h
Δ^{A}	10 min	Δ^{X}	5h
$d^{\mathrm{A}}_{\mathrm{load}}$	1 min	$d^{\mathrm{X}}_{\mathrm{load}}$	5 min
$d^{\mathrm{A}}_{\mathrm{unload}}$	1 min	$d^{\mathrm{X}}_{\mathrm{unload}}$	5 min
$p^{\mathrm{A}}_{\mathrm{base}}$	16 €	$p^{\mathrm{X}}_{\mathrm{base}}$	16 €
$p^{\mathrm{A}}_{\mathrm{time}}$	0.0016 €/s	$p^{\mathrm{X}}_{\mathrm{time}}$	0.0016 €/s

minutes. Two fleet compositions are considered. In single-purpose vehicle fleet compositions, half of the vehicles are equipped with "A" compartments and the other half with "X" compartments. In contrast, in mixed-purpose vehicle fleet compositions, every vehicle has five "A" and five "X" compartments.

Transportation scenario settings

We define a series of parameters to generate a number of scenarios where many fleet and request aspects are taken into consideration, namely:

- *Number of requests and fleet size* – We consider nine combinations of number of requests n and fleet sizes $|K|$, with $|K| \in \{4,8,16\}$ and $n \in \{8,16,32\}$.
- *Share of freight requests* – For each request set of size n, we check the influence of the proportion of freight requests on the model outcome. Ride-hailing demand typically varies throughout the day, and this may also be the case of freight demand. To further investigate the differences in handling these two commodity types, we consider freight requests may correspond to 25%, 50%, and 75% of the requests.
- *Interval between requests* – In real-world, large-scale transportation systems, many new requests may occur every second. In contrast, in smaller systems or less busy scenarios, intervals between requests might be longer. We consider integer intervals (in minutes) sampled from two ranges: $[0,0]$ (i.e., no interval between requests) and $[5,10]$.
- *Range of trip distances* – We investigate small-distance trips, ranging from 0.5km to 1km, and long-distance trips, ranging from 5km to 10km.
- *Compartment demand per request* – The number of compartments associated with a single request can be either low ($\leq$ 50% of available vehicle compartments) or high ($\geq$50% of available compartments).

For each fleet composition (single-purpose or mixed-purpose), a total of 216 scenarios are generated from the combination of the parameters investigated (summarized in

Table 6.3). We run each scenario considering three different spatial distributions of vehicles and requests, resulting in 1,296 instances (2 fleet compositions × 3 geographical distributions × 216 scenarios). Each distribution is randomly created based on the ODs present on the requests from a different time window of the taxi dataset from Manhattan, New York. We sample vehicle origins from the set of request origins and requests from the set of ODs whose distance required to cover the trip is within the scenario's range. Finally, we extract travel times between vehicle and request locations from the Mapbox Matrix API (www.mapbox.com) using the driving profile.

Table 6.3: *Summary of scenarios' parameters.*

Parameter	Values
Number of vehicles $\|K\|$	$\{4,8,16\}$
Number of requests n	$\{8,16,32\}$
Share of freight requests	$\{25\%,50\%,75\%\}$
Interval between requests	$\{[0,0],[5,10]\}$
Range of route distance	{0.5km-1km, 5km-10km}
Compartment demand/request	{low($\leq 50\%$), high($\geq 50\%$)}
Fleet composition	{single-purpose, mixed-purpose}

6.5 Results

Test instances were solved on an Intel Core i7, 2.30GHz CPU, 16GB RAM computer. Gurobi 7.0.2 Python interface was used to implement the SARPLP model, and the maximum runtime of each instance was set to ten minutes.

Table 6.4 presents the results achieved for both fleet compositions (i.e., mixed-purpose and single-purpose). We only consider the instances where the MIP gap is lower than 1%, such that we can draw more accurate conclusions on the fleet's performance. As a result, both fleet compositions are ultimately assessed over 149 scenarios (i.e., about 30% of the scenarios with non-optimal solutions were discarded). For each combination of fleet size $|K|$ and number of requests n, we present for each fleet composition the number of vehicles dispatched in the solution (#Veh.), the occupancy rate (Occ.) of these vehicles, and the total profit. In the last columns, we indicate the number of scenarios whose instances could be optimally solved to calculate the averages presented.

Mixed-purpose fleets have been found to reach superior results in 92% of the instances tested, having profits, on average, 12.3% higher than single-purpose fleets. However, the additional profits come at a cost. On average, 18% more mixed-purpose vehicles must be assigned, resulting in a 22.5% lower occupancy rate. Nevertheless, from the perspective of fleet operators, who want to make the most of their assets, idleness is acceptable only when it does not harm profitability. It is also worth mentioning that since we are dealing with a static setting, all people and freight demands are known in advance. This way, single-purpose vehicles can be timely dispatched to service each type of commodity request. However, when mixed-type demands occur dynamically, single-purpose vehicles may miss several opportunities to address overlapping human and freight routes.

Table 6.4: *Average results of SARPLP instances for each fleet composition.*

		Mixed-purpose			Single-purpose			N. of
$\|K\|$	n	Occ.(%)	#Veh.	Profit(€)	Occ.(%)	#Veh.	Profit(€)	scenarios
4	8	28.5	3.4	253.6	34.7	2.7	224.6	24
	16	27.8	3.7	465.4	39.7	3.2	375.0	17
	32	29.6	4.0	731.5	42.7	3.3	551.6	8
8	8	29.3	4.3	273.8	33.5	3.6	260.7	24
	16	27.6	6.1	542.7	33.3	5.4	504.7	17
	32	25.7	7.1	972.0	31.8	6.3	861.0	8
16	8	30.0	4.8	294.4	33.2	4.3	275.8	24
	16	29.4	8.4	590.7	33.7	7.4	566.5	18
	32	27.9	11.4	1,074.9	30.7	10.2	1,010.1	9

Table 6.5: *Profit breakdown of mixed-purpose and single-purpose fleets.*

		Mixed-purpose		Single-purpose		Profit
$\|K\|$	n	Revenue(€)	Cost(€)	Revenue(€)	Cost(€)	increase
4	8	290.7	37.0	256.92	32.35	13%
	16	528.9	63.5	423.62	48.63	24%
	32	825.0	93.5	620.72	69.11	33%
8	8	310.8	36.9	296.08	35.41	5%
	16	610.0	67.4	565.54	60.86	8%
	32	1087.7	115.7	971.22	110.20	13%
16	8	330.0	35.6	310.77	35.00	7%
	16	651.4	60.7	625.97	59.48	4%
	32	1,156.6	81.7	1,093.53	83.39	6%

This phenomenon commonly arises in busy scenarios, where the fleet capacity is insufficient to service requests adequately. In these scenarios, service will inevitably be denied to a larger share of the demands, with vehicles seeking to fulfill the most profitable set of requests while keeping low operational costs. Table 6.5 makes this relation more explicit, compiling the average revenue and costs for both fleet compositions, as well as the profit increase. The busier the logistical scenario, the higher is the superiority of the average profit of mixed-purpose fleets over single-purpose fleets. This relation can be particularly verified for the four-vehicle instances, where mixed-purpose fleets reach 13%, 24%, and 33% higher revenues when servicing 8, 16, and 32 requests, respectively. For this small fleet size scenario, it can also be verified that profits are greatly influenced by the increase in revenues, which far surpasses the increase in the operational costs. On the other hand, both fleet compositions exhibit similar performance for larger fleet sizes once virtually every request can be privately serviced by a vehicle.

6.6 Conclusions

This chapter answers research sub-question SQ5. We proposed a MILP formulation to deal with a variation of the people and freight integration transportation problem, in which mixed-purpose SAVs service both passenger and parcel requests. We assess

whether dividing internal vehicle spaces among these commodities is financially advantageous for a fleet operator. Our results have shown that fleets of mixed-purpose vehicles can accrue more profits than their single-purpose counterparts. In this novel setting, overlapping people and freight itineraries can be further combined to design more efficient routes. Nevertheless, although we have tested many different scenarios with varied characteristics, the results are still highly dependent on the general parameters assumed. For instance, the adoption of different compartment fares could create a bias towards a specific commodity. Alternatively, higher operational costs could drive to solutions where farther requests are not worth attending. Hence, future research shall be focused on striking a balance between these factors to provide a sensible range of parameter options for fleet operators.

Chapter 7

Conclusions and future research

This thesis addresses the challenges entailed by the dynamic fleet management of shared autonomous vehicles in a range of logistical scenarios. The underlying motif pervading all chapters is a concern with service quality, especially in terms of responsiveness, reliability, privacy, and accessibility. Service quality is crucial for user acceptance and, ultimately, widespread adoption, a prerequisite to enabling the benefits entailed by a sharing paradigm. To this end, we have proposed dynamic and stochastic methods that cater to heterogeneous user preferences while improving fleet utilization in a variety of transportation settings. In this chapter, we summarize our conclusions (Section 7.1) and contributions (Section 7.2), and provide an outlook on future research (Section 7.3).

7.1 Conclusions

This thesis aimed to answer the main research question:

RQ: *How can AMoD systems leverage supply and demand information as well as cities' infrastructure to balance the goals of all mobility stakeholders?*

In Chapter 1, we defined five research sub-questions, which were subsequently addressed through Chapters 2-6. In Section 7.1.1, we summarize the answer to these questions, and in Section 7.1.2, we present how the main research question is answered across chapters from the perspective of each stakeholder.

7.1.1 Answers to sub-research questions

SQ1: *How can fleet operators deal with the operational restrictions arising in the early stages of AV deployment?*

In Chapter 2, we developed a mixed-integer programming model for AMoD providers operating on hybrid transportation networks, comprised of mixed autonomous and non-autonomous vehicle zones. Our model allows providers to choose the optimal fleet size and mix to cover the entire demand under a range of AVZ deployment factors, such

as automation cost depreciation, intra- and inter-zone demand patterns, and automated driving network coverage. In particular, we analyze the adoption of dual-mode vehicles, consisting of vehicles driven manually (outside AVZs) and autonomously (inside AVZs), therefore capable of servicing any request. We show that the request demand patterns are the most critical parameter driving up the number of dual-mode vehicles. Once the cost of operating dual-mode vehicles is the highest (the provider has to pay a driver over the AV technology), our findings stress the necessity of deploying AVZs that reflect the user demand patterns to avoid zone crossing. Hence, in the early stages of AV technology deployment, providers may find it beneficial to work in tandem with cities to deploy AVZs that can ultimately alleviate congestion, improve accessibility, and minimize operational costs.

SQ2: *How can AMoD systems guarantee service quality, in terms of responsiveness, reliability, and privacy, while improving fleet productivity?*

In Chapter 3, we introduced a new class of fleet management problems where AMoD providers are no longer limited to static fleet sizes, increasing and decreasing the number of vehicles in the short term to meet diversified user expectations. In turn, these expectations are the basis for establishing service quality contracts whereby heterogeneous users can choose ride experiences that best match their preferences (e.g., waiting times and willingness to share). Since fully honoring these contracts may sometimes be onerous, our proposed method allows providers to steer the vehicle supply to first service the highest priority or most profitable customer segment. This way, providers are better equipped to make a compromise between avoiding service level violations and hiring extra vehicles, according to the particular conditions of an operational environment (e.g., availability of idle vehicles, hiring costs, user dissatisfaction costs). Regardless of the compromise chosen, however, our method is shown to guarantee that hiring occurs only to the extent that it preserves minimum service level requirements. Using a lexicographic method, we solved a multi-objective function where the primary goal is to minimize rejections (i.e., the most significant source of inconvenience) to subsequently minimize fleet size and then service-level violations. We have formalized the problem using a MILP formulation and proposed a matheuristic to assign vehicles to new requests optimally. Based on an experimental study using New York City taxi data, we have found that the developed approach allows us to significantly improve the service quality of all considered user categories. By enforcing the proposed service-level constraints, we can meet the expectations of 85.7% of the users across classes, a 53% average increase in comparison to conventional ridesharing systems. Additionally, when hiring is enabled, we can meet the expectations of 95.6% of the user requests, at the expense of a mild fleet inflation (a maximum surplus of 168 FAVs is observed at the evening demand peak).

SQ3: *How can AMoD systems explore the stochastic information surrounding privately-owned vehicle supply and heterogeneous demand?*

In Chapter 4, we propose an approximate dynamic programming algorithm to the stochastic information related to both privately-owned vehicle supply and heterogeneous demand composition. Our approach rebalances and hires vehicles in anticipation of transportation demand, positioning the fleet at promising locations according to the

user distribution. We show that the policy learned by sampling the demand from a particular weekday is generic enough to adequately address the demand patterns of all similar weekdays throughout a whole year. Without any hiring, such a policy could consistently outperform a reactive optimization policy, servicing on average about 18% more requests. Although both policies manage to service most requests when hiring is considered, the proposed policy has been shown to do it more efficiently, using fewer FAVs, and providing better service levels. Additionally, we provide a detailed experimental study on the performance of a range of fleet repositioning strategies that combine both long- and short-range rebalancing targets, also consisting of regional centers. To deal with multiperiod resource-transformation times, we propose a discount mechanism to dampen post-decision states' value functions, allowing vehicles to consider varying rebalance ranges. This mechanism has been shown to perform well and can be an option to typical formulations that add an extra attribute to the resource attribute vector to account for arrival times at destination locations. Also, since we consider a realistic transportation network, our method allows city managers to exert greater traffic control by limiting the maximum number of vehicles inbound to each intersection. Experimentally, we show that doing so can improve value function approximations, leading to better quality solutions. Moreover, we propose a novel method to define spatial aggregation levels based on the facility location problem. Our method can harness high-granularity street networks' properties to summarize locations in increasingly broader regional centers. From these centers, vehicles can access a set of neighboring locations within a set amount of time that ultimately can be tuned to reflect user service levels.

SQ4: *How can cities steer providers towards achieving equity goals?*

In Chapter 5, we show that cities can make operators' goals converge to improve equitable access to transportation. Since transport accessibility is a precondition to achieving other life opportunities, our approach ultimately allows cities to improve social equity. We propose an approximate dynamic programming algorithm to rebalance vehicles to targeted city regions where accessibility is assumed to be lacking. We investigate the extent to which incentives to reach these regions can alter the natural rebalancing biases arising from the transportation data. This way, cities could, for example, increase service levels of disadvantaged regions without having to manage or owning transportation assets.

SQ5: *How can AMoD systems handle passenger and cargo demands interchangeably to improve fleet productivity?*

In Chapter 6, we propose a MILP formulation to model the operation of versatile mixed-purpose SAVs able to deal with both passenger requests and parcel loads interchangeably. Our experiments suggest that consolidating different commodity flows in the same vehicles can lead to further profits, compared to handling them separately, using traditional single-purpose vehicles. The advantages of using mixed-purpose fleets are even more pronounced in busy scenarios, where additional people and freight overlapping itineraries are likely to be matched, therefore increasing providers' productivity.

7.1.2 Answer to research question

In Chapter 1, we identified four stakeholders, namely, users, AMoD providers, independent AV owners, and cities. In the following, we answer the research question from the perspective of each stakeholder, describing how their needs were addressed across chapters.

Users

Our first priority has been set to fulfilling the needs of heterogeneous users in terms of responsiveness, reliability, privacy, and accessibility. First, Chapter 2 covers the transport accessibility aspect from a mobility perspective, allowing providers to physically reach request ODs in the early stages of vehicle automation where autonomous and non-autonomous zones may co-exist. Chapters 3 and 4 are dedicated to fulfilling user service quality expectations through the implementation of strict service quality contracts. These contracts allow users to further personalize their ride experience by letting them to choose precisely the expected minimum service quality requirements and be compensated accordingly when these are not fulfilled. Further, Chapter 5 proposes strategies to improving equitable access to transportation, enabling users from different areas to enjoy the benefits of AMoD systems. Next, in Chapter 6, we consider the requirements of freight transportation requests. Since parcels are more amenable to waiting, we show that platforms can improve fleet utilization by servicing them interchangeably with people requests.

AMoD providers

Providers' goals are inextricably linked to users' goals once revenue maximization depends on building customer loyalty, which, in an AMoD setting, is directly related to meeting users' service quality expectations. Hence, for providers, the goal has been set to meeting these expectations while designing cost-efficient routes, improving fleet utilization, and minimizing hires. We show for different AV development stages how providers can employ heterogeneous vehicle fleets to fulfill a diverse set of user requirements. Starting from Chapter 2, we show how providers can thrive in the transition phase of vehicle automation, determining the fleet composition necessary to operate in various AV deployment scenarios. Next, in Chapter 3, we demonstrate how adopting SQCs can add nuance to the standard user performance indicators. On the one hand, SQCs let users bargain minimum service quality requirements with providers and be compensated when violations occur. On the other hand, providers can enjoy greater leeway in route creation by exploiting personalized user requirements (e.g., tolerance delays, privacy preferences). Chapter 4 proposes a formulation where providers go the extra mile to satisfy user needs by compensating users for contract breaches. We show these compensations, which are penalties from the provider's perspective, can improve fleet utilization, increasing the effectiveness of our anticipatory rebalancing algorithm. Lastly, Chapter 6 shows how providers can make use of fleet downtime even further by addressing two commodity types, leveraging their different service quality requirements to enable extra ridesharing options.

Independent AV owners

From the perspective of independent owners, their vehicles are supposed to accrue the highest possible revenue throughout the time they are available to work. In this thesis, we assumed providers do not impose a minimum revenue threshold, that is, idle vehicles are always better off by picking up requests. In Chapter 3, we focused solely on examining the impact of fleet elasticity. We assumed that privately-owned AVs were scattered throughout the city and readily available in sufficient numbers to spontaneously join the working fleet. In Chapter 4, we contemplated more realistic AV owner preferences. We considered a fixed supply of freelance AVs was distributed throughout the city following a stochastic process. Our approached accounts for a series of owner preferences, such as the announcement time at the AMoD system, the availability period, and the origin location.

Cities

From the perspective of city managers and traffic authorities, our work proposes several ways of using city infrastructures smartly. We consider that these stakeholders aim to reduce urban congestion and improving accessibility. In Chapter 2, we specifically evaluate the effect of deploying AV-ready zones on fleet management strategies. Chapter 3 addresses the major causes of increased congestion due to AMoD systems' operation, namely, the excessive number of vehicles and inefficient fleet management. We explicitly aim to minimize fleet size, ensuring providers can pick up users adequately without having to own large fleets. In Chapter5, we integrate street network infrastructural limitations to our rebalancing algorithms, making sure vehicles are prevented from flooding intersections. Our method is general enough to deal with time-dependent parking costs and congestion pricing, such that the proposed learning algorithm can be trained to consider city constraints. Lastly, in Chapter 6, we present a model for integrating people and freight flows, therefore increasing transportation efficiency due to cargo consolidation and reducing the need for single-purpose vehicles.

7.2 Contributions of the thesis

This thesis addresses the challenges entailed by deploying AMoD systems in a range of logistical scenarios. In summary, our contributions are the following:

- In Chapter 2, we propose a mathematical model for managing autonomous and non-autonomous vehicles in a heterogeneous network where the infrastructure is partially ready to accommodate automated operations. Parts of this chapter have been published in:

  B. A. Beirigo, F. Schulte, and R. R. Negenborn. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1325–1330, Maui, HI, United States, 2018.

- In Chapter 3, we consider a fully autonomous setting, where an AMoD provider aims to meet user expectations fully. To overcome the limitations related to fixed-

size models, we consider fleet size elasticity, a feature whereby providers can use idle privately-owned AVs (hired on short notice) to sustain service quality. To minimize hirings while improving user experience, we propose a multi-objective matheuristic that uses a lexicographical method to hierarchically address a series of objectives. Parts of this chapter have been submitted to:

B. A. Beirigo, F. Schulte, and R. R. Negenborn. A business class for autonomous mobility-on-demand: Modeling service quality contracts in ridesharing systems. *Submitted to a journal*.

- In Chapter 4, we build upon Chapter 3 by considering the stochastic nature inherent to the distribution of requests and the availability of privately-owned vehicles. We harness this stochasticity to propose an anticipatory optimization approach using an approximate dynamic programming framework to manage a heterogeneous fleet of company- and privately-owned AVs. Parts of this chapter have been submitted to:

  B. A. Beirigo, F. Schulte, and R. R. Negenborn. A learning-based optimization approach for autonomous ridesharing platforms with service level contracts and on-demand hiring of idle vehicles. *Transportation Science* (in press, 2021).

- In Chapter 5, we investigate service quality from a city-wide perspective, where the ultimate goal is to guarantee even access to mobility across different regions of the city. We propose an approximate dynamic programming algorithm to anticipatorily distribute vehicles to targeted areas, where they service the first-mile requests of disadvantaged populations. Parts of this chapter have been published in:

  B. A. Beirigo, F. Schulte, R. R. Negenborn. Overcoming mobility poverty with shared autonomous vehicles: A learning-based optimization approach for Rotterdam Zuid. In *Proceedings of the 11th International Conference on Computational Logistics*, pages 492-506, Enschede, the Netherlands, 2020.

- In Chapter 6, we propose a mathematical model for a people and freight integrated transportation system, where passenger and parcel demands are handled interchangeably by a fleet of compartmentalized AVs. Parts of this chapter have been published in:

  B. A. Beirigo, F. Schulte, and R. R. Negenborn. Integrating people and freight transportation using shared autonomous vehicles with compartments. In *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems*, pages 392–397, Savona, Italy, 2018.

7.3 Future research

In this section, we describe future research directions related to enabling more robust and versatile AMoD systems.

7.3.1 The edge of crowdsourced AMoD systems

In Chapters 3 and 4, we investigate the benefits of fleet size elasticity, a capability already used at large by current TNCs but still overlooked in the AV fleet management research. We considered idle freelance privately-owned AVs could occasionally join a company-owned AV fleet to sustain user service quality expectations, especially at high-demand times. As pointed out by Campbell [14], there are enough incentives for future AV owners to simultaneously own and share their vehicles on ridesharing platforms, such that a future where vehicle supply is crowdsourced from multiple sources, rather than a single fleet owner, is likely to happen. In such a sharing-dominant future, an AMoD provider relying on company- and private-owned AV fleets could aim to find the minimum fleet size able to complement the available supply at crowdsourced vehicles, which unfolds dynamically and stochastically throughout the day. Hence, in contrast with traditional fleet sizing methods that use the transportation demand to determine the fleet size, in this scenario, demand and third-party-owned AV supply would be used to determine an adequate minimum fleet size. Further, by considering that privately-owned vehicles have heterogeneous preferences regarding minimum workload or profits, formulations could also weigh vehicle acquisition costs in the timeline of AV development and expected increase of the AV fleet.

7.3.2 Distributing accessibility based on fine-grained indicators

In Chapter 5, we proposed an approach to overcome mobility poverty by improving transport accessibility for disadvantaged users departing from low-income areas. However, by adopting more refined accessibility indicators (e.g., [111]), cities could instrumentalize AMoD systems to implement more robust plans to distribute accessibility over different regions and demographics. Through detailed demand (e.g., itineraries, social-economic profile) and supply (e.g., mode availability, schedule) information, learning-based fleet management approaches could be tuned to fix both structural or time-dependent accessibility shortcomings. As technology develops and AV rides become more affordable, subsidizing AMoD transportation to evenly improve city residents' mobility may eventually become the best measure.

7.3.3 Learning-based parking and curbside management strategies

Since AVs do not have to necessarily park in the vicinity of the traveler [101], scheduling and routing strategies of mobility services may be adjusted to make use of this feature. Owners can make substantial savings if AVs opt for parking in central/urban areas or suburban locations over central business districts [59], a trajectory change in future parking demand validated in a simulation study by Zhang and Wang [122]. However, cruising for parking and increased curbside usage may also incur underlying costs for

AV owners and cities. The former must weigh out parking-related costs, additional fuel expenditure, and longer delays, whereas the latter may suffer from increased traffic due to empty vehicle driving and curbside parking-induced congestion. Hence, AVs may have to account for a range of spatial- and time-dependent factors such as advanced congestion pricing, empty vehicle cruising fees, curbside loading/unloading fees, traffic delays, and parking availability. Modeling the details and stochasticity entailed by this rich operational scenario can be a significant step in implementing realistic AV fleet management strategies.

7.3.4 Scheduling maintenance and cleaning

The acceptance rate of AVs is known to be highly dependent on consumer's safety perception. When compared with regular vehicles, AVs rely on a greater number of hardware and software components, and the slightest inaccuracy on them might result in fatal events. Hence, a fleet owner must schedule regular maintenance appointments to guarantee all elements empowering autonomous technology work effectively while operating. Moreover, unless self-cleaning methods are developed and/or strict usage norms are established, vehicles would also require frequent cleaning. Additionally, due to rising concerns regarding the spread of infectious diseases, the demand for sanitation measures could further increase. The experience with taxi and public transportation has shown that some passengers litter, smoke, spill food and drinks, spit, bring pets, and in some cases vandalize vehicles [59].

7.3.5 Modular autonomous vehicle (MAV) ecosystem

Modular autonomous vehicles (MAVs) consist of a conceptual vehicle design that allows the separation of upper and lower parts. This design allows for flexible operation and efficient asset utilization, once a generic lower part, or platform, can handle both passengers and parcels throughout the day, provided that the upper part, or body, is swapped accordingly and timely. Such setup creates an entirely new ecosystem comprised of diverse body types, platforms, and swapping facilities, where parts are switched and stored.

This ecosystem builds upon the solution presented in Chapter 6, where a transportation provider aims to minimize the operational costs of servicing both passenger and parcel requests. This setting can be expanded in many interesting directions. For example, one could consider the optimal facility deployment and size as well as heterogeneous swapping delays (i.e., the time required to outfit a platform with a new body) at each facility. Alternatively, a scenario featuring body and platform independence could also be investigated. In this scenario, one could assume that autonomy capabilities are embedded in the lower part only, with upper parts coming from different suppliers, giving birth to a new crowdsourced market. For instance, due to hygiene concerns (in the shadow of pandemics) or for convenience's sake, some passengers could own personal pods and demand from operators independent lower parts to pick them up.

7.3.6 Demand-tailored fleet composition

As an immediate consequence of the AV fleet elasticity model, providers have the chance to configure fleets to precisely reflect the demand preferences. For instance, in Chapter 3, we showed that most hired third-party vehicles joined the platform to service users belonging to the high-priority service quality class, consisting of the shortest delays and private rides. In a scenario of disseminated ownership, it is likely that vehicles, or vehicle bodies, differ markedly in their built and available features. Hence, besides leveraging third-party vehicles' stochastic distribution to decide where and when to hire (as done in Chapter 4), providers can also decide which vehicle types best fulfill the demand needs at a given time. This decision is crucial when independent owners can stipulate their leasing preferences and cities impose further AV operation restrictions (e.g., empty-driven distances, parking rates). Moreover, considering the MAV ecosystem described in Section 7.3.5, tailoring fleet composition according to historical demand patterns becomes even more flexible. Stochastic information about the availability of bodies and platforms, as well as the working capacity of facility centers, could be exploited to anticipatorily outfit vehicles, creating a time-dependent fleet configuration that reflects future demand states.

Bibliography

[1] L. Al-Kanj, J. Nascimento, and W. B. Powell. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research*, pages 1–40, 2020.

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

[3] J. Alonso-Mora, A. Wallar, and D. Rus. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3583–3590, Vancouver, Canada, 2017.

[4] C. Archetti and M. G. Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.

[5] C. Archetti, M. Savelsbergh, and M. G. Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016.

[6] A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk. Crowdsourced delivery – A dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.

[7] B. A. Beirigo, F. Schulte, and R. R. Negenborn. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1325–1330, Maui, HI, United States, 2018.

[8] B. A. Beirigo, F. Schulte, and R. R. Negenborn. Integrating people and freight transportation using shared autonomous vehicles with compartments. In *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems*, pages 392–397, Savona, Italy, 2018.

[9] B. A. Beirigo, F. Schulte, and R. R. Negenborn. Overcoming mobility poverty with shared autonomous vehicles: A learning-based optimization approach for Rotterdam Zuid. In *Proceedings of the 11th International Conference on Computational Logistics (ICCL2020)*, Enschede, the Netherlands, 2020.

[10] G. Berbeglia, J. F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

[11] H. Billhardt, A. Fernández, L. Lemus, M. Lujak, N. Osman, S. Ossowski, and C. Sierra. Dynamic coordination in fleet management systems: Toward smart cyber fleets. *IEEE Intelligent Systems*, 29(3):70–76, 2014.

[12] P. M. Boesch, F. Ciari, and K. W. Axhausen. Autonomous vehicle fleet sizes required to serve different levels of demand. *Transportation Research Record: Journal of the Transportation Research Board*, 2542:111–119, 2016.

[13] T. Bulhões, M. H. Hà, R. Martinelli, and T. Vidal. The vehicle routing problem with service level constraints. *European Journal of Operational Research*, 265(2): 544–558, 2018.

[14] H. Campbell. Who will own and have propriety over our automated future? Considering governance of ownership to maximize access, efficiency, and equity in cities. *Transportation Research Record*, 2672(7):14–23, 2018.

[15] J. Castiglione, D. Cooper, B. Sana, D. Tischler, T. Chang, G. Erhardt, S. Roy, M. Chen, and A. Mucci. TNCs & Congestion. Technical report, San Francisco County Transportation Authority, San Francisco, CA, United States, Oct. 2018.

[16] Centraal Bureau voor de Statistiek (CBS). Wijk- en buurtkaart, 2019.

[17] T. D. Chen and K. M. Kockelman. Management of a shared autonomous electric vehicle fleet: Implications of pricing schemes. *Transportation Research Record*, 2572(1):37–46, 2016.

[18] T. D. Chen, K. M. Kockelman, and J. P. Hanna. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243–254, 2016.

[19] Z. Chen, F. He, Y. Yin, and Y. Du. Optimal design of autonomous vehicle zones in transportation networks. *Transportation Research Part B: Methodological*, 99: 44–61, 2017.

[20] S. Cohen and C. Cabansagan. A framework for equity in new mobility. Technical report, Transform, June 2017.

[21] S. Cohen and S. Sahar. Can we advance social equity with shared, autonomous and electric vehicles? Technical report, Institute of Transportation Studies at the University of California, Davis, Feb. 2017.

[22] J. Cohn, R. Ezike, J. Martin, K. Donkor, M. Ridgway, and M. Balding. Examining the equity impacts of autonomous vehicles: A travel demand model approach. *Transportation Research Record*, 2673(5):23–35, 2019.

[23] J.-F. Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3):573–586, 2006.

[24] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo. *Handbooks in Operations Research and Management Science*, volume 14, chapter Vehicle routing, pages 367–428. Elsevier, 2007.

[25] H. Creger, J. Espino, and A. S. Sanchez. Autonomous vehicle heaven or hell? Creating a transportation revolution that benefits all. Technical report, The Greenlining Institute, Jan. 2019.

[26] J. R. Daduna. Developments in city logistics - The path between expectations and reality. In *Proceedings of the 10th International Conference on Computational Logistics (ICCL2019)*, volume 11756 LNCS, pages 3–21, Barranquilla, Colombia, 2019.

[27] L. Dahle, H. Andersson, and M. Christiansen. The vehicle routing problem with dynamic occasional drivers. In *Proceedings of the 8th International Conference on Computational Logistics*, pages 49–63, Southampton, United Kingdom, 2017.

[28] L. Dahle, H. Andersson, M. Christiansen, and M. G. Speranza. The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122–133, 2019.

[29] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[30] J. Dean, A. J. Wray, L. Braun, J. M. Casello, L. McCallum, and S. Gower. Holding the keys to health? A scoping study of the population health impacts of automated vehicles. *BMC Public Health*, 19(1):1–10, 2019.

[31] P. T. Do, N. V. D. Nghiem, N. Q. Nguyen, and Q. D. Pham. A time-dependent model with speed windows for share-a-ride problems: A case study for Tokyo transportation. *Data and Knowledge Engineering*, 114:67–85, 2018.

[32] D. J. Fagnant and K. M. Kockelman. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies*, 40:1–13, 2014.

[33] D. J. Fagnant and K. M. Kockelman. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[34] D. J. Fagnant and K. M. Kockelman. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation*, 45(1): 143–158, 2018.

[35] E. Fatnassi, J. Chaouachi, and W. Klibi. Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics. *Transportation Research Part B: Methodological*, 81:440–460, 2015.

[36] D. Fiedler, M. Čertický, J. Alonso-Mora, and M. Čáp. The impact of ridesharing in mobility-on-demand systems: Simulation case study in Prague. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1173–1178, Maui, HI, United States, 2018.

[37] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013.

[38] M. Gendreau, O. Jabali, and W. Rei. 50th anniversary invited article – Future research directions in stochastic vehicle routing. *Transportation Science*, 50(4): 1163–1173, 2016.

[39] A. George, W. B. Powell, and S. R. Kulkarni. Value function approximation using multiple aggregation for multiattribute resource management. *Journal of Machine Learning Research*, 9:2079–2111, 2008.

[40] A. P. George and W. B. Powell. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1): 167–198, 2006.

[41] M. Gueriau and I. Dusparic. SAMoD: Shared autonomous mobility-on-demand using decentralized reinforcement learning. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1558–1563, Maui, HI, United States, 2018.

[42] K. M. Gurumurthy and K. M. Kockelman. Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from Orlando, Florida. *Computers, Environment and Urban Systems*, 71:177–185, 2018.

[43] S. Hassler. 2017: The year of self-driving cars and trucks, 2016. URL http://spectrum.ieee.org/transportation/advanced-cars/2017-the-year-of-selfdriving-cars-and-trucks. Accessed: 2017-03-27.

[44] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. *Building Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., 1983.

[45] S. C. Ho, W. Y. Szeto, Y. H. Kuo, J. M. Leung, M. Petering, and T. W. Tou. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111:1–27, 2018.

[46] M. Hyland and H. S. Mahmassani. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign avs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, 92:278–297, 2018.

[47] M. F. Hyland and H. S. Mahmassani. Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. *Transportation Research Record: Journal of the Transportation Research Board*, 2653(1):26–34, 2017.

[48] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone. Data-driven model predictive control of autonomous mobility-on-demand systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, Brisbane, QLD, Australia, 2018.

[49] R. M. Jorgensen, J. Larsen, and K. B. Bergvinsdottir. Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society*, 58(10): 1321–1331, 2007.

[50] R. Krueger, T. H. Rashidi, and J. M. Rose. Preferences for shared autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 69:343–355, 2016.

[51] R. Lahyani, M. Khemakhem, and F. Semet. Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1): 1–14, 2015.

[52] T. V. Le, A. Stathopoulos, T. Van Woensel, and S. V. Ukkusuri. Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, 103(3):83–103, 2019.

[53] A. Lee and M. Savelsbergh. Dynamic ridesharing: Is there a role for dedicated drivers? *Transportation Research Part B: Methodological*, 81:483–497, 2015.

[54] M. W. Levin and S. D. Boyles. Effects of autonomous vehicle ownership on trip, mode, and route choice. *Transportation Research Record*, 2493(1):29–38, 2015.

[55] B. Li, D. Krushinsky, H. A. Reijers, and T. Van Woensel. The share-a-ride problem: People and parcels sharing taxis. *European Journal of Operational Research*, 238 (1):31–40, 2014.

[56] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers. An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers and Operations Research*, 66:170–180, 2016.

[57] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers. The share-a-ride problem with stochastic travel times and stochastic delivery locations. *Transportation Research Part C: Emerging Technologies*, 67:95–108, 2016.

[58] K. Lin, R. Zhao, Z. Xu, and J. Zhou. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1774–1783, London, United Kingdom, 2018.

[59] T. Litman. Autonomous vehicle implementation predictions: Implications for transport planning. Technical report, Victoria Transport Policy Institute, 2017.

[60] M. Lokhandwala and H. Cai. Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC. *Transportation Research Part C: Emerging Technologies*, 97:45–60, 2018.

[61] K. Lucas, G. Mattioli, E. Verlinghieri, and A. Guzman. Transport poverty and its adverse social consequences. *Proceedings of the Institution of Civil Engineers - Transport*, 169(6):353–365, 2016.

[62] K. Lucas, B. Van Wee, and K. Maat. A method to evaluate equitable accessibility: Combining ethical theories and accessibility-based approaches. *Transportation*, 43(3):473–490, 2016.

[63] S. Ma, Y. Zheng, and O. Wolfson. Real-time city-scale taxi ridesharing. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1782–1795, 2015.

[64] B. Madadi, R. van Nes, M. Snelder, and B. van Arem. Assessing the travel impacts of subnetworks for automated driving: An exploratory study. *Case Studies on Transport Policy*, 7(1):48–56, 2019.

[65] McKinsey & Company and Bloomberg New Energy Finance. An integrated perspective on the future of mobility. Technical report, McKinsey & Company, Oct. 2016.

[66] J. Meyer, H. Becker, P. M. Bösch, and K. W. Axhausen. Autonomous vehicles: The next jump in accessibilities? *Research in Transportation Economics*, 62:80–91, 2017.

[67] Y. Molenbruch, K. Braekers, and A. Caris. Operational effects of service level variations for the dial-a-ride problem. *Central European Journal of Operations Research*, 25(1):71–90, 2017.

[68] Y. Molenbruch, K. Braekers, and A. Caris. Benefits of horizontal cooperation in dial-a-ride services. *Transportation Research Part E: Logistics and Transportation Review*, 107:97–119, 2017.

[69] A. Mourad, J. Puchinger, and C. Chu. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, 123: 323–346, 2019.

[70] B. Nag, B. L. Golden, and A. Assad. Vehicle routing with site dependencies. *Vehicle Routing: Methods and Studies*, pages 149–159, 1988.

[71] B. Nagy. Ford employees' self-driving "Autolivery" concept demonstrate ideas for more sustainable "City of Tomorrow", 2017. URL https://media.ford.com/content/fordmedia/feu/en/news/2017/02/27/ford-employees_-self-driving-autolivery-concept-demonstrate-idea.html. Accessed: 2020-11-02.

[72] S. Narayanan, E. Chaniotakis, and C. Antoniou. Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, 111:255–293, 2020.

[73] T. Nguyen. ETA phone home: How uber engineers an efficient route, 2015. URL https://eng.uber.com/engineering-an-efficient-route/. Accessed: 2020-11-02.

[74] M. Nourinejad, S. Bahrami, and M. J. Roorda. Designing parking facilities for autonomous vehicles. *Transportation Research Part B: Methodological*, 109:110–127, 2018.

[75] J. Oyola, H. Arntzen, and D. L. Woodruff. The stochastic vehicle routing problem, a literature review, part i: Models. *EURO Journal on Transportation and Logistics*, 7(3):193–221, 2018.

[76] K. Pangbourne, M. N. Mladenović, D. Stead, and D. Milakis. Questioning mobility as a service: Unanticipated implications for society and governance. *Transportation Research Part A: Policy and Practice*, 131:35–49, 2020.

[77] J. Paquette, J. F. Cordeau, and G. Laporte. Quality of service in dial-a-ride operations. *Computers and Industrial Engineering*, 56(4):1721–1734, 2009.

[78] S. N. Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5):912–930, 2011.

[79] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31 (7):839–854, 2012.

[80] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[81] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality: Second Edition*. John Wiley and Sons, 2011.

[82] W. B. Powell, H. P. Simão, and B. Bouzaiene-Ayari. Approximate dynamic programming in transportation and logistics: A unified framework. *EURO Journal on Transportation and Logistics*, 1(3):237–284, 2012.

[83] H. N. Psaraftis, M. Wen, and C. A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

[84] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *Autonomous Robots*, 42:1427–1442, 2018.

[85] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (J3016_201806). Technical report, 2018. URL https://www.sae.org/standards/content/j3016_201806/. Accessed: 2020-11-02.

[86] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, 111(37):13290–13294, 2014.

[87] D. O. Santos and E. C. Xavier. Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications*, 42(19):6728–6737, 2015.

[88] M. Savelsbergh and M. Sol. Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4):474–490, 1998.

[89] M. Savelsbergh and T. V. Woensel. City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

[90] Y. O. Scherr, B. A. Neumann-Saavedra, M. Hewitt, and D. C. Mattfeld. Service network design for same day delivery with mixed autonomous fleets. *Transportation Research Procedia*, 30:23–32, 2018.

[91] Y. O. Scherr, B. A. Neumann Saavedra, M. Hewitt, and D. C. Mattfeld. Service network design with mixed autonomous fleets. *Transportation Research Part E: Logistics and Transportation Review*, 124:40–55, 2019.

[92] Y. O. Scherr, M. Hewitt, B. A. Neumann Saavedra, and D. C. Mattfeld. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195, 2020.

[93] D. Shoup. *The high cost of free parking: Updated edition*. Routledge, 2017.

[94] H. P. Simão, J. Day, A. P. George, T. Gifford, J. Nienow, and W. B. Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, 43(2):178–197, 2009.

[95] A. Simonetto, J. Monteil, and C. Gambella. Real-time city-scale ridesharing via linear assignment problems. *Transportation Research Part C: Emerging Technologies*, 101:208–232, 2019.

[96] A. Smith and J. Anderson. Digital life in 2025: AI, robotics and the future of jobs. Technical Report August 6, Pew Research Center, 2014.

[97] S. L. Smith, M. Pavone, F. Bullo, and E. Frazzoli. Dynamic vehicle routing with priority classes of stochastic demands. *SIAM Journal on Control and Optimization*, 48(5):3224–3245, 2010.

[98] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone. Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore. In G. Meyer and S. Beiker, editors, *Road Vehicle Automation*, pages 229–245. 2014.

[99] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, K. Leyton-Brown, W. P. David Parkes, A. Saxenian, J. Shah, M. Tambe, and A. Teller. Artificial intelligence and life in 2030. Technical report, One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University, Stanford, CA, Sept. 2016.

[100] R. Tachet, O. Sagarra, P. Santi, G. Resta, M. Szell, S. H. Strogatz, and C. Ratti. Title: Scaling law of urban ride sharing. *Nature Publishing Group*, 7:1–6, 2017.

[101] T. Tettamanti, I. Varga, and Z. Szalay. Impacts of autonomous cars from a traffic engineering perspective. *Periodica Polytechnica Transportation Engineering*, 44 (4):244–250, 2016.

[102] H. Topaloglu and W. B. Powell. Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing*, 18(1):31–42, 2006.

[103] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

[104] P. Toth and D. Vigo. *Vehicle Routing Problem, Methods, and Application*. Society for Industrial and Applied Mathematics, 2014.

[105] Toyota. E-Pallete, 2020. URL https://www.toyota-europe.com/startyourimpossible/e-palette. Accessed: 2020-11-02.

[106] M. Tsao, R. Iglesias, and M. Pavone. Stochastic model predictive control for autonomous mobility on demand. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3941–3948, Maui, HI, United States, 2018.

[107] Uber. How surge pricing works, 2020. URL https://www.uber.com/us/en/drive/driver-app/how-surge-works/. Accessed: 2020-11-02.

[108] M. W. Ulmer and S. Streng. Same-day delivery with pickup stations and autonomous vehicles. *Computers and Operations Research*, 108:1–19, 2019.

[109] United Nations, Department of Economic and Social Affairs, Population Division. World urbanization prospects: The 2018 revision. Technical Report ST/ESA/SER.A/420, United Nations, New York, United States, May 2018.

[110] U.S. Census Bureau. Time of departure to go to work (Manhattan borough, NY), 2015. 2015: American Community Survey 5-year estimates (Table B08302).

[111] A. S. van der Veen, J. A. Annema, K. Martens, B. van Arem, and G. H. de Almeida Correia. Operationalizing an indicator of sufficient accessibility – A case study for the city of Rotterdam. *Case Studies on Transport Policy*, 8(4):1360–1370, 2020.

[112] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti. Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706):534–538, 2018.

[113] A. Wallar, W. Schwarting, J. Alonso-Mora, and D. Rus. Optimizing multi-class fleet compositions for shared mobility-as-a-service. In *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2998–3005, Auckland, New Zealand, 2019.

[114] S. Weikl and K. Bogenberger. Relocation strategies and algorithms for free-floating car sharing systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4):100–111, 2013.

[115] J. Wen, J. Zhao, and P. Jaillet. Rebalancing shared mobility-on-demand systems: A reinforcement learning approach. In *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 220–225, Yokohama, Japan, 2017.

[116] K. Winter, O. Cats, K. Martens, and B. van Arem. Identifying user classes for shared and automated mobility services. *European Transport Research Review*, 12 (1), 2020.

[117] K. I. Wong, S. C. Wong, H. Yang, and J. H. Wu. Modeling urban taxi services with multiple user classes and vehicle modes. *Transportation Research Part B: Methodological*, 42(10):985–1007, 2008.

[118] Z. Xu, Y. Yin, and J. Ye. On the supply curve of ride-hailing systems. *Transportation Research Part B: Methodological*, 132:29–43, 2020.

[119] V. S. Zeimpekis, C. D. Tarantilis, G. M. Giaglis, and I. E. Minis. *Dynamic fleet management: Concepts, systems, algorithms & case studies*, volume 38. Springer Science & Business Media, 2007.

[120] V. A. Zeithaml, R. T. Rust, and K. N. Lemon. The customer pyramid: Creating and serving profitable customers. *California Management Review*, 43(4):118–142, 2001.

[121] R. Zhang, F. Rossi, and M. Pavone. Model predictive control of autonomous mobility-on-demand systems. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1382–1389, Stockholm, Sweden, 2016.

[122] W. Zhang and K. Wang. Parking futures: Shared automated vehicles and parking demand reduction trajectories in Atlanta. *Land Use Policy*, 91, 2020.

[123] W. Zhang, S. Guhathakurta, J. Fang, and G. Zhang. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. *Sustainable Cities and Society*, 19:34–45, 2015.

Glossary

List of symbols and notations

Below follows a list of the most frequently used symbols and notations in this thesis divided by the modeling approach.

DARP variants

k	Vehicle index
i, j	Node/request indices
n	Total number of requests
C	Set of SQ classes or compartments
H	Human compartments
F	Freight compartments
K	Vehicles
P	Pick-up nodes and request indices
P^c	Pickup nodes and request indices of SQ class c
D	Delivery nodes
O	Origin nodes o^k of vehicles k
N	$= P \cup D \cup O$
E	Set of edges $(i, j)\ \forall i, j \in N$
G	Graph with node set N and edge set E
$\mathscr{Q}$	Valid visits (k, i) for vehicle k and node i
$\mathscr{X}$	Valid rides (k, i, j) for vehicle k between nodes i and j
L	Vehicle types and driving modes
Z	Discrete street network locations
$Z_{i,j}$	Shortest path between locations i and j in Z
m^k	Type of vehicle k
o^k	Origin node of vehicle k
e^k	Contract start time of vehicle k
l^k	Contract end time of vehicle k
Q^k	Capacity of vehicle k
p^k_{base}	Base fare for attending a passenger using vehicle k

p^k_{time}	Time-dependent rate for attending passenger using vehicle k
c^k_{time}	Time-dependent operational cost of vehicle k
σ^c	Service-level enforcement rate of SQ class c
w^c_{pickup}	Expected max. pickup delay of users in SQ class c
$w^c_{\text{tolerance}}$	Total delay tolerance of users in SQ class c
ρ^c	(Binary) 1 if SQ class c does not allow ridesharing, 0 otherwise
p^c_{base}	Base fare of compartment c
p^c_{time}	Time-dependent fare of compartment c
d^c_{load}	Loading delay of compartment c
d^c_{unload}	Unloading delay of compartment c
d_i	Service duration at node i
q_i	Number of passengers of request i
q^c_i	Number of compartments c requested by i
w_{pickup}	Maximum pickup time delay
w_{ride}	Maximum ride time delay
e_i	Earliest time at node i
l_i	Latest time at node i
c_i	SQ class of request i
q^c_i	Number of compartments c requested by i
$t_{i,j}$	Travel time from node i to node j
$t^m_{i,j}$	Travel time from node i to node j in mode $m \in L$
$x^k_{i,j}$	(Binary) 1 if vehicle k traverses arc (i,j) 0 otherwise
y_i	(Binary) 1 if user i service level is achieved, 0 otherwise
τ^k_i	Arrival time of vehicle k at node i
δ_i	Pickup delay of user i
Δ^k_i	In-vehicle delay of user i in vehicle k
ω^k_i	Load of vehicle k after visiting node i

Matheuristic

Z^*	Regional centers of street network locations Z
K^{PAV}	Company vehicles
K^{FAV}_t	Hireable vehicles at time t
K^{H}_t	Vehicles hired at time t
K^{P}_t	Parked vehicles at time t
$\mathscr{P}^k$	Passengers (picked up requests) of vehicle k
$\mathscr{R}^k$	Assigned requests (non picked up) of vehicle k

$\mathscr{S}^k$	Node itinerary of vehicle k
v^k	Visiting plan $\{\mathscr{P}^k, \mathscr{R}^k, \mathscr{S}^k\}$ of vehicle k
$\mathscr{V}_R^k$	Candidate visiting plans where k is in the *rebalancing* state
$\mathscr{V}_S^k$	Candidate visiting plans where k is in the *servicing* state
v_{idle}^k	Candidate visiting plan where k is in the *idle* state
$\mathscr{V}^k$	Feasible visiting plans $\mathscr{V}_S^k \cup \mathscr{V}_R^k \cup \{v_{\text{idle}}^k\}$ for vehicle k
$\mathscr{V}$	Visiting plans $\bigcup_{k \in K} \mathscr{V}^k$ in ERTV graph
$\mathscr{V}_i$	Visiting plans in $\mathscr{V}$ including request i
$\mathscr{V}_i^{\text{SL}}$	Visiting plans in $\mathscr{V}$ that meet request i target SL
P^{A}	Requests previously assigned to vehicles
B_t	Request batch placed in period t
P_t	$= B_t \cup P^{\text{A}}$
P_t^c	Requests in P_t of class c
O_t^*	Origins $o^k \in Z^*$ of hired vehicles K_t^{H}
P_t^{U}	Pickup nodes of service-level violated requests at time t
J_t	Rebalancing targets $O_t^* \cup P_t^{\text{U}}$

γ_{hire}	Hiring penalty
γ_{sl}	Service-level violation penalty
γ_{reject}	Rejection penalty
κ	Round duration
l^k	Contract deadline of vehicle k
T	Total time horizon
s	Maximal hiring delay

h^k	(Binary) 1 if vehicle $k \in K_t^{\text{FAV}}$ is hired, 0 otherwise
x_v	(Binary) 1 if visiting plan v is chosen, 0 otherwise
y_i	(Binary) 1 if service level of request i is achieved, 0 otherwise
z_i	(Binary) 1 if request i is rejected, 0 otherwise
δ_{iv}	Pickup delay of user i in visiting plan v
Δ_v	Delay sum of all requests in visiting plan v
Δ_v^c	Delay sum of class c requests in visiting plan v
a^c	Number of service-level violations in class c

Approximate Dynamic Programming

N	Set of nodes (locations)
N^g	Set of hexagon bins covering nodes in N at level g
E	Set of directed edges (streets)
L	Discrete geographical area comprising nodes in N
$U(.)$	Function that maps each location in N to an area in L

a	Resource attribute vector
a'	Post-decision attribute
k^a	Vehicle k with attribute a
k_j^{inbound}	Number of vehicles inbound to $j \in N$
$k_j^{\max}$	Max. number of vehicles allowed at $j \in N$
b	Demand attribute vector
d	Elementary decision
b_d	Demand associated wit decision d
t	Time period
$\tau(t,a,d)$	Travel time to k^a carry out decision d departing at t
t'	Post-decision time
T	Time horizon
$\mathscr{A}$	Set of all possible resource attribute vectors
$\mathscr{B}$	Set of all possible demand attribute vectors
$\mathscr{T}$	Set of time periods
R_{ta}	Number of vehicles with attribute vector a at time t
$R_t = (R_{ta})_{a \in \mathscr{A}}$	The resource state vector at time t
R_t^x	Post-decision resource vector
D_{tb}	The number of trips with attribute vector b at time t
$D_t = (D_{tb})_{b \in \mathscr{B}}$	The request state vector at time t
D_t^x	Post-decision demand vector
d^{stay}	Decision to stay parked in the current location
d^{return}	Decision to return to the station (FAV only)
$\mathscr{D}^{\text{R}}$	Set of all decisions to rebalance
$\mathscr{D}^{\text{S}}$	Set of all decisions to service a user
$\mathscr{D}_c^{\text{S}}$	Subset of decisions in $\mathscr{D}^{\text{S}}$ with service quality $c \in C$
$\mathscr{D}_{a,j}$	Decisions leading vehicles k^a to post-decision locations j
$\mathscr{D}$	Set of all decisions
x_{tad}	Number of times decision d is applied to k^a at time t
x_t	Decision vector at time t
$S_t = (R_t, D_t)$	System state vector
S_t^x	Post-decision state vector
a^M	Transition function (attribute a to a')
$S^M(.)$	Transition function (state S_t to S_{t+1})
$S^{M,x}(.)$	Transition function (state S_t to S_t^x)
$\delta_{a'}(.)$	Transition function equals to 1, if $a^M(a,d) = a'$, and 0, otherwise

$\hat{R}_t$	FAVs entering/leaving the system between $t-1$ and t
$\hat{D}_t$	Requests placed between $t-1$ and t
$W_t = (\hat{R}_t, \hat{D}_t)$	Exogenous information arriving between $t-1$ and t
β^k	Platform profit margin when using vehicle k
p^c_{base}	Base fare of request quality class c
p_{time}	Time-dependent fare
c^k_{time}	Time-dependent operational cost of vehicle k
c^c_{delay}	Penalty due to the excess delay w_{delay}
w_{delay}	Excess pickup delay of user from class c
$c^{t,j}_{\text{stay}}$	Cost of staying at location j at time t
C_t	Total profit function at time t
P_t	Total penalty function at time t
Δt_{trip}	Request trip duration
Δt_{pickup}	Request pickup duration
$\Delta t_{\text{rebalance}}$	Rebalance travel duration
Δt_{return}	FAV return to station travel duration
g	Level of aggregation
$\mathcal{G}$	Set of hierarchical aggregation levels
$ta^{(g)}$	Time t and attribute vector a at level g
$(\mathcal{T} \times \mathcal{A})^{(g)}$	Space $\mathcal{T} \times \mathcal{A}$ at the g^{th} aggregation level
$G^g(ta)$	Hierarchical aggregation function returning $ta^{(g)}$ given ta and g
$w^{(g,n)}_{ta}$	Weight put on approximations v_{ta} at level g and iteration n
$\left(\overline{\sigma}^2_{ta}\right)^{(g,n)}$	Variance of estimate $\overline{v}^{(g,n)}_{ta}$
$\left(\overline{\mu}^{(g,n)}_{ta}\right)^2$	Aggregation bias
$\hat{v}^n_{ta}$	Samples drawn from a at time t and iteration n
$\overline{v}^n_{ta}$	Value function approximation of a at time t and iteration n
$V_t(.)$	Value function around S_t
$\overline{V}_t(.)$	Value function approximation around S^x_t
n	ADP algorithm iteration
I	Number of iterations
γ	Discount factor
α_n	Stepsize at iteration n
$\overline{\alpha}$	Constant stepsize (McClain's rule)
π	Policy
Π	Set of potential decision functions
$\mathcal{S}$	Set of feasible states
$\mathcal{X}_t$	Set of feasible decisions at time t
$X^\pi_t(.)$	Decision function representing a policy π that maps S_t to x_t

Facility location

x_j	1, if a facility is located at $j \in N$, 0 otherwise
$t_{i,j}$	Travel time between nodes $i, j \in N$
s	The maximal service delay of a vehicle departing from a facility
$N_{p,s}$	Subset of locations able to reach location $p \in N$ within s time units

List of abbreviations

ACC	Adaptive cruise control
AD	Automated driving
ADP	Approximate dynamic programming
AMoD	Autonomous mobility on demand
AMoD-H	Autonomous mobility on demand with hiring
AV	Autonomous vehicle
AVZ	Autonomous vehicle zone
B&P	Branch and price
cA2C	Contextual multi-agent actor-critic
CPU	Central processing unit
CV	Conventional vehicle
CVRP	Capacitated VRP
CVZ	Conventional vehicle zone
DAR	Dial a ride
DARP	Dial-a-ride problem
DD	Dynamic and deterministic
DQN	Deep Q-network
DS	Dynamic and stochastic
DSDARP	Dynamic and stochastic dial-a-ride problem
DSVRP	Dynamic stochastic VRP
DTRP	Dynamic traveling repairperson problem
DV	Dual-mode vehicle
DVRP	Dynamic VRP
ERTV	Extended request trip vehicle
ETA	Estimated time of arrival
FAV	Freelance autonomous vehicle
FMS	Fleet management system
GPDP	General pickup and delivery problem
GPS	Global positioning system
GRASP	Greedy randomized adaptive search procedure
HDARP	Heterogeneous dial-a-ride problem
HVRP	Heterogeneous fleet VRP
ICT	Information and communication technology
ID	In-vehicle delay
LP	Linear programming
LSTM	Long short-term memory

MAV	Modular autonomous vehicle
MDSDDARP	Multi-depot site-dependent dial-a-ride problem
MILP	Mixed-integer linear programming
MIP	Mixed-integer programming
MoD	Mobility on demand
MPC	Model predictive control
NS	Neighborhood Search
OD	Occasional drivers
OD	Origin and destination
PAV	Platform-owned autonomous vehicle
PD	Pickup delay
PDP	Pickup and delivery problem
PFIT	People and freight integrated transportation
RAM	Random access memory
RC	Regional center
RL	Reinforcement learning
RR	Request request
RSL	Regional service level
RTV	Request trip vehicle
RV	Request vehicle
RVRP	Rich vehicle routing problems
SAE	Society of automotive engineers
SAMoD	Shared autonomous mobility on demand
SARP	Share-a-ride problem
SARPLP	Share-a-ride with parcel lockers problem
SAV	Shared autonomous vehicle
SD	Static and deterministic
SDVRP	Site-dependent vehicle routing problem
SQC	Service quality contract
SS	Static and stochastic
TNC	Transportation network company
TOD	Transportation on demand
TS	Tabu search
TW	Time window
VFA	Value function approximation
VKT	Vehicle kilometers traveled
VOTT	Value of travel time
VRP	Vehicle routing problem
VRPPD	VRP with pickups and deliveries
VRP-SL	Vehicle routing problem with service levels
VRPTW	VRP with time windows
WIMSE	Weighting by inverse mean squared errors

Samenvatting

Autonome voertuigen (AV's) zijn de sleutel tot een toekomst van gedeelde mobiliteit waarin vervoer efficiënter, gemakkelijker en goedkoper is. Het ideaalbeeld van een AV-utopie kan echter alleen werkelijkheid worden als de meerderheid van de gebruikers erop vertrouwt dat autonome systemen voor vraaggestuurde mobiliteit (AMoD, Autonomous Mobility-on-Demand) als de kwaliteit van de dienstverlening op hetzelfde niveau is in vergelijking tot het bezit van een eigen voertuig. Aangezien de perceptie van kwaliteit zeer subjectief is, stellen wij een meer gepersonaliseerde benadering van mobiliteit op aanvraag voor, waarbij gebruikers op basis van hun wensen in kwaliteitsklassen voor vervoersdiensten worden ingedeeld. Deze klassen omvatten minimumeisen zoals responsiviteit en privacy, waardoor wij een reeks gebruikersprofielen kunnen modelleren die geformaliseerd zijn aan de hand van strikte kwaliteitscontracten voor vervoersdiensten. Door deze contracten na te leven, kunnen aanbieders het vertrouwen van de gebruikers winnen en hun loyaliteit versterken, wat op grotere schaal kan bijdragen tot een snellere overgang naar een toekomst met gedeelde mobiliteit.

In deze dissertatie wordt een reeks strategieën gepresenteerd die de kwaliteit van de dienstverlening garanderen in alle operationele scenario's die zich voordoen in de tijdlijn van de invoering van AV-technologie. De eerste maatstaf voor de kwaliteit van de dienstverlening bij autonoom vervoer is de veiligheid. Tijdens een overgangsfase naar volledige automatisering zal de exploitatie van AV waarschijnlijk worden beperkt tot gebieden waar een veilige exploitatie is gegarandeerd, wat leidt tot de vorming van hybride wegennetten die bestaan uit autonome en niet-autonome voertuigzones. Naast de veiligheid is dekking de maatstaf om de kwaliteit van de dienstverlening te kunnen beoordelen. Dit is ingewikkeld omdat de mobiliteitsdiensten toegang moeten hebben tot gebieden die wel en gebieden die niet klaar zijn voor AV's. In deze dissertatie worden dan ook oplossingen voorgesteld om de uitdagingen te overwinnen die een dergelijk overgangsscenario met zich meebrengt, waarbij infrastructuren, regelgevende maatregelen en AV-technologie zich geleidelijk ontwikkelen.

Ervan uitgaande dat wijdverspreid geautomatiseerd rijden de nieuwe status quo is, zijn we begonnen met het modelleren van rijke autonome transportscenario's die bestaan uit heterogene gebruikers en voertuigen. Centraal in deze analyse staat het vinden van een goed evenwicht tussen de omvang van de vloot en de kwaliteit van de dienstverlening. In traditionele AMoD-systemen kunnen aanbieders beperkt in de mogelijkheden om ontevredenheid bij de gebruikers te voorkomen, dit is vooral een kwestie van een ruim aanbod van voertuigen creeren. Wanneer de vraag groter is dan het aanbod, krijgen gebruikers onvermijdelijk te maken met langere vertragingen of zelfs weigeringen,

wat uiteindelijk leidt tot het verlies van vertrouwen in de dienst. Deze tekortkomingen zullende toekomstige vervoerssystemen echter alleen parten spelen als het bepalen van de omvang en samenstelling van de vloot een strategische beslissing blijft. In tegenstelling tot de meeste verwante literatuur, onderzoekt deze dissertatie een verspreid AV-bezitscenario, waarbij rittendeelplatformen af en toe beschikbare AV's in privébezit op aanvraag kunnen huren. In dit scenario kunnen klanten tegelijkertijd AV's bezitten en delen, een opzet die beter lijkt op de werking van de huidige transportnetwerkbedrijven (TNC's, Transportation Network Companies), die volledig afhankelijk zijn van microoperators. Als gevolg daarvan kunnen AMoD-systemen het aanbod van voertuigen op korte termijn vergroten of verkleinen, waardoor de vlootomvang wordt verschoven naar het operationele planningsniveau.

Bovendien moet het systeem, in overeenkomst met andere vervoerswijzen, rekening houden met een gediversifieerd gebruikersbestand met uiteenlopende verwachtingen met betrekking tot de kwaliteit van de dienstverlening. Deze opzet geeft aanbieders meer speelruimte om de vertragingstoleranties van verzoeken te onderzoeken en zo efficiënte routes te ontwerpen. Om de verwachtingen van de gebruikers in evenwicht te brengen en een overaanbod van voertuigen te vermijden, stellen we een multi-objectieve matheuristiek voor die dynamisch AV's van derden huurt om aan de vraag te voldoen. Onze aanpak vormt een aanvulling op de recente literatuur door aanbieders in staat te stellen prioriteit te geven aan verschillende klantensegmenten, naast het kiezen van de precieze afweging tussen het voldoen aan de behoeften van elk segment en het inhuren van extra voertuigen. Op die manier kan het optimalisatieproces bij een tekort aan voortuigen de oplossing voor het afstemmen van de ritten sturen in de richting van de verzoeken van de gebruikers, in volgorde van belangrijkheid (bv. het meest lucratieve eerst). Om het maximum te halen uit de momenteel werkende voertuigenwordt er ook een herpositioneringsalgoritme ontwikkelt dat de disbalans tussen vraag en aanbod herstelt door de schendingen van het serviceniveau van de gebruikers als stimulans te gebruiken.

Om anticiperende besluitvorming mogelijk te maken, integreert dit proefschrift de stochastische informatie over zowel het aanbod van AV in privébezit als de heterogene passagiersvraag in het vlootbeheerproces. Wij stellen een zelflerende optimalisatieaanpak voor die gebruik maakt van de duale variabelen van het onderliggende toewijzingsprobleem om de marginale waarde van voertuigen op elk tijdstip en elke locatie iteratief te benaderen onder verschillende beschikbaarheidsinstellingen. Deze benaderingen worden op hun beurt gebruikt in de minimalisatiefunctie van het optimalisatieprobleem om de effecten af te wegen van herdistribueren, herbalanceren en het incidenteel inhuren van voertuigen. Door gebruik te maken van de historische kennis over zowel vraag- als aanbodpatronen tonen wij aan dat AMoD-systemen aanzienlijk beter zijn toegerust om aan de behoeften van de gebruikers te voldoen, met als bijkomend voordeel dat zij niet noodzakelijk grote AV-vloten hoeven te bezitten.

Doorgaans versterken strategieën voor vlootbeheer op basis van leerprocessen de vooringenomenheid in de vraaggegevens, waardoor ze vaak in de richting gaan van de meest welvarende en dichtbevolkte gebieden van steden, waar reeds veel alternatieve mobiliteitskeuzes bestaan. Hoewel lucratief voor de aanbieders, druist deze strategie van vlootbeheer in tegen een bredere doelstelling van de stad om de bereikbaarheid evenredig te verdelen over alle regio's en bevolkingsgroepen. Om de vertekeningen in

de vraag te compenseren, onderzoeken we in welke mate een beleid van tariefsubsidies het leerproces kan sturen in de richting van het sturen van voertuigen naar bepaalde regio's waar de bereikbaarheid onvoldoende is. Onze resultaten suggereren dat beleidsmakers, door een adequaat systeem van stimulansen te gebruiken, vervoerders kunnen aansporen om niet alleen de lucratieve keuze te maken, waardoor AV's worden gebruikt voor evenredig verdeelde mobiliteit.

Ten slotte, zodra we strategieën hebben ontworpen die de doelstellingen van steden, onafhankelijke eigenaars, vlooteigenaars en gebruikers in evenwicht brengen, richten we ons op een andere aanpak om de productiviteit van het wagenpark in stedelijke omgevingen te maximaliseren. Hoe efficiënt een vlootoptimalisatiemethode ook kan zijn, door AV's te beperken tot één soort goederen (d.w.z. mensen), wordt het vlootgebruik en bijgevolg de winst beperkt door de vraagpatronen van de passagiers. Naarmate de autonome technologie evolueert, ontstaan er echter nieuwe mogelijkheden om het gebruik van mobiliteitsmiddelen te verbeteren. We eindigen dit proefschrift met een model voor een veelzijdig transportsysteem waar gecompartimenteerde AV's voor gemengde doeleinden zowel passagiers als goederen tegelijk kunnen vervoeren. Met de groei van e-commerce en leveringen op dezelfde dag, biedt onze aanpak een uitgangspunt om meer flexibele korteafstandsintegratiesystemen te bestuderen om passagiers- en goederenstromen te consolideren.

Summary

Autonomous vehicles (AVs) have been heralded as the key to unlock a shared mobility future where transportation is more efficient, convenient, and cheaper. However, the AV utopia can only come to fruition if the majority of users trust that *autonomous mobility-on-demand* (AMoD) systems are on a par with owning a vehicle in terms of service quality. Once the perception of quality is highly subjective, we propose a more personalized approach to on-demand mobility, in which users are segmented into service quality classes. These classes comprise minimum requirements regarding responsiveness and privacy, allowing us to model a series of user profiles formalized using strict service quality contracts. By honoring these contracts, providers can build users' trust and gain their loyalty, which on a grander scheme can contribute to a faster transition to a shared mobility future.

This thesis presents a series of strategies to guaranteeing service quality throughout operational scenarios arising in the timeline of AV technology deployment. First, a precondition to providing service quality in autonomous transportation is safety. During a transition phase to full automation, AV operation will likely be restricted to areas where safe operations are guaranteed, leading to the formation of hybrid street networks comprised of autonomous and non-autonomous vehicle zones. In this setting, meeting user service quality expectations is primarily a matter of coverage, once mobility services will have to access both AV-ready and not AV-ready areas. Accordingly, this thesis proposes solutions to overcome the challenges entailed by such a transition scenario, where infrastructures, regulatory measures, and AV technology are gradually evolving.

Then, assuming that widespread automated driving is the new status quo, we set out to model rich autonomous transportation scenarios comprised of heterogeneous users and vehicles. Central to our analysis is finding an adequate tradeoff between fleet size and service quality. In traditional AMoD systems, providers can do only so much to prevent user dissatisfaction since, to some extent, this is a matter of having enough vehicles. When the demand outstrips the supply, users inevitably experience longer delays or even rejections, ultimately undermining trust in the service. However, these shortcomings may plague future transportation systems only if setting the fleet size and mix remains a strategic decision. In contrast to most related literature, this thesis investigates a disseminated AV ownership scenario, where ridesharing platforms can occasionally hire available privately-owned AVs on-demand. In this scenario, customers can simultaneously own and share AVs, a setup that better resembles the operation of today's *transportation network companies* (TNCs), which rely entirely on micro-operators.

As a result, AMoD systems can increase and decrease vehicle supply in the short term, thus shifting fleet sizing to the operational planning level.

Moreover, analogously to other transportation modes, we consider that the system must deal with a diversified user base with different service quality expectations. This setup allows providers greater leeway to explore requests' delay tolerances to design efficient routes. To balance user expectations and avoid an oversupply of vehicles, we propose a multi-objective matheuristic that dynamically hires third-party AVs to meet the demand. Our approach adds to recent literature by allowing providers to prioritize different customer segments, besides choosing the exact tradeoff between meeting each segment's needs and hiring extra vehicles. This way, when vehicles are lacking, the optimization process can steer the ride-matching solution towards addressing user requests in order of importance (e.g., most lucrative first). To make the most of currently working vehicles, we also design a repositioning algorithm that fixes supply and demand imbalances using users' service level violations as stimuli.

Further, to enable anticipatory decision making, this thesis incorporates the stochastic information surrounding both privately-owned AV supply and heterogeneous passenger demand in the fleet management process. We propose a learning-based optimization approach that uses the underlying assignment problem's dual variables to iteratively approximate the marginal value of vehicles at each time and location under different availability settings. In turn, these approximations are used in the optimization problem's objective function to weigh the downstream impact of dispatching, rebalancing, and occasionally hiring vehicles. By harnessing the historical knowledge regarding both demand and supply patterns, we show that AMoD providers are substantially better equipped to meet user needs without necessarily having to own large AV fleets.

Typically, learning-based fleet management strategies end up reinforcing biases present in the demand data, therefore frequently moving towards cities' most affluent and densely populated areas, where alternative mobility choices already abound. Although lucrative for providers, this fleet management strategy runs counter to a broader city goal of equitably distributing accessibility across all regions and population demographics. To counterbalance the demand biases, we investigate the extent to which fare subsidization policies can drive the learning process towards sending vehicles to targeted regions where accessibility is lacking. Our results suggest that by using an adequate scheme of incentives, policymakers can orchestrate transportation providers to diminish the insidious effects of "cream-skimming" practices, thus using AVs in favor of mobility equity.

Lastly, once we have designed strategies that balance the goals of cities, independent owners, fleet owners, and users, we focus on a different approach to maximizing fleet productivity in urban environments. No matter how efficient a fleet optimization method can be, by limiting AVs to service a single commodity type (i.e., people), fleet utilization and consequently profits are bounded by passenger demand patterns. As autonomous technology evolves, however, new opportunities to improve asset utilization arise. We end this thesis with a model for a versatile transportation system where mixed-purpose compartmentalized AVs can address both passengers and goods simultaneously. With the growth of e-commerce and same-day deliveries, our approach provides a starting point to study more flexible short-haul integration systems to consolidate passenger and freight flows.

Curriculum vitae

Breno Alves Beirigo was born on January 15, 1991 in Divinópolis, Minas Gerais, Brazil. His passion for computers led him to obtain a technical degree in informatics at the Federal Center of Technological Education of Minas Gerais (CEFET-MG) in 2009, and later a bachelor's degree in Computer Science at the Federal University of Viçosa (UFV), where he graduated with honors in 2014. As part of his bachelor studies, he developed several junior research projects, and in 2011 he was granted a one-year scholarship to study advanced sensor applications at the Hanze University of Technology in Assen, the Netherlands. In 2016, Breno obtained a master's degree in Computer Science from UFV. His work focused on metaheuristics for combinatorial optimization problems arising in tourist trip planning. In parallel with his master's degree, he has come full circle by also working as a lecturer at CEFET-MG, teaching technology-related disciplines to both technical and bachelor degree students. From November 2016, he started a Ph.D. project in the Section of Transport Engineering & Logistics, Department of Maritime & Transport Technology, Delft University of Technology, the Netherlands, under the supervision of Prof. Rudy R. Negenborn and Dr. Frederik Schulte, the results of which are presented in this dissertation. His research interests lie in the intersection of transportation, operations research, and artificial intelligence.

Publications

1. B. A. Beirigo, F. Schulte, and R. R. Negenborn. A learning-based optimization approach for autonomous ridesharing platforms with service level contracts and on-demand hiring of idle vehicles. *Transportation Science* (in press, 2021).
2. B. A. Beirigo, F. Schulte, and R. R. Negenborn. A business class for autonomous mobility-on-demand: Modeling service quality contracts in ridesharing systems. *Submitted to a journal*.
3. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1325–1330, Maui, HI, United States, 2018.

4. B. A. Beirigo, F. Schulte, R. R. Negenborn. Overcoming mobility poverty with shared autonomous vehicles: A learning-based optimization approach for Rotterdam Zuid. In *Proceedings of the 11th International Conference on Computational Logistics*, pages 492-506, Enschede, the Netherlands, 2020.
5. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Integrating people and freight transportation using shared autonomous vehicles with compartments. In *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems*, pages 392–397, Savona, Italy, 2018.

Extended abstracts

1. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Modeling service class Constraints in autonomous mobility-on-demand systems: A data-driven approach for dispatching and rebalancing vehicles. In *Proceedings of the 7th INFORMS Transportation Science and Logistics Society Workshop*, Vienna, Austria, 2019.
2. B. A. Beirigo, F. Schulte, and R. R. Negenborn. How mobility services can become the better alternative to a self-owned vehicle: Modeling quality in mobility-as-a-service systems. In *Proceedings of the International Conference on Integrated Transport (CIT)*, KTH Royal Institute of Technology, Stockholm, Sweden, 2019.
3. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Reliable mobility-on-demand platforms: Hiring private-owned AVs to ensure service quality. In *Proceedings of the TRAIL PhD Congress*, Utrecht, the Netherlands, 2019.
4. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Dual-mode vehicle routing in mixed autonomous & non-autonomous zone networks. In *Proceedings of the TRAIL PhD Congress*, Utrecht, the Netherlands, 2018.
5. B. A. Beirigo, F. Schulte, and R. R. Negenborn. Mixed-purpose shared autonomous vehicles with parcel lockers: Insights on integrating passenger and freight transportation. In *Proceedings of the TRAIL PhD Congress*, Utrecht, the Netherlands, 2017.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 200 titles see the TRAIL website: www.rsTRAIL.nl. The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure, and logistics.

Beirigo, B.A., Dynamic Fleet Management for Autonomous Vehicles: Learning- and optimization-based strategies, T2021/12, March 2021, TRAIL Thesis Series, the Netherlands

Zhang, B., Taking Back the Wheel: Transition of control from automated cars and trucks to manual driving, T2021/11, February 2021, TRAIL Thesis Series, the Netherlands

Boelhouwer, A., Exploring, Developing and Evaluating In-Car HMI to Support Appropriate use of Automated Cars, T2021/10, January 2021, TRAIL Thesis Series, the Netherlands

Li, X., Development of an Integrity Analytical Model to Predict the Wet Collapse Pressure of Flexible Risers, T2021/9, February 2021, TRAIL Thesis Series, the Netherlands

Li, Z., Surface Crack Growth in Metallic Pipes Reinforced with Composite Repair System, T2021/8, January 2021, TRAIL Thesis Series, the Netherlands

Gavriilidou, A., Cyclists in Motion: From data collection to behavioural models, T2021/7, February 2021, TRAIL Thesis Series, the Netherlands

Methorst, R., Exploring the Pedestrians Realm: An overview of insights needed for developing a generative system approach to walkability, T2021/6, February 2021, TRAIL Thesis Series, the Netherlands

Walker, F., To Trust or Not to Trust? Assessment and calibration of driver trust in automated vehicles, T2021/5, February 2021, TRAIL Thesis Series, the Netherlands

Schneider, F., Spatial Activity-travel Patterns of Cyclists, T2021/4, February 2021, TRAIL Thesis Series, the Netherlands

Madadi, B., Design and Optimization of Road Networks for Automated Vehicles, T2021/3, January 2021, TRAIL Thesis Series, the Netherlands

Krabbenborg, L.D.M., Tradable Credits for Congestion Management: support/reject?, T2021/2, January 2021, TRAIL Thesis Series, the Netherlands

Castelein, B., Accommodating Cold Logistics Chains in Seaport Clusters: The development of the reefer container market and its implications for logistics and policy, T2021/1, January 2021, TRAIL Thesis Series, the Netherlands

Huang, B., The Influence of Positive Interventions on Cycling, T2020/20, December 2020, TRAIL Thesis Series, the Netherlands

Xiao, L., Cooperative Adaptive Cruise Control Vehicles on Highways: Modelling and Traffic Flow Characteristics, T2020/19, December 2020, TRAIL Thesis Series, the Netherlands

Polinder, G.J., New Models and Applications for Railway Timetabling, T2020/18, December 2020, TRAIL Thesis Series, the Netherlands

Scharpff, J.C.D., Collective Decision Making trough Self-regulation, T2020/17, November 2020, TRAIL Thesis Series, the Netherlands

Guo, W. Optimization of Synchromodal Matching Platforms under Uncertainties, T2020/16, November 2020, TRAIL Thesis Series, the Netherlands

Narayan, J., Design and Analysis of On-Demand Mobility Systems, T2020/15, October 2020, TRAIL Thesis Series, the Netherlands

Gong, X., Using Social Media to Characterise Crowds in City Events for Crowd Management, T2020/14, September 2020, TRAIL Thesis Series, the Netherlands

Rijal, A., Managing External Temporal Constraints in Manual Warehouses, T2020/13, September 2020, TRAIL Thesis Series, the Netherlands

Alonso González, M.J., Demand for Urban Pooled On-Demand Services: Attitudes, preferences and usage, T2020/12, July 2020, TRAIL Thesis Series, the Netherlands

Alwosheel, A.S.A., Trustworthy and Explainable Artificial Neural Networks for choice Behaviour Analysis, T2020/11, July 2020, TRAIL Thesis Series, the Netherlands

Zeng, Q., A New Composite Indicator of Company Performance Measurement from Economic and Environmental Perspectives for Motor Vehicle Manufacturers, T2020/10, May 2020, TRAIL Thesis Series, the Netherlands

Mirzaei, M., Advanced Storage and Retrieval Policies in Automated Warehouses, T2020/9, April 2020, TRAIL Thesis Series, the Netherlands

Nordhoff, S., User Acceptance of Automated Vehicles in Public Transport, T2020/8, April 2020, TRAIL Thesis Series, the Netherlands

Winter, M.K.E., Providing Public Transport by Self-Driving Vehicles: User preferences, fleet operation, and parking management, T2020/7, April 2020, TRAIL Thesis Series, the Netherlands

Mullakkal-Babu, F.A., Modelling Safety Impacts of Automated Driving Systems in Multi-Lane Traffic, T2020/6, March 2020, TRAIL Thesis Series, the Netherlands

Krishnakumari, P.K., Multiscale Pattern Recognition of Transport Network Dynamics and its Applications: A bird's eye view on transport, T2020/5, February 2020, TRAIL Thesis Series, the Netherlands