

Optimizing the utilization of existing vehicle flows in last-mile passenger transport and logistics systems

Présentée le 3 novembre 2023

Faculté de l'environnement naturel, architectural et construit
Laboratoire de systèmes de transports urbains
Programme doctoral en génie civil et environnement

pour l'obtention du grade de Docteur ès Sciences

par

Patrick Stefan Adriaan STOKKINK

Acceptée sur proposition du jury

Prof. A. J. Sonta, président du jury
Prof. N. Geroliminis, directeur de thèse
Prof. D. Pisinger, rapporteur
Dr M. Kaspi, rapporteur
Prof. M. Bierlaire, rapporteur

"And I knew exactly what to do.
But in a much more real sense, I
had no idea what to do."

Michael Scott
The Office, Season 5, Episode 15

Acknowledgements

The four years of my PhD have truly been a rollercoaster. Moving to another country is already a challenge, let alone when a global pandemic starts shortly after. Despite this challenge, I can confidently say that my time in Lausanne and at EPFL has been amazing. The environment at EPFL has been enriching in knowledge and experiences. On top of that, being able to hike in the Swiss Alps every weekend was like a dream come true.

First, I would like to thank my supervisor Prof. Nikolas Geroliminis for giving me the opportunity to join LUTS. From the start, the collaboration has been highly enjoyable and successful. His guidance, advice, and encouragement have helped me set the foundation of my academic career. Most importantly, I would like to thank Nikolas for his kindness and understanding and the value he attributes to my personal life and mental health. Without his continuous support, my PhD would not have been such a success.

I would like to thank my co-authors André de Palma and Jean-François Cordeau for their useful comments, interesting discussions, and valuable collaborations that helped to shape our research. In addition to this, I would like to thank Jean-François for hosting me at the HEC in Montréal during the summer of 2022.

I would like to cordially thank all the members of my jury, Prof. Michel Bierlaire, Prof. Mor Kaspi, Prof. David Pissinger, and Prof. Andrew Sonta, for the time and energy they put into reviewing my thesis, as well as for all the constructive comments and the interesting discussion that we had in the oral exam.

I would like to thank my colleagues in LUTS and TRANSP-OR for the great times we had during and after work and at conferences. The lunches, coffee breaks, and after-work beers were the highlights of my work day. A special thanks goes to Dimitris and Zhenyu for their support and friendship.

I would like to thank my friends in the Netherlands. Especially, my friends from Erasmus University: Menno, Jaron, Celine, Emma, and Michelle for visiting for their long-distance friendship and for taking the time to repeatedly visit me in Switzerland. And my friends in Switzerland. Especially, Shahab, Shamim, Gonçalo, and Rita for the infinitely many hikes and dinners.

A large thanks goes to my parents and my sister for their moral support during the time we spent apart. Due to our weekly video calls, they never felt far away

and they were always there for me in case I needed them. I also appreciate the time we spent together in Switzerland, during which I could show them the beauty of the place that was my home.

Finally, my biggest thanks go to Fleur, who has been my mental supporter, hiking partner, travel partner, chef cook, peer reviewer for papers and presentations, and plus-one to conferences and dinners. I want to thank her for taking the leap and joining me in Switzerland and supporting me when I was not feeling well. I want to thank her for the infinitely many adventures that we have been on in and outside Switzerland and of which I am sure we will have many more in the future. My time in Switzerland would not have nearly been the same without her.

Lausanne, 13/10/2023

Patrick Stokkink

Abstract

The new era of shared economy has raised our expectations to make mobility more sustainable through better utilization of existing resources and capacity. In this thesis, we focus on the design of transport systems that stimulate multi-purpose trips with the aim of reducing congestion while simultaneously leveraging the existing commuters better. Multi-purpose trips can improve the efficiency, sustainability, and profitability of passenger transport systems, through vehicle relocation in vehicle-sharing systems and ride-sharing. Similar improvements can be made in last-mile logistics systems, through crowd-shipping.

A predictive user-based relocation approach through incentives is proposed for car relocation in one-way car-sharing systems. This approach consists of a bi-level optimization approach to find the optimal incentive and a Markov chain to describe the state of the system. Numerical results indicate that these user-based relocations can significantly improve the profit and the service level of a car-sharing system and are more sustainable than staff-based relocations.

The effect of congestion on ride-sharing is studied. Numerical results indicate that ride-sharing is more appealing during congestion, due to the high availability of riders and drivers. Thereby, theoretical and numerical results show that bottleneck congestion makes the schedules of riders and drivers more flexible, thereby increasing the matching opportunities. Transfers of riders between modes and between drivers can further improve the performance of a ride-sharing system. By allowing for transfers, the trips of the rider and the driver only need to be partially similar. The problem is formulated as a path-based integer programming problem and travel time uncertainty is included by reformulating the problem as a stochastic programming problem.

Crowd-shipping is a last-mile delivery concept in which commuters pick up and deliver parcels on their pre-existing paths. By constructing depot locations for picking up parcels, more potential crowd-shippers can be attracted and the service area can be extended. Numerical results show that determining these depot locations using predictive strategies can improve the overall performance of the system by 15% compared to when non-predictive strategies are used. Using transfers between crowd-shippers allows for further expanding the service area and improving the overall performance. A column and row generation approach is proposed to

solve the problem of matching parcels to crowd-shippers. Numerical results show that transfers can improve the service level and profit of the system by 30%.

Keywords: Car-sharing, Ride-sharing, Crowd-shipping, Transfers, Mixed integer linear programming, Optimization under uncertainty, Large-scale optimization

Résumé

La nouvelle ère de l'économie de partage a renforcé nos attentes pour rendre la mobilité plus durable grâce à une meilleure utilisation des ressources et des capacités existantes. Dans cette thèse, nous nous focalisons sur la conception des systèmes de transport qui favorisent des trajets à usage multiple, dans l'objectif de réduire les embouteillages routiers, tout en optimisant l'exploitation des flux pendulaires déjà existants. Les trajets à usage multiple se révèlent être une voie prometteuse pour améliorer l'efficacité, la durabilité et la rentabilité des systèmes de transport des passagers, par le biais de la relocalisation des véhicules au sein de plateformes d'auto-partage et de co-voiturage, ainsi que par les systèmes logistiques du dernier kilomètre, à travers le crowd-shipping.

Une approche prédictive de relocalisation basée sur l'utilisateur à travers des incitations est proposée pour la relocalisation des véhicules dans les systèmes d'auto-partage à sens unique. Cette approche repose sur une optimisation à deux niveaux visant à déterminer l'incitation optimale et sur une chaîne de Markov pour caractériser l'état du système. Les résultats numériques indiquent que ces relocalisations basées sur l'utilisateur ont le potentiel d'améliorer de manière significative la rentabilité et l'efficacité du service du système d'auto-partage, surpassant les critères de durabilité des stratégies de relocalisation basées sur le personnel.

L'impact des embouteillages routiers sur le covoiturage est l'objet de cette étude approfondie. Les résultats numériques mettent en évidence le fait que le covoiturage devient d'autant plus attractif en période d'embouteillage routier, du fait de l'abondance de passagers et de conducteurs disponibles. C'est ainsi que les résultats théoriques et numériques démontrent que les embouteillages, considérés comme une contrainte, rendent l'aménagement temporel des passagers et des conducteurs plus flexibles, favorisant ainsi les opportunités de mise en relation entre eux. Les transferts des passagers entre les divers moyens de transport et les conducteurs peuvent optimiser davantage les performances du système de covoiturage. En permettant ces transferts, les trajets des passagers et des conducteurs n'ont besoin d'être que partiellement similaires. Le problème est formulé sous la forme d'une programmation linéaire en nombres entiers (PLNE) basée sur les trajets, et l'incertitude liée au temps de trajet est intégrée en reformulant le problème en tant que problème de programmation stochastique.

Le crowd-shipping est un concept de livraison du dernier kilomètre, où les pendulaires s'engagent à récupérer et acheminer des colis le long de leurs trajets préexistants. En établissant des emplacements de dépôt pour la prise en charge des colis, un plus grand nombre de participants potentiels au crowd-shipping peut être attiré, et la zone de service peut être plus étendue. Les résultats numériques démontrent que la détermination de ces emplacements de dépôt à l'aide de stratégies prédictives peut améliorer les performances globales du système de 15 % par rapport à l'utilisation de stratégies non prédictives. L'utilisation de transferts entre les participants au crowd-shipping permet d'étendre davantage la zone de service et d'améliorer la performance globale. La méthode de génération de colonnes et de lignes est proposée pour résoudre le problème d'appariement des colis avec les participants au crowd-shipping. Les résultats numériques indiquent que les transferts peuvent améliorer le niveau de service et la rentabilité du système de l'ordre de 30 %.

Mots clés: Auto-partage, Covoiturage, Crowd-shipping, Transferts, Programmation linéaire mixte en nombres entiers (PLMNE), Optimisation sous incertitude, Optimisation à grande échelle

Contents

1	Introduction	1
1.1	Motivation and background	2
1.1.1	State-of-the-art on relocation in car-sharing systems	5
1.1.2	State-of-the-art on ride-sharing	8
1.1.3	State-of-the-art on crowd-shipping	9
1.2	Thesis objectives	13
1.3	Thesis contributions	15
1.4	Thesis structure	19
I	First and last-mile mobility systems	21
2	Predictive user-based relocation through incentives in one-way car-sharing systems	23
2.1	Introduction	23
2.2	Problem description and modeling framework	24
2.2.1	Problem description	24
2.2.2	Relocation policy	25
2.2.3	Modeling the state of the system and vehicle imbalances	27
2.2.4	Staff-based relocations	29
2.2.5	User decision-making	30
2.2.6	Learning from user behaviour	33
2.3	Experimental results	34
2.3.1	Case study: Grenoble car-sharing system	35
2.3.2	Simulation model	36
2.3.3	Model evaluation	37
2.3.4	Learning evaluation	41
2.3.5	Increasing user flexibility	42
2.3.6	Adapting staff-based relocation policy	45
2.3.7	Sensitivity analysis: user value of time	47
2.4	Summary	49

3	Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users	51
3.1	Introduction	51
3.2	Matching without congestion	53
3.2.1	Basic carpooling model: equal number of drivers and passengers	55
3.2.2	Unequal number of drivers and passengers	58
3.2.3	Scheduling delay	59
3.3	Matching in the presence of bottleneck congestion	60
3.3.1	Equilibrium analysis	65
3.4	Numerical results	67
3.4.1	Optimal matching without congestion	67
3.4.2	Optimal matching under bottleneck congestion	68
3.5	Summary	76
4	Multi-modal ride-matching with transfers and travel-time uncertainty	79
4.1	Introduction	79
4.2	Deterministic matching	81
4.2.1	Assumptions and mathematical formulation	81
4.2.2	Computation of costs and parameter values	85
4.2.3	Departure time choice	86
4.3	Two-stage stochastic matching	87
4.3.1	Assumptions and mathematical formulation	87
4.3.2	Computation of costs and parameter values	92
4.3.3	Benchmarks	93
4.4	Results	94
4.4.1	Case study	94
4.4.2	Influence of transfers on modal split, costs, and VHT	96
4.4.3	Spatio-temporal distribution of riders	97
4.4.4	Sensitivity analysis	99
4.4.5	Stochastic programming results	101
4.5	Summary	104
II	Last-mile logistics systems	105
5	A continuum approximation approach to the depot location problem in a crowd-shipping system	107
5.1	Introduction	107
5.2	Methodology	109

5.2.1	Discrete formulation	110
5.2.2	Continuum approximation	114
5.2.3	Determining depot locations	119
5.3	Discrete event simulator	121
5.3.1	Stage 2: parcel-depot assignment	122
5.3.2	Stage 3: parcel-crowd-shipper matching	123
5.4	Results	125
5.4.1	Case study	125
5.4.2	Comparison of continuum approximation to static and dynamic assignment strategies	127
5.4.3	Comparison of CA approach to discrete formulation	129
5.4.4	Results on the network	131
5.4.5	Optimal number of depots	132
5.4.6	Comparison to non-predictive strategies	133
5.5	Summary	135
6	A column and row generation approach to the crowd-shipping problem with transfers	137
6.1	Introduction	137
6.2	Problem description and formulation	138
6.2.1	Concepts and notation	138
6.2.2	Mathematical formulation	140
6.3	Methodology	141
6.3.1	Master problem	141
6.3.2	Pricing problem	142
6.3.3	Shortest path algorithm - Graph construction	144
6.3.4	Modified Dijkstra's algorithm	145
6.3.5	Locker and shipper capacity	149
6.4	Results	152
6.4.1	Case study	152
6.4.2	Algorithm evaluation	154
6.4.3	Performance compared to locally optimized benchmark	156
6.4.4	Crowd-shipper capacity	157
6.4.5	Parcel locker capacity	159
6.4.6	Sensitivity for cost parameters	160
6.5	Summary	161
7	Conclusions and future research	163
7.1	Main findings	163
7.2	Future research	168

Appendices

A	Proofs for theorems in Chapter 2	173
B	Proofs for theorems in Chapter 3	175
C	Cost formulations and proofs for theorems in Chapter 4	181
C.1	Cost formulations	181
C.2	Theorems and proofs	184
C.2.1	Optimality propositions	184
C.2.2	Departure time choice proofs	185
C.2.3	Optimal departure time with stochastic travel times	187
	References	191
	Curriculum vitae	203

1

Introduction

This chapter is based on the articles:

- P. Stokkink and N. Geroliminis (2021). “Predictive user-based relocation through incentives in one-way car-sharing systems”. In: *Transportation Research Part B: Methodological* 149, pp. 230–249
- A. de Palma, P. Stokkink, and N. Geroliminis (2022). “Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users”. In: *Transportation Research Part B: Methodological* 155, pp. 479–498
- P. Stokkink and N. Geroliminis (2023). “A continuum approximation approach to the depot location problem in a crowd-shipping system”. In: *Transportation Research Part E: Logistics and Transportation Review* 176, p. 103207
- P. Stokkink, A. de Palma, and N. Geroliminis (2023). “Multi-modal ride-matching with transfers and travel-time uncertainty”. Available in: THEMA Working Papers (Currently under review in a scientific journal)
- P. Stokkink, J.-F. Cordeau, and N. Geroliminis (2023). “A column and row generation approach to the crowd-shipping problem with transfers”. Available in: HEC Working Papers (Currently under review in a scientific journal)

1.1 Motivation and background

The new era of shared economy has raised our expectations to make mobility more sustainable through better utilization of existing resources and capacity. Traffic congestion is one of the biggest problems in passenger transport. As a result of a large number of commuters, the preference for private car usage leading to low vehicle occupancy, urban-centered lifestyle, and the high similarity of work schedules from a temporal perspective, the morning and evening commutes on (urban) road networks are often heavily congested. Traffic congestion causes inconvenience on a private level, with large delays, tardiness, and idle time spent on congested roads. Thereby, congestion has many negative social effects, such as CO_2 emissions, fuel and energy waste, and degradation of air quality.

In urban areas, the effect of road congestion is even more prevalent. Because of the dense urban networks with one-way streets, congestion easily propagates through the network. This is further amplified by roadblocks caused by road maintenance or on-street parking of delivery vehicles. Traditional delivery vehicles are often highly polluting but remain commonly used up to now.

Although the large number of commuters has many negative private and social effects, in this thesis we describe multiple ways to leverage existing commuters to improve last-mile mobility, transport, and logistic systems. By stimulating multi-purpose, where commuters are fulfilling a second purpose during their commute, we try to improve vehicle occupancy and improve the efficiency and sustainability of transport and logistics systems. In this thesis, we consider user-based vehicle relocation, ride-sharing, and crowd-shipping as multi-purpose trips, as illustrated in Figure 1.1. Car-Sharing Systems (CSSs) have become an interesting alternative to private car ownership, due to their benefits in terms of mobility and sustainability. Benefits for the individual users include reduced transportation cost and mobility enhancement, while society as a whole benefits from reduced congestion and emissions (see for example Martin and Shaheen, 2011 and Baptista, Melo, and Rolim, 2014). Over the last years, the number of car-sharing users has increased rapidly. A recent study by Frost & Sullivan, 2016 has shown that the increase in the number of users of CSSs is likely to continue over the coming years. One of the main problems in one-way car-sharing systems is the relocation of cars which is needed because of the asymmetry in demand, leading to imbalances. Staff-based vehicle relocations are a potential solution to this problem, but lead to new vehicle and personnel movement.

In line with the vision of this thesis, we explore the potential of user-based vehicle relocation in Chapter 2. The main benefit of this is that rather than using staff members, users with pre-existing itineraries are incentivized to slightly change their



Figure 1.1: Multi-purpose trips

trip and thereby contribute to the relocation of vehicles. In this way, the imbalances of the car-sharing system can be resolved in a more sustainable way.

Commuting by private car remains one of the most used modes of transport. With average car occupancy in the US being only 1.5 in 2019 (U.S. Department of Energy (DOE), 2022), it is clear that the majority of commuters travels alone. Given the empty space in private vehicles, this can be seen as a waste of resources. Ride-sharing is a good potential solution to this problem, where individuals maintain the flexibility of private transportation, while car occupancy is increased. In ride-sharing, commuters with pre-existing travel itineraries share a ride for a part of their journey.

For ride-sharing to be a competitive mode of transport, there are some cost components that need to be considered. Clearly, fuel and parking costs can be shared which reduces the costs of the driver. However, either the driver, the rider, or both need to make a detour to reach each other, which increases their inconvenience. Thereby, if the desired arrival times of the driver and rider do not coincide, this will lead to schedule delay penalties. In this thesis, we consider all these cost components and obtain a matching that minimizes these costs. A matched driver and rider need to have highly similar itineraries (i.e., routes and desired arrival times) for a match to be low-cost and therefore competitive with private transport. This is a highly restrictive requirement which has hindered the widespread adoption of ride-sharing as a mode of transport.

We evaluate two scenarios that can help to relax these requirements. In Chapter 3,

we evaluate the effect of congestion during the morning commute on ride-sharing potential. In addition to the economies of scale during the morning commute (there are more potential drivers and riders), congestion leads to more flexibility in the time schedules of commuters. In equilibrium, commuters may have an interval of departure/arrival times for which their costs are roughly similar, as opposed to a specific desired arrival time that minimizes their cost in the non-congested case. This, therefore, makes the matching of drivers and riders more flexible with respect to time. In Chapter 4 we evaluate the effect of transfers in a ride-sharing system. By allowing riders to transfer from one driver to another, or even transfer between different modes of transport, a match is only required to be partially similar. This can also increase the potential of ride-sharing. According to Li et al., 2007, family carpooling makes up nearly 75% of all carpools. Since families share an origin but not necessarily a destination during their morning commute, transfers can drastically increase the potential of family carpooling.

Rather than sharing the commute with another person, individuals can also share their commute with a small parcel. Such a situation is referred to as crowd-shipping, where the last-mile delivery of small parcels is outsourced to individual commuters who can pick up and deliver a parcel on their pre-existing route. Although the similarities between crowd-shipping and ride-sharing are apparent, there are some fundamental differences. The static nature of a parcel makes it so that all the delivery effort needs to come from the driver (also referred to as a crowd-shipper). However, parcels do not have the same strict time schedules as people and do not perceive inconvenience in the same way. This makes storing a parcel at an intermediate point a reasonable option. Based on this, we consider a depot-based crowd-shipping system in Chapter 5, where we develop an efficient algorithm to find the optimal depot locations. Furthermore, in Chapter 6, we consider transfers of parcels between various crowd-shippers. Compared to the developed methodology in Chapter 4, due to the flexibility of parcels with respect to transfers and inconvenience, the size of the matching problem is significantly larger. Therefore, we develop a column-and-row generation approach to find the optimal matching and routes for parcels.

Specifically, we consider bike-based crowd-shipping systems. These systems are more sustainable than car-based crowd-shipping systems. Especially in urban areas, bikes can more easily move through the network, stop without blocking a street, and enter low-emission or car-free zones. Despite this, we still focus on bikers with pre-defined itineraries. By leveraging existing bikers, this concept aligns with the multi-purpose vision of this thesis.

This thesis focuses on leveraging existing vehicle flows for last-mile mobility, transport, and logistics systems by stimulating multi-purpose trips. Through

user-based vehicle relocation, ride-sharing, and crowd-shipping, we aim to improve the efficiency and sustainability of these systems, decrease car ownership, increase car occupancy, and reduce congestion. In the remainder of this section, we summarize the existing literature for these three types of mobility and logistics systems and state the specific objectives and contributions of this thesis.

1.1.1 State-of-the-art on relocation in car-sharing systems

Over the last decades, car-sharing systems, as well as other vehicle-sharing systems such as bike- and ebike-sharing, have received increasingly more attention. In many large cities, vehicle-sharing systems emerge for various modes of transportation. The systems can be classified as either free-floating or station-based. The first refers to the case where vehicles can be dropped off at any location where parking is permitted within the specified operating area. This type of system has been considered by, among others, Weikl and Bogenberger, 2013 and Herrmann, Schulte, and Voß, 2014. The environmental effects of such a system are significant, as described by Firnkorn and Müller, 2011. However, they are often also harder to handle. Li, Liao, Timmermans, Huang, and Zhou, 2018 incorporate free-floating car-sharing in a dynamic user equilibrium model and thereby illustrate the supply-demand interaction for shared cars. Station-based CSSs on the other hand, require vehicles to be picked up and dropped off at a limited number of stations. A major advantage of this type of system is that electric vehicles can be charged at these stations. As described by Li, Ma, Cui, Ghiasi, and Zhou, 2016, this innovative mobility service has benefits in terms of sustainability and the environment.

Station-based CSSs can be either one-way or two-way systems. Two-way systems require the customers to drop-off the vehicle at the same station as where they picked it up. One-way systems allow the customer to drop their vehicle off at any other station of their choice. Due to the increased level of flexibility of these systems, they are commonly viewed as a more attractive alternative for customers compared to two-way systems. As described by Boyacı, Zografos, and Geroliminis, 2015, the attractiveness of a CSS is not only determined by its flexibility, but also by its level of service. The level of service consists of two important factors: accessibility and availability. Accessibility refers to the distance of the origin and destination of a customer to the available vehicle. Availability refers to the availability of a vehicle at the right time and the right place.

System performance (i.e. profit and service level) is optimized on a strategic, tactical and operational level. On a strategic and tactical level, long and midterm decisions are made. These decisions include the locations of stations (Kumar and Bierlaire, 2012, Brandstätter, Kahr, and Leitner, 2017), the size of the fleet of vehicles (George and Xia, 2011, Nair and Miller-Hooks, 2011) and the number of

staff members Kek, Cheu, Meng, and Fung, 2009. In this paper, we focus on the operational decisions, which regard the redistribution of vehicles in the network to guarantee a minimum service level.

In one-way vehicle-sharing systems, the availability of vehicles is often problematic. Uncertain and asymmetric demand are the main causes of the existence of balancing problems. At a location where the demand for vehicles is high, the number of available vehicles declines rapidly. On the other hand, at a location where the supply for vehicles is high, the number of available parking places declines. Due to the limited availability of both vehicles and parking spaces the service level of CSSs decreases. Non-reserved vehicles should be relocated either to create available vehicles for stations with high density of origins or to create available slots for stations with high density of destinations. Nevertheless, due to limited resources for relocation and the fact that vehicles are unavailable during this movement, an optimization framework should be integrated.

To solve this balancing problem, vehicles should be relocated. For an extensive review of relocation strategies in one-way car-sharing systems, the reader is referred to Illgen and Höck, 2019. In the literature, both static and dynamic relocation policies are considered. The static relocation policy assumes that no demand occurs during the relocation of vehicles, suggesting the vehicles are relocated at night. Chemla, Meunier, and Calvo, 2013 consider a capacitated pickup and delivery problem to describe the static bike relocation problem. Static relocation problems are easier to solve as they are less prone to uncertainty. However, they are also less effective as temporary imbalances during the day can not be resolved.

A dynamic relocation policy considers the relocation of vehicles during the day. This is discussed by among others Caggiani and Ottomanelli, 2013 and Boyacı, Zografos, and Geroliminis, 2015. Dynamic relocation policies are more effective as vehicles can be relocated throughout the day. However, as customers arrive dynamically, uncertainty forms a major burden. Due to this uncertainty, most relocation policies rely on simple benchmarks such as a minimum number of vehicles at each station. As these policies do not incorporate expected future demand, they are classified as non-predictive. Predictive methods incorporate expected future demand and thereby expected future states of the system. Such a predictive relocation policy was developed by Repoux, Kaspi, Boyacı, and Geroliminis, 2019. They use a Markovian model to describe the state of the system and optimize their staff-based relocation policy based on this.

In practice, vehicle relocations are mostly performed by staff members. Staff

members pick up vehicles at over-saturated locations and deliver them to under-saturated locations. In bike-sharing systems, a truck can be used to relocate multiple bikes at the same time by a single staff member (Caggiani and Ottomanelli, 2013). However, in car-sharing systems this procedure is less efficient as only a single car can be moved at the same time by one staff member. User-based relocation offers a more sustainable and less costly alternative to staff-based relocation. User-based relocation refers to the case where users are stimulated to relocate the vehicles themselves, thereby contributing to a more balanced system. An example of such a method is paid relocation, as described by Schulte and Voß, 2015, where users are paid free minutes or other bonuses. Jorge, Molnar, and Almeida Correia, 2015 use dynamic trip pricing to reduce imbalances. They offer higher prices to trips that increase imbalances and lower prices to those trips that improve the state of the system.

The most common type of user-based relocation is customer incentivization. In this case, customers are stimulated to change their pickup or delivery location by offering them a discount. By doing this, a less favorable location in terms of access time may be chosen by the customer, which aims to reduce the balancing problem. Correia, Jorge, and Antunes, 2014 investigate that if customers are more flexible in their choice for pickup and delivery locations, a significant increase in profit can be obtained by incentivizing customers. Angelopoulos, Gavalas, Konstantopoulos, Kyriadis, and Pantziou, 2016 provide discounts to customers if they contribute to the balancing process. Their decision is based on priorities that are assigned based on the capacity and occupancy of the stations. Similarly, Brendel, Brauer, and Hildebrandt, 2016 assume that the price of a ride is a function of the extra time that is required to perform a relocation. They assume the same value of time applies to all customers, thereby disregarding customer heterogeneity. Most of the literature considers policies where incentivization decisions are made based on threshold values (Clemente, Fanti, Iacobellis, Nolich, and Ukovich, 2017) or problematic scenarios at stations such as being completely full or completely empty (Singla et al., 2015). Di Febbraro, Sacco, and Saeednia, 2018 determines the best incentive stations and discount in a sequential manner. They determine the best station based on the relative demand for vehicles at all stations and the best discount value is determined to maximize the systems' profit. These approaches can be classified as non-predictive, in the sense that they do not incorporate expected future demand loss caused by insufficient vehicles or parking spaces. Future demand is integrated by Pfrommer, Warrington, Schildbach, and Morari, 2014, who incorporate the difference between supply and demand rates of bikes in their decisions. They use truck routing and dynamic incentives to relocate bikes in a bike-sharing system.

In on-demand transportation systems, incentives or dynamic (surge) pricing are

often used to balance demand and supply. For example, Yang, Shao, Wang, and Ye, 2020 design a reward scheme integrated with surge pricing for the ride-sourcing market. Similarly, Zha, Yin, and Du, 2018 propose equilibrium models for supply in ride-sourcing and investigate the effect of surge pricing. Xiong et al., 2019 design an incentive scheme to create energy efficient mobility systems using personalized traveler information. Ma, Ban, and Szeto, 2017 propose an emission pricing model for dynamic traffic networks. They determine the optimal first-best emission pricing by solving an optimal control problem.

1.1.2 State-of-the-art on ride-sharing

In ride-sharing people share a ride for a part of their journey, which reduces the time a driver is traveling with a partially empty vehicle. Teal, 1987 provides an early definition of carpooling and distinguishes between different types of carpoolers. Shaheen and Cohen, 2019 give an overview of the various shared-ride services that exist in the modern day. The two most important ones are ride-sharing (commuters that have a predefined trip purpose share a ride) and ride-hailing (also known as ride-sourcing, which is more similar to a taxi service). Whereas studies have shown that ride-hailing generally leads to an increase in congestion (Beojone and Geroliminis, 2021; Schaller, 2021), ride-sharing generally reduces congestion by increasing vehicle occupancy and thereby reducing the number of vehicles on the road (Caulfield, 2009; Gurumurthy, Kockelman, and Simoni, 2019; Palma, Javaudin, Stokkink, and Tarpin-Pitre, 2022). Ride-sharing may lead to environmental and societal benefits but brings forth many optimization challenges.

For a review of the optimization challenges in ride-sharing, the reader is referred to Agatz, Erera, Savelsbergh, and Wang, 2012. One of the most important optimization problems in ride-sharing is the matching of drivers and passengers. Matching algorithms can aim to find system optimal matching, as described by Özkan and Ward, 2020, or a stable matching where no individual can improve their match as described by Wang, Agatz, and Erera, 2018. Various extensions to the traditional ride-sharing framework have been proposed. For example, Santi et al., 2014 consider the sharing of taxi services to reduce its negative effect at the cost of increased inconvenience perceived by the passengers. Alonso-Mora, Samaranayake, Wallar, Frazzoli, and Rus, 2017 consider high-capacity ride-sharing with a dynamic trip-vehicle assignment and also display the trade-off between passenger inconvenience and negative externalities of commuting. The standard matching models are very much used in labor economics (Zenou, 2009) and in the economics of the family (Browning, Chiappori, and Weiss, 2014). In urban economics, matching is also used to justify the micro-foundations of agglomeration effects or wide economic effects (Duranton and Puga, 2004).

The matching problem can be extended to include transfers between various vehicles. Herbawi and Weber, 2012 model the ride-matching problem with transfers and time windows and use a genetic algorithm to solve this problem. Masoud and Jayakrishnan, 2017 consider multi-hop ride-matching where a driver can carry multiple riders and riders can join multiple drivers. Huang, Bucher, Kissling, Weibel, and Raubal, 2018 include carpooling in the trip planning of commuters next to public transport. Commuters are allowed to transfer between drivers or between modes. Lu, Liu, Wang, Zhou, and Hu, 2020 consider ride-sharing with transfers in short-notice evacuations such as during natural or man-made disasters.

In the literature, carpooling has been modeled both as a competitor of public transport (Li, Li, and Zhang, 2021) or as a complement to public transport (Kong, Zhang, and Zhao, 2020). The former considers public transport as an alternative mode of transport (Palma, Javaudin, Stokkink, and Tarpin-Pitre, 2022; Palma, Stokkink, and Geroliminis, 2022). In this case, carpooling can reduce public transport users and therefore has negative societal effects. The latter considers public transport as a feeder to carpooling or carpooling as a feeder to public transport (Kumar and Khani, 2021; Ma, Rasulkhani, Chow, and Klein, 2019; Masoud, Nam, Yu, and Jayakrishnan, 2017). In that case, the two services may help to improve each other and form a competitive alternative against private car usage. In this work, we consider both alternatives simultaneously to properly consider the interaction of the two transport modes when riders are allowed to make transfers.

Ride-sharing may be influenced by sudden changes in travel time. Previous studies have focused on a robust optimization approach to ride-sharing with travel time uncertainty (Li, Gao, Wang, Huang, and Nie, 2022; Li and Chung, 2020). Long, Tan, Szeto, and Li, 2018 consider a bi-objective ride-sharing-matching model under travel-time uncertainty. They consider delay and schedule delay penalties that may change according to this uncertainty. Our work is most similar to the latter work, but as we allow for transfers, travel time uncertainty does not only affect (schedule) delay penalties but may also affect the feasibility of some matches. We consider the schedule delay structure for commuters as defined previously by Vickrey, 1963 and Small, 1982.

1.1.3 State-of-the-art on crowd-shipping

The last-mile delivery of parcels is a well-studied topic in the optimization literature. Traditionally, goods are delivered by using delivery vans. In this case, the problem can be formulated as a Pickup-and-Delivery Problem (PDP) (Savelsbergh and Sol, 1995) or a Vehicle Routing Problem (VRP) (Toth and Vigo, 2002). These problems have been extended to include various problem-specific aspects such as time-windows (Dumas, Desrosiers, and Soumis, 1991; Ropke and Cordeau, 2009) or

uncertainty (Fabri and Recht, 2006).

Due to the increase in online shopping, a large number of traditional vans is needed to serve all demand. According to Iwan, Kijewska, and Lemke, 2016, delivery vans for last-mile delivery are one of the main causes of congestion in urban areas. As a consequence, many companies are looking for more sustainable options to replace the aforementioned traditional delivery methods. Iwan, Kijewska, and Lemke, 2016 analyze the use of parcel lockers where customers can pick up and send small parcels. Results from a pilot survey in Poland indicate that the use of these lockers can potentially reduce the environmental impact of last-mile delivery. Another alternative is drone delivery (Agatz, Bouman, and Schmidt, 2018; Karak and Abdelghany, 2019), for which it has been shown that combining truck delivery with a drone can significantly reduce transportation costs. Thereby, drones cause less congestion compared to delivery vans. Akeb, Moncef, and Durand, 2018 propose a model that relies on the interaction of a network of neighbors to enhance parcel delivery in urban areas.

Another promising alternative to traditional last-mile delivery methods is crowd-shipping. In a crowd-shipping system, the last-mile delivery of small parcels is (partially) outsourced to individual commuters that can deliver the parcel on their pre-existing route. Various empirical studies have investigated the potential and determinants of crowd-shipping (Ermagun and Stathopoulos, 2018; Le and Ukkusuri, 2019b; Punel, Ermagun, and Stathopoulos, 2019). These studies have shown the potential demand for crowd-shipping and the concerns of potential users. Thereby, they highlight the importance of the availability of occasional couriers. Potential crowd-shippers have a pre-existing itinerary (origin, destination, and approximate departure and arrival times) and trip purpose (for example, a work commute or a leisure trip). Therefore, only parcels that do not create significant inconvenience can be assigned to the crowd-shipper that has to be compensated for the inconvenience through some (monetary) incentive.

Recently, substantial research has been done on the operational problems that arise in crowd-shipping systems. For a review of recent academic research as well as recent practice, the reader is referred to Le, Stathopoulos, Van Woensel, and Ukkusuri, 2019. Pourrahmani and Jaller, 2021 give an overview of the operational challenges and research opportunities that exist in this field. One of these operational challenges is the matching of parcels to crowd-shippers which has been studied by, among others, Li, Krushinsky, Reijers, and Van Woensel, 2014 and Soto Setzke et al., 2017. Another important operational problem is pickup and delivery routing. Clearly, these problems are intertwined and therefore are often tackled jointly. Archetti, Savelsbergh, and Speranza, 2016 model the static routing problem as a

Vehicle Routing Problem with Occasional Drivers (VRPOD). It is assumed that an occasional driver is willing to make a delivery if the extra distance traveled to make the delivery is less than a pre-specified portion of the total distance traveled. Li, Krushinsky, Reijers, and Van Woensel, 2014 consider a crowd-shipping scenario where people and parcels share a taxi. They model the problem as a Share-a-Ride Problem (SARP), which is an extension of the Dial-a-Ride Problem (DARP). Dahle, Andersson, and Christiansen, 2017 consider a two-stage stochastic program to model the VRP with dynamic occasional drivers. The first-stage decision models the route of the traditional delivery truck. After the occasional drivers make themselves known in the second stage, they are assigned to parcels and the truck route can be changed. Arslan, Agatz, Kroon, and Zuidwijk, 2019 model the problem as a dynamic PDP. Their heuristic assigns crowd-shipping tasks to occasional (ad-hoc) drivers dynamically. Cohn, Root, Wang, and Mohr, 2007 integrate matching and routing decisions for carriers of small packages. Yildiz and Savelsbergh, 2019 introduce the service and capacity planning problem. With their model, they aim to answer questions that arise in a crowd-shipping system, both on strategic and operational levels.

As the availability of potential crowd-shippers is a key determinant of the performance of a crowd-shipping system, parcels may be stored at intermediate depot locations (or transshipment points) such that they are easily reachable by potential suppliers. Wang, Zhang, Liu, Shen, and Lee, 2016 consider “pop-stations” distributed around the city where crowd-shippers can perform pickups. For a fixed set of transshipment points, they optimize the utilization of crowd-shippers for last-mile delivery. Raviv and Tenzer, 2018 and Macrina, Pugliese, Guerriero, and Laporte, 2020 consider a crowd-shipping system where crowd-shippers can pick up parcels either from the depot or from transshipment points. Their results show the economic benefits of such transshipment nodes. Similarly, Yıldız, 2021a also considers transshipment points but uses a dynamic programming algorithm to solve their problem. Contrary to the fixed transshipment points in the previous works, Mousavi, Bodur, and Roorda, 2022 consider mobile depots. They do not consider the routing of vehicles, but they determine the optimal location of these mobile depots under uncertainty in supply.

The literature on crowd-shipping with transfers can be roughly divided into two types of transshipments. On the one hand, there are transfers between crowd-shippers and another mode of transport, usually traditional delivery vehicles (Macrina, Pugliese, Guerriero, and Laporte, 2020). Such transfers are commonly modeled as two-echelon systems (Laporte and Nobert, 1988). Kafle, Zou, and Lin, 2017 consider crowd-shippers performing first-leg pickups or last-leg deliveries, with relays to trucks performing the middle leg. Various alternatives of the two-echelon delivery

system with crowd-shippers have been introduced, such as mobile satellites (Lan, Liu, Ng, Gui, and Lai, 2022), parcel lockers (Enthoven, Jargalsaikhan, Roodbergen, Broek, and Schrottenboer, 2020; Santos, Viana, and Pedroso, 2022) and delivery options (Vincent, Jodiawan, and Redi, 2022). Others have considered two-echelon systems with transfers to mobile depots (Mousavi, Bodur, and Roorda, 2022) and public transport (Kızıl and Yıldız, 2022) rather than a traditional delivery vehicle.

On the other hand, there are transfers among the crowd-shippers themselves. This can again be divided into two groups of studies. One with transfers taking place at dedicated transfer locations with for example parcel lockers (Raviv and Tenzer, 2018) and one with time-synchronized transfers, where parcels are transferred directly from one crowd-shipper to another and cannot be left unattended (Chen, Mes, and Schutten, 2018). The latter is highly similar to what is classified by Agatz, Erera, Savelsbergh, and Wang, 2012 as multi-hop ride-sharing. Multi-hop ride-sharing has received considerably more attention (Chen et al., 2019; Drews and Luxen, 2013; Herbawi and Weber, 2011; Lu, Liu, Wang, Zhou, and Hu, 2020; Masoud and Jayakrishnan, 2017). We also note the similarity with public transport modeling, where passengers can make stops and transfers when traveling through a public transport network (Spiess and Florian, 1989). The most important difference between multi-hop ride-sharing and multi-stage crowd-shipping is the fact that passengers incur psychological costs when making detours and transfers and when they are waiting at transfer points. Parcels, on the other hand, are more flexible and can make large detours with various transfers as long as they arrive on time.

Chen, Mes, and Schutten, 2018 allow for transfers between crowd-shippers but require time synchronization such that parcels are directly passed on from one to another crowd-shipper. In their approach, a parcel cannot be left unattended. Sampaio, Savelsbergh, Veelenturf, and Van Woensel, 2020 consider a crowd-shipping system with a single transfer at a dedicated transfer point, where parcels can be stored temporarily. As their crowd-shippers do not have predetermined paths, their problem is similar to a pickup and delivery problem with transfers (Mitrović-Minić and Laporte, 2006; Rais, Alvelos, and Carvalho, 2014). The itinerary of crowd-shippers is considered by Voigt and Kuhn, 2022, but they do not consider time windows for crowd-shippers nor parcels. Such a system is clearly less attractive for potential crowd-shippers that wish to deliver a parcel during their commute, where time windows are imposed. Such a system is considered by Yıldız, 2021a, who develop a dynamic programming approach to solve their problem. The authors later extend this problem by considering stochasticity in demand (Yıldız, 2021b). Their crowd-shippers are inflexible and do not deviate from their routes. As a result, crowd-shippers are paid a fixed compensation. Raviv and Tenzer, 2018 offer compensations for stopping and handling. In their work, they assume Poisson

arrivals of occasional couriers, that have a predetermined sequence of transfer points that they will visit. Based on this assumption, they use a stochastic dynamic programming algorithm to find an optimal policy. Nieto-Isaza, Fontaine, and Minner, 2022 take a strategic perspective and focus on finding the optimal locations for mini-depots that function as transshipment points. DiPugliaPugliese, Guerriero, Macrina, and Scalzo, 2021 consider transfers between two types of crowd-shippers: long-distance crowd-shippers and short-distance crowd-shippers in an urban area. Due to this classification, they can more easily model transfers.

We also consider the strategic planning problem of network design. Specifically, we focus on finding the optimal depot locations. This is closely related to the Facility Location Problem (FLP) (Cornuéjols, Nemhauser, and Wolsey, 1983), where optimal locations of facilities are chosen in a network. This approach has been commonly used to determine the location of depots in freight transportation problems (Fernandes et al., 2014; Gendron, Khuong, and Semet, 2016). We also note the similarities with hub location problems in passenger transportation. There, hubs function as locations where passengers can switch between mobility modes in a multi-modal shared mobility system (Blad, Almeida Correia, Nes, and Annema, 2022) or between public transport modes (Yatskiv and Budilovich, 2017). A depot for freight transportation is fundamentally different from a hub for passenger transportation. Whereas freight can be kept at a depot for a long time before a pickup, passengers are sensitive to time and desire rapid transfers. A comprehensive review of various solution algorithms for different variants of the hub location problem is presented by Wandelt, Dai, Zhang, and Sun, 2022.

1.2 Thesis objectives

The overarching objective of this thesis is to improve the sustainability and efficiency of last-mile logistics and (on-demand) transport systems by leveraging existing vehicle flows. Congestion is one of the main problems in transport networks, which is mainly caused by the large number of commuters. The majority of these commuters are traveling alone by car, leading to a lot of unused space. The large number of commuters combined with the inefficient vehicle occupancy provides room for improvement. In this thesis, we explore the possibility to leverage existing vehicle and user flows in car-sharing, ride-sharing, and crowd-shipping systems. By doing this, we aim to improve the operational performance of these systems, increase vehicle occupancy and explore the effect of multiple-purpose trips. Specifically, the following objectives are set, organized in chapters following the structure of the thesis:

- **Chapter 2: Predictive user-based vehicle relocation through incentives in one-way car-sharing systems.**

Car-sharing systems are an attractive alternative to private vehicles due to their benefits in terms of mobility and sustainability. However, the distribution of vehicles throughout the network in one-way systems is disturbed due to asymmetry and stochasticity in demand. As a consequence, vehicles need to be relocated to maintain an adequate service level. Staff-based relocations impute new vehicle flows into the network, whereas user-based vehicle relocation aims to incentivize users with existing travel itineraries to contribute to (implicit) vehicle relocations. The aim of this chapter is to develop a user-based vehicle relocation approach through the incentivization of customers based on a predictive model. We explore the potential of predictive user-based vehicle relocation in comparison to staff-based relocation.

- **Chapter 3: Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users.**

The aim of this chapter is to investigate the effect of dynamic traffic congestion on the matching of heterogeneous users in a carpooling system. The effect of carpooling on congestion is a well-studied problem. The reverse effect, on the other hand, is less studied. We model the two-way causality between carpooling and dynamic congestion with scheduling preferences using a bi-level optimization approach. The first level considers the optimal matching problem and the second level incorporates the equilibrium departure time choices of commuters through a dynamic bottleneck model.

- **Chapter 4: Multi-modal ride-matching with transfers and travel-time uncertainty.**

The aim of this chapter is to develop a framework for multi-modal ride-matching that allows for transfers between drivers and between modes and incorporates travel-time uncertainty. An important operational limitation of direct ride-sharing is that a pairing of drivers and riders needs to be found with matchable itineraries. This means that a driver needs to be able to pick up and drop off the matched rider without deviating too much from their original route. In addition to this, desired arrival times of the rider and the driver need to be similar. Dissimilar matches increase the costs of drivers and riders, such that ride-sharing is no longer competitive with private or public transport. By allowing for transfers, this limitation can be overcome, as the itinerary of drivers and riders only needs to be partially similar. In this chapter, we explore to what extent transfers can increase the potential of ride-sharing.

- **Chapter 5: A continuum approximation approach to the depot location problem in a crowd-shipping system.**

The aim of this chapter is to develop a framework to determine the best depot locations for a crowd-shipping system in a large urban area. The success of a crowd-shipping system heavily relies on the availability of crowd-shippers and the potential to match them to demand requests without large detours. If the pickup locations of parcels are poorly accessible by potential crowd-shippers, few parcels can be delivered by crowd-shippers. We find optimal depot locations based on the interaction between the expected travel patterns of crowd-shippers and the expected spatial distribution of demand for parcels. This problem is especially difficult because of the dependency of lower-level operational decisions and costs on upper-level strategic decisions.

- **Chapter 6: A column and row generation approach to the crowd-shipping problem with transfers**

The aim of this chapter is to bridge the gap between short-distance trips of crowd-shippers in a bike-based crowd-shipping system and the long distance some parcels need to traverse between origin and destination. We propose a framework that allows for transfers of parcels between crowd-shippers and that aims to improve the revenue and service level of a crowd-shipping system. We design a column generation algorithm to solve large-scale realistic instances to optimality. We extend the problem to allow crowd-shippers to carry multiple parcels at the same time and for this extend the algorithm to simultaneously column-and-row generation.

1.3 Thesis contributions

Driven by the stated objectives and based on the developed methodology and obtained results that are elaborated on in the next chapters, this thesis leads to the following contributions. The contributions are listed and elaborated per chapter as follows.

- **Chapter 2: Predictive user-based vehicle relocation through incentives in one-way car-sharing systems.**

In this chapter, we introduce a new predictive user-based relocation strategy that determines the optimal incentive based on both the current state of the system (i.e. distribution of vehicles throughout the network) and expected future demand. In doing so, we aim to anticipate future demand and therefore avoid expected future demand losses. Our user-based relocation strategy builds on unknown customer preferences. These preferences can be approximated by learning from previously offered incentives. The obtained estimates can, in turn, be used to dynamically determine the optimal value of an incentive, as

well as the optimal pickup and delivery location of the vehicle. Our method is adaptive, in the sense that the value of the incentive is adjusted to the value of time of customers, as well as the current and expected future states of the car-sharing system. By offering incentives, the operator stimulates customers to relocate vehicles from over-saturated to under-saturated locations. We evaluate our strategy using an event-based simulation model with synthetic data from a real experiment, which allows us to compare our methods to existing relocation policies. Our results show that incentives can increase the service level of car-sharing systems and decrease the number of staff members needed to achieve this level. Furthermore, our results indicate that incentives are a more profitable and sustainable way of relocating vehicles, compared to staff-based relocations. By using a hybrid operator-user-based relocation strategy, profit, and service levels can be maximized.

- **Chapter 3: Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users.**

In this chapter, we model the two-way causality between carpooling and dynamic congestion with scheduling preferences using a bi-level optimization approach. We formulate the optimal matching problem (first stage) as an integer linear programming problem. Using this formulation, we can determine the matching that minimizes the sum of the cost of the detour and the potential inconvenience costs while all passengers are matched to a driver. Inconvenience cost can be a psychological cost, and/or more specifically some extra schedule delay cost due to the need for coordination of the drivers' and passengers' departure times. Congestion is incorporated through a dynamic bottleneck model (for the second stage). Using an iterative approach, we obtain the optimal matching for a dynamic traffic equilibrium with congestion. By comparing multiple scenarios and performing extensive sensitivity analysis, we evaluate the effect of congestion on carpooling matching. Specifically, we evaluate the effect of congestion on departure time choices and with that the effect on matching decisions. Thereby, we inspect the tendency of commuters for early departures and the tendency of carpoolers to avoid the peak of congestion.

- **Chapter 4: Multi-modal ride-matching with transfers and travel-time uncertainty.**

In this chapter, we develop a framework for multi-modal transport of riders that considers public transport, solo driving, and ride-sharing. The framework allows for transfers between modes and between drivers. We formulate the multi-modal ride-matching problems with multiple transfer hubs as a path-based integer programming problem. We model the same problem with uncertain travel times as a two-stage stochastic programming problem. We

evaluate the effect of transfers on mode-choice, generalized travel cost, and vehicle hours traveled. Furthermore, we investigate the effect of limited information on travel times on the performance of the system and specifically on the appeal of transfers.

- **Chapter 5: A continuum approximation approach to the depot location problem in a crowd-shipping system.**

In this chapter, we develop a framework to determine the best depot locations for a crowd-shipping system in a large urban area. This problem is especially difficult because of the dependency on lower-level operational decisions and costs on upper-level strategic decisions. To track these interactions, we solve the lower-level assignment problem of parcels to potential crowd-shippers through a Continuum Approximation (CA) approach, allowing us to determine the lower-level costs efficiently in a short time. These estimates are based on the physical properties of the matching procedure, as well as expectations of the set of parcels and the set of crowd-shippers, fed by historical data. We develop a large neighborhood search heuristic that exploits the CA estimates to efficiently search a good set of depots that minimizes the operational costs. In addition to this, these estimates are used to design a smart dynamic assignment strategy of parcels to crowd-shippers that outperforms existing strategies. A comparison of our approach to solving a discrete formulation of the problem shows that on small networks the objective obtained by our CA-based approach is slightly better than that of the discrete formulation. In terms of computation time, our CA-based approach is between 200 and 1000 times faster. We highlight the importance of the interaction between supply and demand patterns, rather than solely considering geographically central locations. The results show that using CA-based strategies in all three layers of decision-making can improve overall performance by 15% compared to non-predictive strategies.

- **Chapter 6: A column and row generation approach to the crowd-shipping problem with transfers**

In this chapter, we propose a general framework that allows the incorporation of both time-synchronized transfers as well as transfers with intermediate storage at transfer points for a crowd-shipping system. In addition to this, we consider the original itinerary of crowd-shippers including their departure times, but we consider some flexibility in their routing decisions, making crowd-shipping accessible to daily commuters. On top of this, we consider a detailed compensation scheme for crowd-shippers, which includes rewards for stops, detours, and the inconvenience of carrying a parcel for a longer distance. Furthermore, we consider heterogeneous crowd-shippers and parcels. We propose a column-generation approach to solve our problem. This method

is highly scalable and allows solving larger instances than those previously considered in the literature for similar problems. Our results are evaluated on a realistic large-scale case study of the city of Washington DC.

Other contributions

In the period of the work towards the dissertation, significant contributions were made to two works that were developed alongside the main body and contributions of this thesis. The first was an extension of the work described in Chapter 3 towards a large-scale network application. The second was the development of an incentive-based electric vehicle charging scheme for managing bottleneck congestion, which took aspects from the works described in Chapter 2 and 3. A summary of these works is provided in the following.

Despite the clear benefits of ride-sharing in terms of reduced congestion, ride-sharing is not yet widely accepted. We propose a specific ride-sharing variant, where drivers are completely inflexible. This variant can form a competitive alternative against private transportation, due to the limited efforts that need to be made by drivers. However, due to this inflexibility, matching of drivers and riders can be substantially more complicated, compared to the situation where drivers can deviate. In this work, we propose a four-step procedure to identify the effect of such a ride-sharing scheme. We use a dynamic mesoscopic traffic simulator that computes departure-time choices and route choices for each commuter. The optimal matching of potential drivers and riders is obtained outside the simulation framework through an exact formulation of the problem. We evaluate the potential of this ride-sharing scheme on a real network of the Paris metropolitan area for the morning commute. We show that even with inflexible drivers and when only a small share of the population is willing to participate in the ride-sharing scheme, ride-sharing can alleviate congestion. This study is published as a stand-alone article as:

- A. de Palma, L. Javaudin, P. Stokkink, and L. Tarpin-Pitre (2022). “Ride-sharing with inflexible drivers in the Paris metropolitan area”. In: *Transportation*, pp. 1–24

We propose an incentive-based traffic demand management policy to alleviate traffic congestion on a road stretch that creates a bottleneck for commuters. The incentive targets electric vehicle owners by proposing a discount on the energy price they use to charge their vehicles if they are flexible in their departure time. We show that, with a sufficient monetary budget, it is possible to completely eliminate traffic congestion and we compute the optimal discount. We also analyze the case of a limited budget when the congestion cannot be completely eliminated. We compute analytically the policy minimizing the congestion and estimate the level of inefficiency for different budgets. We corroborate our theoretical findings with numerical simulations that

allow us to highlight the power of the proposed method in providing practical advice for the design of policies. This study is published as a stand-alone article as:

- C. Cenedese, P. Stokkink, N. Geroliminis, and J. Lygeros (2022). “Incentive-based electric vehicle charging for managing bottleneck congestion”. In: *European Journal of Control* 68, p. 100697

1.4 Thesis structure

The thesis is organized in 7 chapters. The structure of each chapter is described and the publications of parts of each chapter in scientific journals are listed below. Chapters 2, 3, 4, 5, and 6 are standalone articles published or under review in scientific journals. These chapters are divided into two parts, based on the field of research. Chapters 2, 3, and 4 are included in Part I: First and last-mile mobility systems. Chapters 5 and 6 are included in Part II: Last-mile logistics systems. The literature review has been removed from each chapter and is presented as a whole for the entire thesis in Chapter 1. Each chapter, being a stand-alone article, has its own notation which is introduced at the start of the chapter. Therefore, the same symbol can be used to represent different variables or parameters in different chapters.

Chapter 2 presents the user-based relocation policy to solve the vehicle relocation problem in oneway car-sharing systems. It includes the nested optimization problem to determine the optimal incentive and the Markovian model to describe the state of the system. This work has been presented at the *24th International Symposium on Transportation and Traffic Theory (ISTTT24)* and the *2020 Forum on Integrated and Sustainable Transportation System (ISTS)*. Chapter 2 is a stand-alone article published as:

- P. Stokkink and N. Geroliminis (2021). “Predictive user-based relocation through incentives in one-way car-sharing systems”. In: *Transportation Research Part B: Methodological* 149, pp. 230–249

Chapter 3 presents the bi-level optimization approach for the carpooling matching of heterogeneous users with scheduling preferences and dynamic congestion. Theoretical results are presented for the uncongested model and simulation results are given for the congested model. Parts of this research have been presented at the *International Transportation Economics Association (ITEA) Annual Conference in 2021* and the *Transportation Research Board (TRB) 101st Annual Meeting*. Chapter 3 is a stand-alone article published as:

- A. de Palma, P. Stokkink, and N. Geroliminis (2022). “Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users”. In: *Transportation Research Part B: Methodological* 155, pp. 479–498

Chapter 4 presents the framework for the multi-modal ride-matching problem with transfers and travel-time uncertainty. The deterministic variant is modeled as a path-based integer programming problem and the stochastic variant is given as a two-stage stochastic programming problem. Part of this research has been presented at the *11th Symposium of the European Association for Research in Transportation (hEART)*. Chapter 4 is a stand-alone article submitted for publication in *Omega: The International Journal of Management Science*, currently under review:

- P. Stokkink, A. de Palma, and N. Geroliminis (2023). “Multi-modal ride-matching with transfers and travel-time uncertainty”. Available in: *THEMA Working Papers*

Chapter 5 presents the continuum approximation approach for the parcel-depot and parcel-crowd-shipper assignment and the large neighborhood search heuristic for finding the optimal depot locations. Parts of this work have been presented at the *Transportation Research Board (TRB) 101st Annual Meeting*, the *Journées de l’Optimisation 2022 (JOPT)*, and the *11th Triennial Symposium on Transportation Analysis (TRISTANXI)*. Chapter 5 is a stand-alone article published as:

- P. Stokkink and N. Geroliminis (2023). “A continuum approximation approach to the depot location problem in a crowd-shipping system”. In: *Transportation Research Part E: Logistics and Transportation Review* 176, p. 103207

Chapter 6 presents the formulation for the crowd-shipping problem with transfers and the column-and-row generation algorithm proposed to solve this problem. Parts of this work have been presented at the *Transportation Research Board (TRB) 102nd Annual Meeting* and the *2nd INFORMS Transportation Science and Logistics (TSL) Society Triennial Conference*. Chapter 6 is a stand-alone article submitted for publication in *Transportation Science*, currently under review:

- P. Stokkink, J.-F. Cordeau, and N. Geroliminis (2023). “A column and row generation approach to the crowd-shipping problem with transfers”. Available in: *HEC Working Papers*

Finally, Chapter 7 summarizes the findings and contributions of the thesis and discusses about possible future research directions.

Part I

First and last-mile mobility systems

2

Predictive user-based relocation through incentives in one-way car-sharing systems

This chapter is based on the article:

- P. Stokkink and N. Geroliminis (2021). “Predictive user-based relocation through incentives in one-way car-sharing systems”. In: *Transportation Research Part B: Methodological* 149, pp. 230–249

2.1 Introduction

Car-sharing is a low-cost and sustainable alternative to private car ownership. Instead of owning a private car that is used occasionally, a shared car can be picked up only when needed. Car-sharing can reduce car ownership and private car trips but also improves mobility for individuals that do not own a private car. This chapter focuses on one of the main operational problems that car-sharing systems encounter: vehicle imbalances. Due to unknown and asymmetric demand, a surplus of vehicles arise in one region of the network, while a deficit of vehicles arises in another region. We propose a predictive user-based relocation scheme through incentives. We formulate the problem of finding the optimal incentives as a bi-level optimization problem. Imbalances are modeled through Markov Chains based on the expected movements of users. Experimental results for a real city are obtained through a discrete event simulation and indicate the advantage of user-based vehicle relocation.

Following the motivation and detailed review of the literature describing the current approaches to vehicle relocation in car-sharing systems, which are given in Sections

1.1 and 1.1.1 of Chapter 1, respectively, this chapter is organized as follows: In Section 2.2 we formally introduce the problem and the modeling framework. The mathematical formulation of the bi-level optimization program is given and the Markov Chains to model the state of the system are described here. The experimental results are given in Section 2.3 for a case study of a car-sharing system in the city of Grenoble, France. The proposed user-based relocation policy is compared to more common staff-based relocation policies. Finally, the chapter is concluded with a summary of the main findings in Section 2.4.

2.2 Problem description and modeling framework

2.2.1 Problem description

The characteristics of the designed system are similar to those considered by Repoux, Kaspi, Boyacı, and Geroliminis, 2019 and occur in many real cities such as Toyota City in Japan and Grenoble in France. We consider a one-way car-sharing system where once users arrive to the system, they select their preferred origin and destination stations. This type of reservation policy is referred to as a complete journey reservation policy. The user is allowed to reserve the vehicle a short time in advance and a parking space is reserved at the destination until the vehicle is returned. We consider that users are possibly offered an alternative and less convenient trip after revealing their preferences, in return for a small discount. Due to a shortage of either vehicles or parking spaces, a user's first-choice trip may be unavailable. In that case, the user can choose to accept the incentive or decline and choose a different mode of transportation. We should keep in mind that the short time for reservations does not allow for proactive relocations based on real information. Nevertheless, a predictive model utilizing historical data and the current state of the system can be proved beneficial compared to standard threshold-based strategies that are among the most established in the state of practice.

We assume that users make reservations using their user ID, which is for example linked to their driving license. For this reason, we can collect user-specific data, which contributes to the learning process. Such a system is commonly used in practice to ensure that only registered people with valid driving licenses can reserve a vehicle.

Besides user-based relocations, we consider that vehicles can be relocated by staff members. Staff members can pick up vehicles at locations with a shortage of parking spaces and deliver them to locations with a shortage of vehicles. We note that using staff members in car-sharing systems is not necessarily efficient, as only one car can be moved by a staff member at a specific time. On the other hand, user-based

relocations are less flexible as users typically do not want to spend too much effort to reach their destination. Therefore, user-based relocations are mainly short-distance relocations. Thus, even if users are willing to change their origin or destination following the recommendations of the system (through some incentives) this cannot guarantee that it can lead to a proper rebalancing of the system.

2.2.2 Relocation policy

Car-sharing operators usually focus on simultaneously maximizing their profit and the level of service they offer to their users, which is reflected in their relocation policy. In our approach, the operator can offer each arriving user an incentive. Upon the arrival of a user, the operator determines i) whether to offer an incentive, ii) what the discount value of the optimal incentive is and iii) between which stations the vehicle should be relocated. Therefore, an optimization problem is solved upon the arrival of every user. In this section, we provide a detailed description of this optimization problem.

We define I as the set of feasible incentives. An incentive is feasible if there are sufficient available vehicles at the pickup location and sufficient available parking spaces at the drop-off station. Sufficiency suggests that at least one vehicle is available at the origin and at least one parking space is available at the destination. Thereby, we limit this set to only contain incentives for which the user can reach the stations within a given time interval (see Section 2.2.5). For every incentive i we define $\Delta_{time}(i)$, the increase in access time the user experiences when accepting the incentive, and $\Delta_{cost}(i)$, the discount value that is offered. The estimated probability that a user accepts an incentive, \hat{P}_{acc} , is based both on $\Delta_{time}(i)$ and $\Delta_{cost}(i)$. The shape of the estimated probability function $\hat{P}_{acc}(\Delta_{time}, \Delta_{cost})$ is described in more detail in Section 2.2.5.

The aim of offering incentives is to relocate the vehicles to omit expected future losses in demand due to vehicle imbalances. We refer to this as the expected omitted demand loss, $ODL(i)$, for the system when the user accepts incentive $i \in I$. Thereby, we define w as the importance of demand loss relative to the cost of incentives. That is, the higher the value of w , the higher the relative importance of demand loss. Given that demand is known in the system operator with a short notice, a predictive framework has to be integrated. The idea is that based on the current state of each station (which is measured in real time), and the historical demand between origins and destinations, the operator can estimate the probability that in a given future window a station will run out of vehicles or slots. If this probability is multiplied by the expected demand for origins (related to available vehicles) or demand for destinations (related to available parking spots) an expected loss can be estimated. We refer to the original pickup and delivery stations as o and d

respectively. The pickup and delivery stations that are chosen as a consequence of the acceptance of the incentive are referred to as o^* and d^* . As depicted in Figure 2.1, the use of one incentive implicitly replaces at most two staff-based relocations.

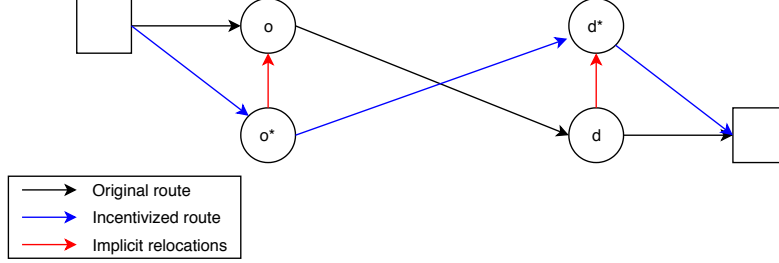


Figure 2.1: Implicit relocations experienced due to incentive

The operator can determine for every possible incentive $i \in I$ what the optimal discount $\Delta_{cost}(i)$ is. For the operator, this decision is based on a trade-off between cost and the probability that the incentive is accepted. Note that for a given incentive i , the corresponding values of $\Delta_{time}(i)$ and $ODL(i)$ are fixed. This can be formulated as follows:

$$f_i(\Delta_{cost}(i)) = \max_{\Delta_{cost}(i) \geq 0} \hat{P}_{acc}(\Delta_{time}(i), \Delta_{cost}(i)) \cdot (w \cdot ODL(i) - \Delta_{cost}(i)) \quad (2.1)$$

The objective is to maximize the expected additional profit of offering the incentive. The additional profit is defined as the extra profit obtained compared to the base case when the incentive is not offered. It consists of the weighted omitted demand loss, minus cost incurred by offering the discount. This discount, of course, needs to be non-negative. An additional constraint may be imposed which says that the discount cannot be higher than the price of the trip. The expected values of w and $ODL(i)$ are inserted in the objective function in a predict-then-optimize fashion. As the objective is linear in these uncertain parameters and given their likely independence, this does not effect optimality (Elmachtoub and Grigas, 2017). The value of $ODL(i)$ is estimated using Markov chains as explained in the remainder of this section. The function P_{acc} is estimated using a logistic regression model, using independent data to avoid bias caused by the optimization. This is explained in detail in Section 2.2.6. The optimal value of each incentive can be found by solving the optimization problem in (2.1), which can be done efficiently as the function has a single stationary point, as stated in Theorem 1. A proof of this theorem is included in Appendix A.

Theorem 1. *If for a given incentive i a profitable discount value $\Delta_{cost}(i)$ exists, there exists a unique most profitable (optimal) discount value $\Delta_{cost}^*(i)$ for which the derivative of the subproblem is equal to 0.*

Following from this theorem and using the fact that discounts are in whole cents and therefore integer, we initialize $\Delta_{cost}^0 = w \cdot ODL$ and iteratively reduce the discount until the objective function starts to decrease or if it is equal to zero. If the function starts to decrease, the optimal incentive is found. If the objective value is maximal at zero, no discount exists for which this incentive is profitable. The optimal discount $\Delta_{cost}^*(i)$ is non-decreasing in the value of $ODL(i)$ if all other variables remain constant. This is shown analytically through Theorem 2, of which a proof is included in Appendix A. This implies that incentives that yield a higher expected omitted demand loss in general receive higher discounts.

Theorem 2. *The optimal discount $\Delta_{cost}^*(i)$ is non-decreasing in the value of $ODL(i)$*

The best incentive can be chosen by optimizing over the set of possible incentives. We refer to the optimal incentive as i^* and to the corresponding optimal value of the incentive as $\Delta_{cost}^*(i^*)$ (which is the argument of the sub-problem). The total objective can be formulated as follows:

$$i^* = \arg \max_{i \in I} f_i(\Delta_{cost}^*(i)) \quad (2.2)$$

2.2.3 Modeling the state of the system and vehicle imbalances

To determine the value of $ODL(i)$ we construct a Markovian model expanding the model proposed by Repoux, Kaspi, Boyacı, and Geroliminis, 2019. We consider a separate Markov chain for every station, which allows us to define the expected loss of future demand given the current state of the system. Through this Markov chain, we incorporate trip reservation information to better predict future states of the system. Every parking spot at a station can have one of the following five states: occupied by an available vehicle (x_{av}), occupied by a reserved vehicle for either a one-way (x_{rv}) or a two-way trip ($x_{rv'}$), not occupied but reserved for a vehicle (x_{rp}) or not occupied and available (x_{ra}). Therefore, the state of a station is defined by the number of parking spots that are in any of the first four states. As the capacity C of a station is fixed and known, the number of parking spots in the fifth state can be deduced from the first four.

The state of a station changes because of arrivals of vehicles or reservations made by users. For this, arrival rates can be determined based on historic data. We determine arrival rates of users looking to rent a vehicle and arrival rates of users returning a vehicle to a reserved parking spot. Hourly arrival rates are used to capture the dynamic demand pattern of the historic data. Using these states and arrival rates we can estimate the expected loss of users. Loss of user demand is encountered if either the desired pickup location has no available vehicles or the desired drop-off

location has no available parking spaces. The expected demand loss is numerically obtained in the same way as in Repoux, Kaspi, Boyacı, and Geroliminis, 2019 using the approximation method described by Raviv and Kolka, 2013. We denote the expected loss given the state of the station as $EL(x_{av}, x_{rv}, x_{rv'}, x_{rp})$.

The omitted demand loss can then be calculated as the difference between the expected demand loss in the original situation and the expected demand loss after the relocation has been performed. Following Repoux, Kaspi, Boyacı, and Geroliminis, 2019, we first determine the omitted demand loss for every station separately, given the implicit relocations in Figure 2.1. The ODL for every station is given in Equations (2.3) - (2.6). For ease of notation, the variables x_{av} , x_{rv} , $x_{rv'}$ and x_{rp} belong to the station of the corresponding ODL . Intuitively, without an incentive, a vehicle is reserved at station o and no change is observed at station o^* . If the incentive is accepted, a vehicle is reserved at station o^* and no change is observed at station o . A similar intuition applies to the destination stations.

$$ODL_o = EL(x_{av} - 1, x_{rv} + 1, x_{rv'}, x_{rp}) - EL(x_{av}, x_{rv}, x_{rv'}, x_{rp}) \quad (2.3)$$

$$ODL_{o^*} = EL(x_{av}, x_{rv}, x_{rv'}, x_{rp}) - EL(x_{av} - 1, x_{rv} + 1, x_{rv'}, x_{rp}) \quad (2.4)$$

$$ODL_d = EL(x_{av}, x_{rv}, x_{rv'}, x_{rp} + 1) - EL(x_{av}, x_{rv}, x_{rv'}, x_{rp}) \quad (2.5)$$

$$ODL_{d^*} = EL(x_{av}, x_{rv}, x_{rv'}, x_{rp}) - EL(x_{av}, x_{rv}, x_{rv'}, x_{rp} + 1) \quad (2.6)$$

Using this, we can compute the total omitted demand loss as a consequence of the incentive. Other than in Repoux, Kaspi, Boyacı, and Geroliminis, 2019 where the relocation is always between a unique origin and a unique destination, user-based relocations depend on the relation between the four stations that may be included in the relocation. If the origin station is not changed because of the incentive, $o = o^*$ and the first two components (i.e. ODL_o and ODL_{o^*}) cancel out. Similarly, the last two components cancel out if the destination station is not changed. We consider other special cases for which two or more stations are equal in a similar manner. For example, in case the original trip is a two-way trip (i.e. $o = d$), we consider the reservation of a round-trip vehicle at that station. If the incentive changes one of the stations, the trip becomes a one-way trip instead. Therefore, we verify for every incentive the exact reservations that were made in the old and the new situation, to accurately estimate the omitted demand loss for every involved station. For example, consider a user travelling from A to B and an incentive being offered to change the destination station from B to A . In the old situation, a vehicle was reserved for a one-way trip at A , whereas in the new situation a vehicle is reserved for a two-way trip. At B , a parking space was reserved in the old situation,

but remains unused in the new situation. This yields the following calculation of the omitted demand loss with respect to this incentive:

$$ODL_A = EL(x_{av} - 1, x_{rv} + 1, x_{rv'}, x_{rp}) - EL(x_{av} - 1, x_{rv}, x_{rv'} + 1, x_{rp}) \quad (2.7)$$

$$ODL_B = EL(x_{av}, x_{rv}, x_{rv'}, x_{rp} + 1) - EL(x_{av}, x_{rv}, x_{rv'}, x_{rp}) \quad (2.8)$$

If the original trip is unavailable, the composition of the omitted demand loss, to which we will refer as ODL' , is also slightly different. Stations o and d are ignored because no change is observed here. If the incentive is accepted, vehicles and parking spaces are reserved at the incentivized location and if the incentive is not accepted, the user is lost and therefore no reservations are made. In case the original trip is not available, the corresponding user is not lost if the incentive is accepted, but is lost if it is not accepted. This user is therefore included in the omitted demand loss.

$$ODL(i) = ODL_o + ODL_{o^*} + ODL_d + ODL_{d^*} \quad (2.9)$$

$$ODL'(i) = 1 + ODL_{o^*} + ODL_{d^*} \quad (2.10)$$

Similar to Repoux, Kaspi, Boyacı, and Geroliminis, 2019, the omitted demand loss is based on a 2-hour time window. As we consider short term omitted demand losses, some incentives may have a negative effect in the long run. This can be reduced by choosing a longer estimation window. However, as the estimations do not incorporate future incentives nor relocations and contain a lot of uncertainty, the estimation quality decreases as the length of the estimation window increases. Most importantly, we aim to improve the system in the short term. The reason for this is that demand is highly asymmetric and stations that require additional vehicles in the short term may no longer require these in the long term. Our user-based relocations focus to solve short-term imbalances in the system, for which a 2-hour time window has shown to be suitable.

2.2.4 Staff-based relocations

Besides the incentivizing method, we consider predictive staff-based relocations. We consider the Markovian relocation policy as described by Repoux, Kaspi, Boyacı, and Geroliminis, 2019 as a benchmark for the performance of our policy. As soon as a staff member is not occupied, his next job is determined by considering all origin and destination stations, which we denote by s_1 and s_2 respectively. The origin and destination stations are selected such that the weighted expected omitted demand loss is maximized. We weight the ODL by the time it takes to get to the origin location, $move(s_1)$, and the time needed to execute the relocation, $drive(s_1, s_2)$. This leads to the following maximization problem:

$$(o^*, d^*) = \arg \max_{s_1, s_2} \frac{ODL_{s_1} + ODL_{s_2}}{move(s_1) + drive(s_1, s_2)} \quad (2.11)$$

To reduce the number of staff-based relocations, we extend this policy by introducing a threshold value. We assume that the relocation is only executed if the expected omitted demand loss $ODL_{s_1} + ODL_{s_2}$ is higher than some threshold value τ . By introducing this threshold value, staff members no longer perform relocations that bring forth little additional demand.

User-based and staff-based relocations have some fundamental differences. Staff-based relocations can only be performed whenever a staff member is available, which limits the total number of relocations. On the other hand, user-based relocations can in theory be performed by every user and therefore does not have this limitation. In terms of feasibility, user-based relocations can only be performed if the user can reach the station without walking too far and they can always decline a request for a change. A staff member does not have this restriction and can therefore do any relocation at the time he is available.

2.2.5 User decision-making

To model the user decisions, we define the probability function P_{acc} . We assume that the probability with which a user accepts an incentive depends on the discount value of the incentive and the additional access time that is experienced because of this incentive. This type of user decisions is commonly modelled using a binomial logistic (logit) model. A similar model has been used by Di Febbraro, Sacco, and Saeednia, 2018. The acceptance probability is defined as follows:

$$P_{acc}(X) = \frac{1}{1 + e^{-(\beta X)}}, \quad (2.12)$$

with $X = [\Delta_{time}, \Delta_{cost}]^T$. As βX may be negative, $P_{acc}(X)$ varies between 0 and 1. Two important notes have to be made considering this probability function. First, users are heterogeneous in the sense that they value time differently. This suggests that the parameters $\beta = [\beta_{time}, \beta_{cost}]$ are user-specific. Second, the actual shape of the probability function P_{acc} is unknown to the operator. The operator can, however, use previously observed data to create an estimation of the parameters $\hat{\beta}$ through learning over time and thereby estimate the probability function \hat{P}_{acc} . The estimation of this function is discussed in detail in Section 2.2.6.

A user's value of time can be determined as the relative importance of the coefficients β_{cost} and β_{time} , which can be estimated by taking the ratio of the two. The higher the ratio of these coefficients, the higher a user's value of time. A user's value of time can then be interpreted as the additional discount a user wishes to receive for one minute of extra access time. If the offered discount is exactly equal to the value of time, the user is indifferent between accepting and not accepting such that the acceptance probability is equal to 0.5. We note that it is also possible to directly

model the value of time of a user by considering the fraction $\frac{\Delta_{cost}}{\Delta_{time}}$.

In addition to this probability model, some hard constraints may apply to the choice to accept an incentive. It is commonly assumed that users are not willing to walk too far to pick up their vehicle or reach their destination after delivering their vehicle. For example, Schulte and Voß, 2015 assume that users choose an alternative mode of transportation if they have to walk for more than 500 meters to reach their vehicle. We use a similar assumption, that says that users never accept an incentive that requires them to increase their one-way access time by 7 minutes (≈ 450 meters). This constraint can be easily incorporated in the definition of the set I . A major advantage of this is that it reduces the computation time of problem (2.2). Low computation time is of importance to the operator, as an incentive has to be offered immediately after users reveal their preference. We emphasize that any other relevant constraints on the feasibility of representatives can also be implicitly incorporated in the set I . This constraint requires that the density of stations should be quite high, so that a number of alternatives within this walking distance exists. While this might be the case for the city centers of major cities with car-sharing services, lower density of stations might exist in the suburbs deteriorating the rebalancing power of this policy. For this reason we will test policies that consider a mixture of incentives and staff relocations.

Truthfulness

An important property of an incentivization policy is that it forces the users to be truthful. An untruthful user purposefully reports wrong information for his/her own benefit. In our case, this means that he/she specifies a wrong pickup or delivery station, as he/she knows he/she will receive a discount for his/her actual preferred station. As untruthful behaviour can have negative effects on the revenue collected by the operator, the policy should avoid untruthful behaviour. In this section, we elaborate on the truthfulness of users under the designed policy.

A user can gain from being untruthful if he/she purposefully report a wrong pickup or delivery station and receive an incentive for his/her actual preferred station, thereby reducing his/her cost. On the contrary, he/she loses from being untruthful if this incentive is not offered and therefore his/her access time is increased. Based on this intuition, the expected gain of being untruthful is defined as the expected incentive value multiplied by the probability of that incentive being offered. The expected loss of being untruthful is the additional time the user needs to walk

if the incentive is not offered, multiplied by his/her value of time (vot) and the probability of no incentive being offered. That is,

$$\mathbb{E}(gain) = \mathbb{E}(\Delta_{cost}) \cdot P(\text{desired incentive offered}), \quad (2.13)$$

$$\mathbb{E}(loss) = \Delta_{time} \cdot vot \cdot P(\text{desired incentive not offered}). \quad (2.14)$$

Of course, users can cancel their reservation if the incentive is not offered and make a new reservation for their actual preferred station. However, this behaviour can be recognized by the reservation tool as untruthful. If a user is recognized to behave untruthfully, no incentive will be offered to this user in the future. Under the assumption that users are risk-neutral, a user will be untruthful if:

$$\mathbb{E}(gain) \geq \mathbb{E}(loss). \quad (2.15)$$

By rewriting this equation, we observe that a user will be untruthful if:

$$\mathbb{E}(\Delta_{cost}) \geq \Delta_{time} \cdot vot \cdot \frac{P(\text{desired incentive not offered})}{P(\text{desired incentive offered})}. \quad (2.16)$$

Equation (2.16) can be seen as a condition for truthfulness. We emphasize that the desired incentive is unknown to the operator but the corresponding probability can be bounded as follows:

$$P(\text{desired incentive offered}) \leq P(\text{any incentive offered}), \quad (2.17)$$

$$P(\text{desired incentive not offered}) \geq P(\text{no incentive offered}). \quad (2.18)$$

Thereby, our results show that in general, Δ_{cost} is not much higher than $\Delta_{time} \cdot vot$. In addition to this, users are in general risk-averse, which means they value losses higher than gains. Following these arguments, we conclude that these conditions are in general satisfied and are therefore not included in the simulation model.

This restriction can be enforced either on a user-based, station-based or a system-based level. In case a violation of the truthfulness restriction is observed, it can either be enforced by reducing the number of offered incentives or by limiting the maximum discount value offered. Both conditions can be incorporated in the optimization problem in Section 2.2.2. Intuitively, some incentives are easier to anticipate than others. For example, experienced users can identify incentives offered when the original trip is not available more easily compared to other incentives. How and within what time-span strategic users are able to predict incentives is an interesting topic of further research.

Incentives can create a new way for individuals to earn money. As individuals are paid to relocate vehicles, this may attract new users that are solely looking to create some income without interest for a specific travel. As these users are new to

the system and are only offered those incentives from which the system benefits, they cannot have a negative effect on the performance of the system and may only improve performance. Nevertheless, a demand model to integrate these actions is beyond the scope of the work and it might require data that are not readily available. Thus, our focus remains only on travelers that are willing to change their origin or destination for some discount in their trip.

2.2.6 Learning from user behaviour

The efficiency of the proposed service depends on the willingness of travelers to accept the offered incentive. Nevertheless, a high value of discount might create losses for the operator. Thus, learning the user behavior and having a model that adequately predicts the acceptance probability as a function of the value of incentive is an important aspect of the framework. In this section, we describe the estimation method of the acceptance probability function. The acceptance probability function has the shape of a binomial logistic (logit) model. The operator does not have any information on the values of the coefficients in β , but it does have full knowledge of the offered incentive and therefore the values of Δ_{time} and Δ_{cost} . In addition to this, the operator can observe the outcomes of the offered incentives. That is, whether the incentive is accepted or not. Using this, we can estimate the probability function using a maximum likelihood estimate of the coefficients in β . As both the dependent (acceptance choice, hereafter also referred to as y) and independent variables (value of the incentive and additional access time, hereafter also referred to as X) are known, they can be used to estimate the corresponding values of the coefficients. The likelihood function corresponding to the binary logit model with n observations is written as follows:

$$L(\beta) = \prod_{i=1}^n P(X_i)^{y_i} (1 - P(X_i))^{1-y_i}. \quad (2.19)$$

The optimal value of β is the one that maximizes the likelihood function. Instead of maximizing the likelihood function, it is easier to maximize the log-likelihood function which is given as follows.

$$l(\beta) = \sum_{i=1}^n y_i \ln(P(X_i)) (1 - y_i) \ln(1 - P(X_i)). \quad (2.20)$$

As no analytical solution exists, we use a numerical optimization approach to find the optimal value for β . We use a steepest-descent algorithm with decaying step-size. This estimation method suggests that we can train our model using previously observed data and use this to forecast the probability that a user accepts the offered incentive. An advantage of the described methods is that, besides the origin and destination location of a user request, no other information is required. This limits the possibilities for users to be untruthful about personal information to maximize

their own profit and therefore contributes to the truthfulness of our method. This method can be used to obtain user-specific estimates or one estimate for the entire population. If a user-specific estimate is obtained, only those observations corresponding to that user are used to train the logit model. If a single estimate is obtained for the entire population, all observations are used. In this case, our method is used to estimate a sample average value of β .

The use of this learning method in combination with the optimization with the optimized incentives as described in Section 2.2.2 will create a measurement bias. The reason for this is that the input variable Δ_{cost} is optimized based on the same acceptance probability function we try to estimate. Experiments show that this generally leads to an overestimation of the value of time. Therefore, we first use a training period to estimate the value of the coefficients using the described maximum likelihood methods. During this training period, the optimal incentive is determined using the methods described in Section 2.2.2, but the discount value Δ_{cost} is randomly drawn from a uniform distribution on the interval $[0, w \cdot ODL]$. After the training period, the performance of the incentivization method is evaluated using the optimized discount using the estimated coefficients $\hat{\beta}$. In case user-specific parameter estimates have to be obtained, newly arriving users are treated in a similar way. The first discounts are determined randomly during a training period until an adequate estimation can be made. Alternatively, discounts for newly arriving users can be determined using estimates of a set of existing users.

In reality, estimates can be further improved by grouping users with similar features. Travelers generally have to create an account to utilize the car-sharing system. They can then be grouped according to relevant features such as age or occupancy, such that group-specific estimates can be obtained. For example, it is likely that students have a lower value of time than elderly people. By using group-specific estimates, acceptance probability estimations can be improved.

2.3 Experimental results

The relative performance of the incentivization method described in the previous sections is evaluated using a case study of the Grenoble car-sharing system. The details of this case study and the cost structures we use in our evaluations are described in Section 2.3.1. In Section 2.3.2 we describe the simulation model. In the following sections, experimental results are provided that give insights into the relative performance of the described methods.

2.3.1 Case study: Grenoble car-sharing system

In our case study, we consider the Grenoble car-sharing system, which has been previously studied by Repoux, Kaspi, Boyacı, and Geroliminis, 2019. The system was operational between September 2014 and November 2017 and was based on a complete journey reservation policy, as described in Section 2.2. The system consisted of 27 stations with a total of 121 parking spots (each station had between 3 and 8 parking spots). In our simulation framework, 40 electric vehicles are available in the system every day. The maximum speed of the vehicles is equal to 50 km/h, corresponding to the speed limit in urban areas in France. In this case study, we disregard the battery restrictions of the vehicles, as previous studies have shown that in station-based systems these influence the results only marginally. As we compare our methods to the Markovian staff-based relocation policy designed by Repoux, Kaspi, Boyacı, and Geroliminis, 2019, we use similar settings for this policy.

Our simulation is based on demand data of the actual car-sharing system. Every simulation run consists of 10 consecutive days. We generate 100 random synthetic demand realizations per day, based on the observed distribution of demand in the actual system. As the exact itinerary of a trip is unknown, trip distance and trip duration are assumed to be independent of incentives. This distribution is based on trip transaction data from the operational period. The system is operational 24 hours per day, but the majority of the trips occur between 7 a.m. and 8 p.m.. As no user information was collected by the car-sharing system, every demand realization is randomly assigned to one of 50 users, which allows us to evaluate the effect of our learning procedure within the set time-horizon. We note that user information is only required for our learning procedure. Each user has a specific value of time. The values of time are drawn from a normal distribution with mean €0.30 per minute and standard deviation of 0.10. We assume β_{time} is fixed at -0.75 (in minutes) such that β_{cost} follows directly from the value of time. We emphasize that the number of users does not influence any of the obtained results other than the learning procedure. The choice of parameter w depends on the importance of the service level relative to the profit. We choose the value for w equal to the average revenue earned for a single demand unit, which is equal to approximately €15.

Walking and public transport times between stations for the city of Grenoble have been extracted from Google, 2019. The public transport time comprises walking time to reach public transport and time spent in public transport. In case walking from origin to destination is the least time-consuming option, walking time is used as the full travel time. Staff members also either walk or use public transport, depending on which is faster, to move between stations when they are not relocating. For consistency, we use the same moving times as considered by

Repoux, Kaspi, Boyacı, and Geroliminis, 2019.

Finally, the profit is based on various costs similar to those defined by Boyacı, Zografos, and Geroliminis, 2015. The profit is calculated as the user revenue based on a cost of €0.20 per minute minus the cost of relocators (€18 per hour), fixed vehicle cost (€20 per day) and a cost of €0.01 per kilometer travelled by both users and relocators. In practice, users pay €3 for every 15 minutes, so their trip duration is rounded up to 15 minutes.

2.3.2 Simulation model

Our experimental results are obtained using an event-based simulation framework. The simulation framework is an extended version of the developed framework by Repoux, Boyacı, and Geroliminis, 2015 and later updated by Repoux, Kaspi, Boyacı, and Geroliminis, 2019. For a detailed description of the framework, the reader is referred to these papers. The framework simulates the actual situation of the Grenoble car-sharing system as described in Section 2.2. The event-based simulator models vehicle reservations, pickups and drop-offs. Thereby, it keeps track of the status of vehicles at stations and on the road and staff members.

The network of stations is taken directly from the Grenoble car-sharing system. Travel times for users between stations are extracted from Google, 2019. We emphasize that we incorporate asymmetries in both walking and transit times. Synthetic data is used to model the demand for vehicles and parking spaces. Arrival rates for origin-destination pairs are estimated based on observed demand during the period when the system was active.

We extend the simulation framework by the described incentivization procedure. For every user entering the system, we solve the problem described in Section 2.2.2 to determine the optimal pickup and delivery location and discount value corresponding to the incentive, if any beneficial incentive exists. After the incentive is offered, the response of the user to this incentive is randomly drawn corresponding to the logistic distribution described in Section 2.2.5. In addition to this, we implement a learning procedure which allows the operator to learn from previously observed user behaviour to determine unobserved user preferences. This procedure is described in detail in Section 2.2.6.

2.3.3 Model evaluation

While user-based relocations only change the origin or the destination station within the proximity of the original trip, they can help to locally rebalance the system. Staff-based relocation can perform any movement of an empty vehicle between two stations, but they might increase the operational cost. Thus, we are interested in the performance of the system for different combinations of user-based and staff-based relocations.

We first consider the general user-based relocation policy as described in Section 2.2.2. We evaluate the relative performance of this policy under the assumption that the operator has perfect information on the value of time of users, that is $\hat{P}_{acc} = P_{acc}$. The maximum one-way access time is equal to 7 minutes (≈ 450 meters). The total additional access time a user experiences may therefore be at most 14 minutes, but this is not commonly observed. The average results of 100 simulations are reported in Table 2.1. We present different performance measures that can ease our understanding of the system from the perspective of the users and the operators. The number of active personnel varies between 0 and 3 with or without incentives.

Table 2.1: Simulation results

Staff	Incentives	% served	# relocations	% accepted	# incentives	KM	Profit
0	No	59.7	0.0	0.0	0.0	5.82	62.79
	Yes	71.5	0.0	56.7	21.5	5.78	183.66
1	No	84.6	33.9	0.0	0.0	6.36	206.93
	Yes	91.8	30.8	45.9	21.4	6.31	263.74
2	No	89.6	62.1	0.0	0.0	6.97	61.17
	Yes	94.7	58.4	40.8	18.6	6.90	95.77
3	No	91.0	86.0	0.0	0.0	7.49	-135.92
	Yes	95.4	81.4	38.5	17.2	7.40	-106.52

The first two columns describe the relocation policy in place (i.e. number of staff members and whether incentives are used). The third column denotes the percentage of served users. The fourth column contains a daily average of the number of relocations performed. The fifth and sixth column contain the percentage of offered incentives that are accepted and the actual number of incentives accepted respectively. The KM travelled is measured as an average per served demand unit and includes both user and staff KM travelled. The profit is denoted in euros per day.

The results indicate that by offering incentives, the service level can be increased significantly. Thereby, by only offering incentives if they are expected to be profitable, the profit also significantly increases. We also observe that incentives are much more sustainable compared to staff-based relocations. Whereas staff-based relocations significantly increase the average kilometers travelled, this is not the case for incentives.

Due to staff-based relocations, the service level can be increased to a percentage between 84.6% and 91%, depending on the number of staff members, but the profit decreases if the number of personnel is higher than 1 (and it becomes negative for 3 or more). By using incentives without any personnel, this is only 71.5%. The main reason for this is that user-based relocations are limited to short-distance relocations, while staff members can also do long-distance relocations. By combining the two policies (one staff member and incentives), the profit is optimized and the service level is higher than the one with three staff members and no incentives. Interestingly, offering incentives with one personnel is capable of serving more users compared to two personnel with no incentives, which also has a significantly higher operational cost.

Daily, approximately 20 incentives are offered and accepted, depending on the policy that is used. Note that the total number of offered incentives can be obtained directly from the number of accepted incentives and the percentage of accepted incentives. As daily demand is equal to 100, this means an incentive is accepted by approximately 20% of the arriving users. This supports the truthfulness of our policy, as discussed in Section 2.2.5. Approximately 55% of the offered incentives are accepted if only incentives are used, which decreases if it is combined with staff members. As proven in Theorem 2 in the Appendix, the discount value is non-decreasing in the expected omitted demand loss. As staff-members reduce imbalances in the systems, the expected omitted demand loss of incentives tends to decrease. In turn, this decreases the offered discounts. As a consequence, the acceptance probability of those incentives decreases and thereby the percentage of accepted incentives decreases. This also means that if the original trip is not available, the offered incentive is much more likely to be accepted as the lost user is incorporated in the objective function.

By offering incentives or performing relocations, less critical situations (i.e. no available vehicles or no available parking places) at stations arise. Figure 2.2 displays the number of stations with at least one available vehicle and at least one available parking space. We compare the scenario where no relocations are performed to the scenario where incentives are used, obtained using a single simulation of 10 days. No staff members are used in both scenarios. The results indicate that, by using incentives, slightly more stations have both available vehicles and parking spaces. Because less critical situations arise, more demand can be served which is in line with the results in Table 2.1.

Figure 2.3 graphically represents the relocations performed by staff members and due to incentives. Figure 2.3A displays the incentives on origin locations (origin locations changed because of incentives), Figure 2.3B displays the average walking

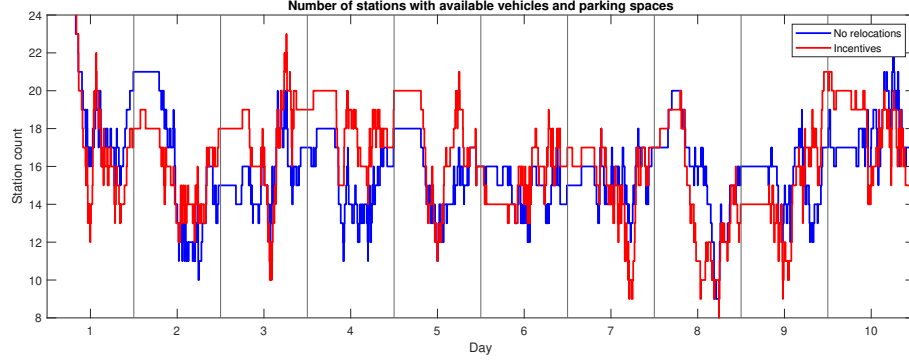


Figure 2.2: Graphical representation of the number of stations with available vehicles and parking spaces over time for one simulation

time between two stations in minutes, Figure 2.3C displays the incentives for which the original trip was unavailable and 2.3D displays the staff-based relocations. Figure 2.3C and 2.3D display those relocations that occur at least once every 20 or 10 days respectively. The relocations correspond to the simulations for which either only incentives or only staff is used and are an average of 100 simulation runs.

The results confirm the intuition that incentives are used for short-distance relocations. The relocations are solely between stations that are within 7 minutes walking distance from each other. Staff-based relocations, on the other hand, can relocate vehicles between any two stations. A similar graph can be obtained for incentives on the destination location. If the original trip is unavailable, the relocations look more like the staff-based relocations as either the origin or destination can be outside the maximum access time range. Incentives on unavailable trips are mostly used to change the origin location of the trip, which can be seen from the stations that are selected as origins in Figure 2.3C. The reason for this is that, due to the high number of parking spaces (121) compared to the number of vehicles (40), the unavailability of vehicles at the origin station is more problematic than the unavailability of parking spaces at the destination station. If we reduce the number of parking spaces, we observe that the number of incentives regarding an unavailable vehicle and those regarding an unavailable parking space become roughly similar. By combining staff and user-based relocations, we are using a hybrid operator-user-based relocation policy. In this policy, user-based relocations are used for short-distance relocations and are extremely effective if the original trip is unavailable. Thereby, staff-based relocations can be used to cover imbalances over longer distances such as between suburbs and the city center, as is illustrated in Figure 2.3. Many incentives apply to stations 4, 5 and 10. These stations are located close to the train station of Grenoble, with 5 located approximately between 4 and 10. Not coincidentally, station 5 is also the station with the highest demand.

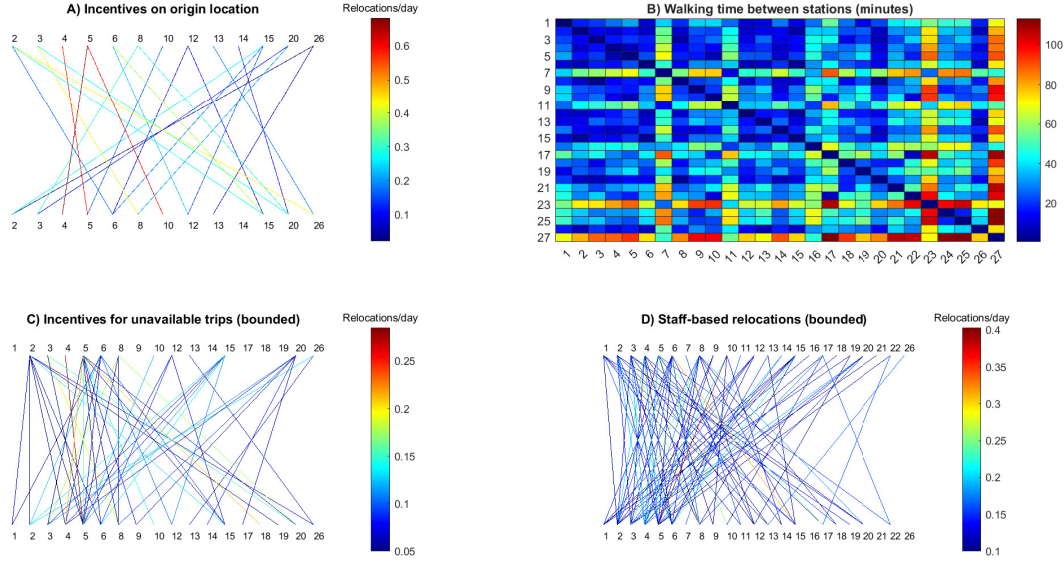


Figure 2.3: Graphical representation of relocations

One of the advantages of our method is that it is applicable in real-time operation. For an incentivization method to be applicable in real-time operation, it should be able to determine the optimal incentive (if any) within seconds. Our simulation results illustrate that this condition is satisfied and our method is computationally very efficient. By limiting the number of feasible incentives and using the property of the subproblem that has at most one stationary point, the optimal discount can be found very fast. This suggests that our model can also be applied to larger cities, where the number of feasible incentives is typically much higher, because stations are located closer together. As the number of feasible incentives is higher in larger cities, our user-based relocation approach is expected to perform even better in these cities.

Demand rates may change due to the used relocation policies. In general, if a user is not served he/she is less likely to return in the future. In addition to this, low availability of vehicles may decrease the demand rates at those stations whereas high availability at other stations may increase the demand rates there. The pricing policy may therefore change the demand rates (as may any staff- or user-based policy). To anticipate this change, demand rates can be re-estimated and the omitted demand loss estimations can be updated accordingly. Using such an iterative process, dynamically changing demand can be anticipated indirectly. To directly anticipate dynamically changing demand a proper demand model is required (depending on service level, pricing and other features), which is outside the scope of this work.

2.3.4 Learning evaluation

In this section, we evaluate the performance of our learning algorithm. We use our learning algorithm described in Section 2.2.6 to estimate a single acceptance probability function for the entire population. We assume that the value for β_{time} is fixed and known for all users, whereas the value for β_{costs} is drawn randomly and unknown. We emphasize that, if enough user-specific data is gathered, the exact same procedure can be used to obtain a user-specific acceptance probability function. If enough data is gathered, the performance using user-specific estimates will attain the performance under perfect information. We compare the performance of the learning algorithm to the performance when the value of time is known exactly and when the value of time is underestimated by 30%. For the latter case, we approximate the value of time by a single estimate which is 30% lower than the population average.

Table 2.2 presents the simulation results for this experiment. A training period of 3 days is used. The simulation results are therefore an average of the last 7 days. We observe that the performance of the learning algorithm increases with the length of the training period. After 3 days, the performance does not increase significantly.

Table 2.2: Simulation results with learning method

Staff	Estimation	% served	% accepted	# incentives	discount	Profit
0	Exact	70.4	56.8	21.3	41.08	173.74
	Learning	69.9	54.1	20.5	42.86	168.76
	Underestimate	68.8	43.6	17.6	33.05	167.48
1	Exact	91.7	45.5	21.3	37.05	267.66
	Learning	91.2	43.8	20.5	38.66	260.80
	Underestimate	90.2	35.4	17.4	30.26	260.15
2	Exact	94.6	40.7	18.5	35.61	99.80
	Learning	94.3	39.0	17.7	36.77	94.71
	Underestimate	93.5	31.6	15.2	29.46	94.48
3	Exact	95.3	38.2	16.9	34.52	-102.83
	Learning	95.1	37.6	16.7	36.07	-107.63
	Underestimate	94.4	29.8	14.2	28.69	-106.43

The first column describes the relocation policy in place (i.e. number of staff members). The second column denotes the method used to estimate the value of time: exact, learning or underestimated. The third column denotes the percentage of served users. The fourth and fifth column contain the percentage of offered incentives that are accepted and the actual number of accepted incentives respectively. The average discount per minute of accepted incentives is given in cents. The profit is given in euros per day.

The results indicate that in case the operator does not have perfect information about the user's value of time, incentives are still effective. The observed differences mainly occur because user heterogeneity is ignored and all users are treated as if their value of time is equal. We observe that, even though the average discount per minute is higher, the percentage of accepted incentives is lower. Consequently, fewer incentives are offered and the profit and service level decrease.

If the value of time is underestimated, the percentage of accepted incentives decreases significantly and so does the average discount. As a consequence, the service level decreases. Similar results can be obtained when the value of time is overestimated. In this case, the average discount value will be higher, causing the profit to decrease. This emphasizes the importance of a correct estimate of the value of time, as a wrong estimate can decrease both the profit and the service level.

Our experiments indicate that users with a higher value of time are offered higher discounts. A regression of the discount value on the actual value of time of a user indicates that the value of time has a significant positive effect on the discount value. From the experiments in this section, we conclude that our learning methods enable the operator to obtain a good estimate of the acceptance probability function of users. Naturally, as dispersion among the users in terms of their value of time increases, the performance of the learning method decreases. However, when more data is gathered, user-specific estimates can be obtained which are not influenced by dispersion. A more detailed analysis of learning the distribution is beyond the scope of this work, as no real data was available for specific users. This can be a research priority for a demand-oriented analysis.

2.3.5 Increasing user flexibility

In the previous experiments, we assumed the maximum one-way time users were willing to walk towards their pickup location and from their destination location was 7 minutes. In this section, we perform a sensitivity analysis to investigate the effect of increasing user flexibility. We assume the operator has perfect information on the users' value of time. First, we consider three scenarios where the maximum walking time is either 5, 7 or 10 minutes (again, this applies separately to origin and destination). The higher the maximum walking time, the more flexible users are. The results of this experiment are displayed in Table 2.3.

We observe that as the maximum walking time increases, more incentives are offered, as more profitable incentives are found. As a consequence, the percentage of demand served and profit increase significantly. As flexibility increases, the percentage of served demand and effectiveness of incentives increases. This result is in line with those provided by Correia, Jorge, and Antunes, 2014. Due to the increasing number of incentives, the number of staff-based relocations also decreases slightly.

Table 2.3: Simulation results for multiple maximum walking times

Access time	Staff	Incentives	% served	# relocations	# incentives	KM travelled	Profit
5	0	No	59.7	0.0	0.0	5.82	62.79
		Yes	65.1	0.0	11.0	5.79	118.68
	1	No	84.6	33.9	0.0	6.36	206.93
		Yes	88.3	32.2	11.9	6.33	239.59
	2	No	89.6	62.1	0.0	6.97	61.17
		Yes	92.2	59.9	10.7	6.94	80.88
	3	No	91.0	86.0	0.0	7.49	-135.92
		Yes	93.4	83.5	10.1	7.44	-115.82
	0	No	59.7	0.0	0.0	5.82	62.79
		Yes	71.5	0.0	21.5	5.78	183.66
7	1	No	84.6	33.9	0.0	6.36	206.93
		Yes	91.8	30.8	21.4	6.31	263.74
	2	No	89.6	62.1	0.0	6.97	61.17
		Yes	94.7	58.4	18.6	6.90	95.77
	3	No	91.0	86.0	0.0	7.49	-135.92
		Yes	95.4	81.4	17.2	7.40	-106.52
	0	No	59.7	0.0	0.0	5.82	62.79
		Yes	79.4	0.0	32.4	5.76	252.32
	1	No	84.6	33.9	0.0	6.36	206.93
		Yes	94.1	29.7	26.6	6.31	276.15
10	2	No	89.6	62.1	0.0	6.97	61.17
		Yes	95.9	57.1	22.7	6.89	98.10
	3	No	91.0	86.0	0.0	7.49	-135.92
		Yes	96.4	80.0	20.5	7.37	-104.39

The first column denotes the maximum extra access time in minutes. The next two columns describe the relocation policy (i.e. number of staff members and whether incentives are used). The fourth column denotes the percentage of served users. The fifth and sixth columns denote the daily average number of relocations and the number of accepted incentives respectively. The KM traveled is measured as an average per served demand unit and includes both user and staff KM traveled. The profit is given in euros per day.

We note that walking for 10 minutes to pick up or drop off a vehicle can be rather undesirable for users. Therefore, we explore the use of public transportation modes to transport users before pick up or after delivery. The public transport data for the city of Grenoble has been extracted from Google, 2019. The travel

time comprises walking time to reach public transport and time spent in public transport. In case walking from origin to destination is the least time-consuming option, walking time is used as the full travel time. On top of that, the user may wish to be compensated for the inconvenience of public transport (which comprises among others waiting time and scheduling delay). We assume users expect to be compensated for inconvenience comparable to five minutes of walking. This is incorporated in the acceptance probability function and therefore indirectly leads to higher compensations in case public transport is used.

The results of this experiment are presented in Table 2.4. We consider two scenarios, where the maximum time to get to or from the vehicle is either 7 or 10 minutes. For the sake of comparison, we assume the value of time is similar for both experiments. We note that the value of time in public transport is likely to be higher than that for walking, as the price of the public transport ticket needs to be paid, but this is omitted in the current work.

The results indicate that if the maximum one-way access time in public transport is 7 minutes, the results are better than walking for 10 minutes. The reason for this is that if users are willing to use public transport, they can reach a higher number of stations within their maximum access time. When the maximum access time is equal to 10 minutes in public transport, the use of incentives outperforms the use of 3 staff members. The main reason for this is that if a user requests an unavailable trip, an alternative origin or destination can almost always be reached within the maximum access time. Therefore, by offering a discount this user can often be saved without the need to relocate vehicles. Although the use of incentives outperforms the use of staff members in this case, 41 incentives per day are needed to achieve this. This means that almost half of the arriving users change their preferred origin or destination station. This has a negative effect on truthfulness and, despite the fact that users are compensated, it is likely to decrease user satisfaction. Thereby, if users demand higher compensations for the inconvenience of using public transportation, the price of incentives will increase. As a consequence, profit goes down and service level goes down as less profitable incentives exist.

The results of this experiment are promising in the sense that if users are willing to combine multiple transportation modes, i.e. public transport and car-sharing, the balancing problem can be solved efficiently without using any staff members. This implies that incentives are a sustainable alternative to staff-based relocations. A downside of this is that in practice users may choose to waive their car-sharing request when they are already in public transport. A thorough user survey is required to evaluate whether the increased service level outweighs the potential demand lost to public transport.

Table 2.4: Simulation results using public transport

Access time	Staff	Incentives	% served	# relocations	# incentives	KM travelled	Profit
7	0	No	59.7	0.0	0.0	5.82	62.79
		Yes	80.6	0.0	32.0	5.77	263.96
	1	No	84.6	33.9	0.0	6.36	206.93
		Yes	95.0	30.4	26.9	6.31	282.00
	2	No	89.6	62.1	0.0	6.97	61.17
		Yes	96.8	58.0	22.9	6.89	105.75
	3	No	91.0	86.0	0.0	7.49	-135.92
		Yes	97.2	81.0	20.6	7.37	-97.69
	0	No	59.7	0.0	0.0	5.82	62.79
		Yes	91.9	0.0	41.6	5.76	370.87
10	1	No	84.6	33.9	0.0	6.36	206.93
		Yes	97.5	29.2	30.9	6.30	289.08
	2	No	89.6	62.1	0.0	6.97	61.17
		Yes	98.2	56.7	25.6	6.87	106.42
	3	No	91.0	86.0	0.0	7.49	-135.92
		Yes	98.5	79.4	23.5	7.34	-96.23

The first column denotes the maximum extra access time in minutes. The next two columns describe the relocation policy (i.e. number of staff members and whether incentives are used). The fourth column denotes the percentage of served users. The fifth and sixth columns denote the daily average number of relocations and the number of accepted incentives respectively. The KM traveled is measured as an average per served demand unit and includes both user and staff KM traveled. The profit is given in euros per day.

2.3.6 Adapting staff-based relocation policy

As we consider incentives and staff-based relocations that both aim to maximize the omitted demand loss, it is interesting to consider the distribution of the *ODL* obtained by a relocation due to an incentive and that obtained by a staff-based relocation. Figure 2.4 depicts the distribution of the *ODL* values for these two types of relocations. The distribution of the *ODL* values for incentives has two peaks. The reason for this is that incentives can be classified as one of two types: incentives if the original trip is available and incentives if the original trip is unavailable. For the second type, one lost user is omitted with certainty if the incentive is accepted, which can be seen from the plus 1 term in Equation (2.10). Therefore, the *ODL* corresponding to this type of relocation is generally high, causing the second peak.

The two distributions indicate that staff-based relocations on average bring forth lower omitted demand losses than relocations due to incentives. This is partially caused by the second type of incentives for unavailable trips. Another reason for this is that due to the trade-off in the optimization problem, incentives with small *ODL* values have lower discounts and therefore lower acceptance probabilities or they may not even be offered. This causes the *ODL* values to be higher in general. We emphasize that the omitted demand loss is based on a 2-hour time interval. In the long term, relocations generally bring forth higher *ODL* values except for those for which the original trip is unavailable. In that case, if the *ODL* value is smaller than 1, this value is likely to decrease in the long term.

By only offering incentives when the original trip is unavailable, the service level only decreases slightly whereas the profit may even increase. Of course, these incentives do not actually contribute to the rebalancing of the system as only users that would have otherwise been lost are redirected to a different station. These incentives are also more likely to induce untruthful behaviour, as they are easier to identify and compensations are higher. On the other hand, by offering incentives only when the original trip is available, the service level and profit are decreased but are still significantly higher compared to the case when no incentives are offered. Contrary to the other type of incentives, a policy where incentives are only offered in case the original trip is available is more resilient to untruthful behaviour and still leads to a system that is properly rebalanced. Overall, offering incentives independent of the availability of the original trip is highly effective and easier to implement in reality. An additional subtlety for originally unavailable trips is that users are in generally more likely to accept a small detour, but this highly depends on their alternative transportation modes. This means that in reality profit can be even higher by reducing the discounts for those trips.

We can adapt the staff-based relocation policy to the use of incentives by changing the threshold value τ , the minimum expected omitted demand loss for a staff-based relocation to be performed, as defined in Section 2.2.4. By changing this value we can reduce the number of relocations performed by staff members and thereby reduce their overall activity. This can be convenient as in practice staff members often perform maintenance jobs and other tasks if they are not relocating vehicles. We consider various threshold values varying between 0.00 and 0.25. The results of this experiment are provided in Table 2.5. For this experiment, we assume relocators are only paid for the number of hours they effectively worked, corresponding to their occupancy rate.

The results indicate that by increasing the threshold value, we reduce the number of relocations and the percentage of time the staff member is relocating vehicles.

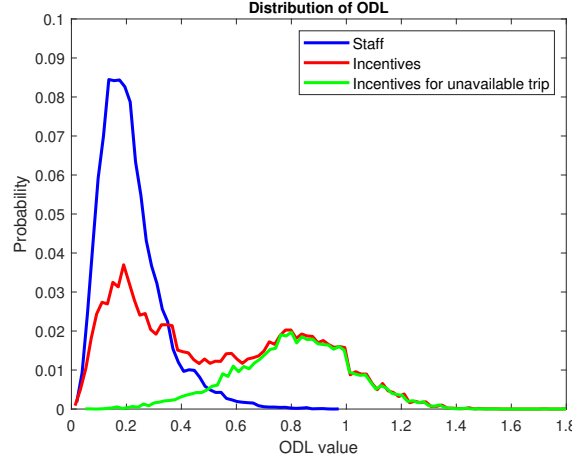


Figure 2.4: Distribution of *ODL* by incentive and staff relocation

Table 2.5: Simulation results for different staff thresholds

τ	% served	# relocations	% occupancy	# incentives	KM travelled	Profit
0.00	91.8	31.0	98.3	21.2	6.32	267.58
0.05	91.9	30.6	98.2	21.2	6.32	268.22
0.10	91.7	28.7	96.4	21.7	6.31	270.96
0.15	91.4	25.0	89.0	21.8	6.25	280.54
0.20	90.5	20.8	77.5	22.2	6.18	290.78
0.25	89.1	16.4	63.5	23.3	6.10	297.64

We consider the case where one staff member is used. The first column denotes the value of τ . The second column denotes the percentage of served users. The third and fifth column are a daily average of the number of relocations performed and the number of incentives accepted respectively. The fourth column denotes the average occupancy of the staff member as a percentage of the total workday. The KM travelled is measured as an average per served demand unit and includes both user and staff KM travelled. The profit is given in euros per day.

A positive finding is that for a value of $\tau = 0.25$, the number of staff-based relocations is almost half compared to $\tau = 0$, but the served demand decreases by only 2.5%. We also observe that the number of incentives increases slightly as the threshold increases. The reason for this is that some of the relocations that are not executed by staff members are now (implicitly) executed by users. By increasing the threshold, the profit increases at the cost of a small decrease in the percentage of served users. The results of this experiment strengthen the idea of a hybrid operator-user-based relocation approach.

2.3.7 Sensitivity analysis: user value of time

In all previous experiments, the user value of time is assumed to be equal to 30 cents per minute of additional walking time. In this section, we perform a sensitivity

analysis on the user value of time to illustrate the robustness of our results. The results of this experiment are displayed in Table 2.6. We consider average values of time which range from 20 to 50 cents per minute. Other than that, the simulation settings are similar to those used in the previous experiments.

Table 2.6: Simulation results for different user value of time

<i>vot</i>	% served	# relocations	% accepted	# incentives	discount/minute	Profit
20	92.0	30.4	55.1	26.2	25.11	273.52
30	91.8	31.0	45.4	21.2	37.42	267.58
40	91.6	31.4	39.4	17.7	49.29	257.83
50	91.3	31.9	35.0	15.2	60.58	250.83

We consider the case where one staff member and incentives are used. The first column describes the scenario, where the value of time is given in cents per minute. The second column denotes the percentage of served users. The third column denotes the percentage of offered incentives that is accepted and the fourth column denotes the actual number of accepted incentives. The fifth column denotes the average discount value per minute for all accepted incentives. The profit is given in euros per day.

The results indicate that as the value of time increases, fewer incentives are profitable and therefore fewer incentives are offered. In addition to this, because of the higher value of time, the average discount value relative to the value of time is lower, which means that the percentage of accepted incentives decreases. For the accepted incentives, we observe that the average discount value increases proportionally with the average value of time.

As fewer incentives are accepted, the service level and profit both decrease. Interestingly, the service level and profit do not decrease significantly even though the number of accepted incentives is almost halved. This is partially caused by the use of staff-based relocations which can replace the incentives as well as later users that are offered a similar incentive. In addition to this, those incentives that are no longer profitable to offer are typically those for which the influence on profit and service level was rather small.

We emphasize that this representation of profit is not realistic, as we use a constant rental price. In cities where the average value of time is higher, a higher rental price can be imposed. Thereby, the value for w (monetary value per unit of lost demand) increases and the effect of value of time will be negligible.

2.4 Summary

In this chapter, we proposed a predictive user-based relocation policy for one-way car-sharing systems. Our method relies on user-based relocations that are stimulated by offering discounts to users. By performing an alternative and less convenient trip, users implicitly contribute to the redistribution of vehicles throughout the system. Our policy uses information on the current state of the system as well as expected future demand to determine appropriate relocations, to reduce expected future demand losses. Our policy is adaptive to the value of time of users. As user preferences such as their value of time are generally unknown, they have to be estimated. We developed a learning algorithm that allows the operator to learn from previously offered incentives and adjust the future offers accordingly.

Our simulation results indicate that, by using our incentivization approach, we can partially solve the balancing problem of vehicles throughout the network and thereby increase the service level. In addition to this, our methods allow the operator to use fewer staff members while attaining a higher service level and thereby increase the profit. Specifically, by using a hybrid operator-user-based relocation policy, service level and profit can be maximized. In this case, user-based relocations perform short-distance relocations, while long-distance relocations are executed by staff members. We also observe that by using user-based relocations, the average KM travelled by staff and users per unit of served demand decreases, suggesting our method is environmentally more sustainable than staff-based policies.

Using a learning algorithm, we can accurately approximate the users' acceptance probability functions. Therefore, we can obtain results that are close to those under the assumption of perfect information. A sensitivity analysis indicates that we can further increase the service level in case users are more flexible. That is if users are willing to walk further to pick up or deliver their vehicle or even use public transport, the effectiveness of our incentivization method increases.

3

Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users

This chapter is based on the article:

- A. de Palma, P. Stokkink, and N. Geroliminis (2022). “Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users”. In: *Transportation Research Part B: Methodological* 155, pp. 479–498

3.1 Introduction

Over the last years, the demand for mobility services such as vehicle-sharing and carpooling (or ride-sharing) has increased tremendously. One of the most popular examples of carpooling is BlaBlaCar, 2020, with 30 million members in 22 countries. Carpooling initiatives are known to have many advantages such as increasing mobility and reducing congestion. Carpooling eases congestion by increasing average vehicle occupancy. According to BlaBlaCar, 2020, they increased the average occupancy of their vehicles to 2.8 compared to an average of 1.6 in Europe. However, carpooling decisions are also influenced by congestion. Congestion brings forth significant economic costs both through extra fuel consumption and waste of time. Furthermore, it induces social costs as it is commonly associated with public health risks. Scheduling preferences are not integrated in most of the matching studies and this inconvenience makes carpooling less attractive. These effects emphasize the need to introduce congestion and scheduling preferences in

carpooling models (literature has remained silent on this issue) to better understand its potential benefits and costs.

We consider the commute of passengers and drivers in a parsimonious framework that will allow us to develop intuition for a challenging problem both in terms of mathematical formulation and economic insights. In our problem, passengers and drivers are located along a horizontal line and all have the same destination at the end of the horizontal line, the Central Business District (CBD). This framework forms a theoretical approximation of many real-life situations, where people live in the suburbs and work in the business district. Suburbs and the CBD are connected by a highway. Contrarily to a formally similar problem, the matching in marriage problem (Chiappori, Oreffice, and Quintana-Domeque, 2012), in our work a central operator matches the drivers and the passengers such that the total matching costs are minimized in the presence of untolled congestion. We refer to this as the *optimal matching problem*. The matching costs include the detour time to pick up a passenger but may also include scheduling delays (associated with early or late arrivals). Congestion can significantly influence travel time and tardiness of all agents and may therefore influence matching decisions. In this work, we incorporate dynamic congestion in matching decisions as well as departure time decisions. In the same unified framework, drivers still determine their time of departure considering scheduling and congestion costs as per W. Vickrey (Vickrey, 1969).

The key element of carpooling is the matching process between drivers and passengers. This is a well-known mathematical problem (the optimal transport model reviewed by Galichon, 2018) that we will apply in this work. In the carpooling context, since the quality of matching depends on travel times, which itself depends on how many matches have been performed, we have an extended version of the standard matching model, that we refer to as the *congested optimal matching model*. Other applications of this congested matching model can be studied, such as the housing market (in that case, matching will potentially change the market prices and the population mix which may feed-back to the matching process) or the taxi market (e.g. the operations research model of stable matching of Bai, Li, Atkin, and Kendall, 2014), but this is clearly outside the scope of this work.

A two-way causality exists between carpooling and congestion. One side of the implication is well-understood and intuitive. If a fraction of the drivers starts to carpool, the number of cars on the road decreases, which eases congestion (Bahat and Bekhor, 2016; Li, Hong, and Zhang, 2016). The other side of the causality is more subtle and unexplored in the standard matching models.

To the best of our knowledge, the effect of dynamic congestion with scheduling

preferences on carpooling matching has not been studied before. Dynamic traffic models (Arnott, Palma, and Lindsey, 1990; Palma, Lefèvre, and Ben-Akiva, 1987) are commonly used to model the relation between congestion and traffic decisions. Smith, 1993 presents a dynamic traffic flow model for peak period traffic flows in urban areas, where road capacity is tight. Morning and evening commutes are commonly modeled using a traffic bottleneck model (Arnott, Palma, and Lindsey, 1993) based on the Vickrey, 1969 model. Later, the corridor model was extended by Qian and Zhang, 2011 to include multi-modal transportation, with among others carpooling. For a recent review of the bottleneck congestion model, the reader is referred to Li, Huang, and Yang, 2020. The effect of fuel prices on carpooling behavior has been studied by Bento, Hughes, and Kaffine, 2013. Their results indicate that flow on high occupancy vehicle lanes on average increases with fuel prices. However, they suggest that local traffic congestion such as bottlenecks may change the way these variables interact. The bottleneck model has been extended with ridesharing options by, among others, Ma and Zhang, 2017, Liu and Li, 2017, Yu, Berg, and Verhoef, 2019, and Li, Huang, and Shang, 2020, however, these models do not directly incorporate the matching of drivers and passengers.

In this chapter, we use a bi-level optimization approach to model the two-way causality between carpooling and congestion. The first level considers the optimal matching problem. We determine the matching that minimizes the total costs, which comprises detour, delay, and inconvenience costs. Congestion is incorporated in the second level through a dynamic bottleneck model. Equilibrium departure times are determined given the optimal matching. Using an iterative process, we can solve the congested matching model to evaluate the effect of congestion on carpooling matching.

The remainder of this chapter is organized as follows. The framework and corresponding theoretical results are described in Section 3.2. The framework is extended with bottleneck congestion in Section 3.3. Section 3.4 discusses the simulation results and in Section 3.5 the chapter is concluded. Most formal proofs of the theorems are relegated to the Appendix.

3.2 Matching without congestion

We consider the morning commute of passengers and drivers. In this case, we assume that the origins of all agents are distributed on Hotelling's line $[0,1]$, while their destination is at the Central Business District (CBD), which is located at 1 and is, therefore, the same for all agents. An example of this with one passenger and one driver is depicted in Figure 3.1. During the evening commute, this problem is reversed. In this case, the origin is the same for all agents, while their destinations

are distributed on Hotelling's line. It is important to note that the morning and evening commutes are similar but not identical since in the evening there is an optimal desired departure time from the origin, rather than a desired arrival time at the destination.

Let the location of passenger j be denoted as y_j and the location of driver i as x_i . Let $\mathcal{C}(i, j)$ denote the (full) cost of a match between driver i and passenger j . A driver always picks up a passenger before driving to the destination. This means that if the location of the passenger is before that of the driver (i.e. driver is closest to the CBD), the matching distance is twice the distance between the customer and the driver. On the contrary, if the passenger is located closest to the CBD, there is no additional matching distance. We let α denote the value of time¹ of drivers. The cost of matching for a driver located at x_i and a passenger located at y_j is defined as follows:

$$\mathcal{C}(i, j) = 2\alpha \max(x_i - y_j, 0) \quad (3.1)$$

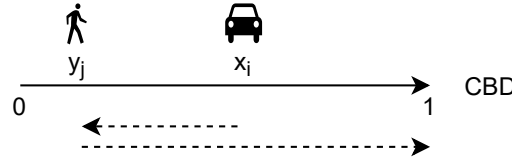


Figure 3.1: Example of a link with one passenger located further away from the CBD than the driver

In our framework, the set of drivers and passengers is assumed to be fixed and known beforehand. This can be interpreted as identifying your role (i.e. passenger or driver) when signing up to a carpooling app. Thereby, we assume that a central operator determines the matching and this can be enforced to the participants. This assumption is reasonable as drivers and passengers are generally unaware of their alternatives in such a carpooling framework and they can inform the operator of their trip information (origin and desired arrival at destination) in advance. We start by assuming the matching costs are limited to the costs for making a detour (A1). This assumption is reasonable in case all individuals have identical desired arrival times, and this assumption will be relaxed in later sections.

Assumption A1 (Simple matching): *The (additional) cost of matching driver i to passenger j depends only on the detour driver i makes to pickup passenger j ,*

¹Different values of time can be considered easily, without changing the mathematical nature of the problem. For simplicity, we assume here the same value of time for passengers and for drivers with and without a passenger.

as defined in Equation (3.1).

The remainder of this section is organized as follows. First we discuss the basic model in Section 3.2.1. Then, we extend the model to incorporate unequal numbers of drivers and passengers in Section 3.2.2. We incorporate scheduling delay in Section 3.2.3.

3.2.1 Basic carpooling model: equal number of drivers and passengers

First we assume that the number of passengers and drivers are equal. We can formulate this as a linear assignment problem. Let I denote the set of drivers and J the set of passengers. To ease notation, the following assumption is made without loss of generality:

Assumption A2 (Sorting): *Without loss of generality, all passengers and drivers are sorted from left to right based on their location on the Hotelling line.*

Following this assumption, driver locations satisfy $x_i \leq x_{i+1}$ and passenger locations satisfy $y_j \leq y_{j+1}$. We study here the matching process, which minimizes the total costs (which includes the potential schedule delay costs). This is thus an optimal assignment provided by a regulator under the following definition.

Definition 1 (Optimal matching without congestion): *The optimal matching minimizes the sum of detour cost and inconvenience cost (including scheduling delay costs) for all involved passengers and drivers. The optimal matching is the solution of **P1**.*

We use decision variables a_{ij} , which are equal to 1 if passenger j is matched with driver i . We then formulate this problem as a linear assignment problem (see Burkard and Cella, 1999, Galichon, 2018). We determine the optimal matching, that is the matching minimizing total cost, given that all drivers are matched to a passenger and all passengers are matched to a driver.

$$\mathbf{P1} : \min \sum_{i \in I} \sum_{j \in J} \mathcal{C}(i, j) a_{ij} \quad (3.2a)$$

$$\sum_{i \in I} a_{ij} = 1, \quad \forall j \in J, \quad (3.2b)$$

$$\sum_{j \in J} a_{ij} = 1, \quad \forall i \in I, \quad (3.2c)$$

$$a_{ij} \in \{0, 1\}, \quad \forall i \in I, j \in J. \quad (3.2d)$$

This is a special case of an integer linear program, which has a totally unimodular constraint matrix. We use the definition of Heller, Tompkins, Kuhn, and Tucker, 1957 for the totally unimodular constraint matrix. According to the authors, the constraint matrix can only contain elements $\{-1, 0, 1\}$ and at most two non-zero elements per column. Thereby, the rows can be partitioned over two disjoint sets such that for every column, two elements with equal sign are in different sets and two elements with same sign are in the same set. For our problem, these disjoint sets are equal to I and J . In this special case, the linear programming (LP) relaxation of the problem will provide the optimal solution and therefore the problem can be solved efficiently. Even for a large number of passengers and drivers, the optimal solution can be found in seconds.

Passengers and drivers are only willing to carpool with each other if they jointly benefit from the match. Therefore, a passenger and a driver are only willing to match if their joint benefit is higher than their joint costs. In practice, the costs and benefits can be redistributed between the matched couple, for example through an online platform. We characterize commuters by their potential role in a carpooling match (i.e. drivers and passengers). Definitions and assumptions follow. Drivers can be solo drivers or carpooling drivers. Passengers are carpooling passengers, public transport passengers (in case they do not own a car and are not matched) or self-driving passengers (in case they own a car and are not matched). In this section, we only consider passengers that do not own a vehicle. Self-driving passengers are considered in Sections 4 and 5. Formally, we define b as the subsidy drivers receive for carpooling and c^T as the fixed opportunity costs passengers experience for taking public transport. As a consequence, a driver and passenger are willing to match if the following condition holds:

$$\mathcal{C}(i, j) \leq b + c^T \quad (3.3)$$

We add a set of dummy passengers and dummy drivers to the sets of passengers and drivers, such that every driver is able to match to a dummy passenger and every passenger is able to match to a dummy driver. We define a dummy driver/passenger as follows:

Definition 2 (Dummy driver/passenger): *A dummy driver or a dummy passenger provides an alternative matching option for passengers or drivers, respectively. An individual matched to a dummy driver/passenger is travelling alone and therefore not carpooling.*

The matching cost to a dummy passenger is equal to b (lost subsidy) and the matching cost to a dummy driver is equal to c^T (cost of public transport). Matching a dummy driver to a dummy passenger is irrelevant and the corresponding cost

is therefore set to zero. Clearly, it will never be optimal to match a driver and a passenger if $\mathcal{C}(i, j) > b + c^T$, i.e. if Constraint (3.3) is violated. The optimal matching of drivers and passengers, given that they only match if they jointly benefit from the match, can then be determined using formulation P1. In general, it is possible for an individual to have multiple alternatives. This can then be incorporated easily by adding multiple dummy variables.

The optimal solution to the matching problem is not necessarily unique as multiple solutions can lead to the same optimal objective value. A trivial example is the case where all drivers are located before all passengers, in which every possible matching has an objective value of 0 as all cars are the same from the passengers' point of view.

Under assumption A1 and given that it is optimal to match all drivers and passengers, the optimal matching of drivers and passengers is formulated in the following theorem. A formal proof can be found in Appendix B.

Theorem 3. *Let the number of drivers be equal to the number of passengers. Under Assumption A1 and given that at optimality all drivers and passengers match, matching the i^{th} driver and the i^{th} passenger is always among the set of optimal matchings.*

Intuitively, drivers and passengers are matched based on the order in which they are ranked from furthest to closest to the CBD. We emphasize that this solution is not necessarily unique in the sense that two passengers can be switched without changing the value of the objective.

A matching of a similar structure is optimal if not all passengers and drivers are matched at optimality. Intuitively, the most costly passengers and drivers remain unmatched and the remaining passengers and drivers are matched in agreement with Theorem 3. This is formally described in Theorem 4. The proof is relegated to the Appendix.

Theorem 4. *Let the number of drivers be equal to the number of passengers and let matching cost be defined according to assumption A1. Assume that at optimality k drivers and k passengers are not matched. The matching where the k left-most passengers remain unmatched, the k right-most drivers remain unmatched and the remaining passengers and drivers are matched in sequence according to Theorem 2, is always among the set of optimal matches.*

An example of Theorem 4 is illustrated in Figure 3.2. The initial matching is given by the black lines. Imagine the detour outweighs the cost of the alternative, such that this matching is suboptimal. Specifically, here, the red-dotted match violates Equation (3.3), which means that this driver and passenger would not be willing to

comply with this match. As a consequence, the most costly passenger and most costly driver, both marked in red, are removed. The remaining passengers and drivers are matched according to Theorem 3, yielding the matching indicated by blue lines. This matching is optimal since no driver makes a detour.

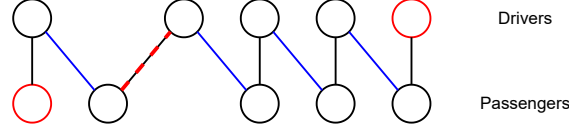


Figure 3.2: Example of Theorem 3

The optimal matching may not necessarily be stable. In a stable matching, there is no other match where both the driver and the passenger prefer each other over their current match. The advantage of such a matching is that no one is able to improve herself by deviating from the current match. The concept of stability in matching is borrowed from marriage economics (Browning, Chiappori, and Weiss, 2014). However, in a carpooling framework, as we consider, it is reasonable to assume that individuals (both drivers and passengers) are unaware of their alternative matching possibilities. In such a framework, enforcing the optimal matching is therefore intuitive and common in practice. Alternatively, the stable formulation by Wang, Agatz, and Erera, 2018 can be used to replace formulation *P1*. In that case, the matching will be stable, but not necessarily optimal under our definition of optimality.

3.2.2 Unequal number of drivers and passengers

When the number of drivers and passengers are unequal, we can still construct and solve the model discussed in Section 3.2.1 in a similar way. The optimal matching when the number of drivers is unequal to the number of passengers can be characterized as in Theorem 5. Dummy drivers are assigned to the left-most passengers whereas dummy passengers are assigned to the right-most drivers. For the remaining drivers and passengers, the same structure as described in Theorem 4 is observed.

Theorem 5. *Consider a set of m drivers and n passengers both ranked from left to right. Under assumption A1, the following match is optimal:*

1. *If $m < n$, the $n - m$ left-most passengers are not matched (i.e. matched to dummy drivers).*
2. *If $m > n$, the $m - n$ right-most drivers are not matched (i.e. matched to dummy passengers).*
3. *The remaining $\min(m, n)$ drivers and passengers are matched according to Theorem 3.*

As for Theorem 4, we observe that the most costly drivers or passengers are removed if the total number of passengers and drivers is unequal. The most costly drivers and passengers are those that are most likely to cause a detour. For drivers, these are located on the right of Hotelling's line, for passengers, these are located on the left.

3.2.3 Scheduling delay

For passengers or drivers, matching costs are often not solely based on the additional distance a driver has to drive to perform the pickup. Additional inconveniences may be experienced by drivers and passengers. Here, we consider a specific type of inconvenience cost: schedule delay cost. The approach below can be easily extended to any arbitrary pattern of inconvenience costs. Until now, we assumed drivers and passengers had no scheduling delay preferences. However, in many scenarios the desired arrival time is different for every individual and individuals are penalized for early and late arrivals. Therefore, we present a model that incorporates time preferences and scheduling delay. Every individual i has a desired arrival time τ_i . In this case, drivers and passengers also need to agree on an arrival time when they are matched. The matching costs are therefore a trade-off between travel time and scheduling delay as first introduced by Vickrey, 1969. If travel time is independent of departure time, the total matching cost including scheduling delay penalties is:

$$\mathcal{C}(i, j, t) = 2\alpha \max(x_i - y_j, 0) + \beta((\tau_i - t)^+ + (\tau_j - t)^+) + \gamma((t - \tau_i)^+ + (t - \tau_j)^+), \quad (3.4)$$

where t is the agreed arrival time, β is the earliness penalty and γ is the lateness penalty. We assume $\gamma \geq \beta$ (lateness is weighted higher than earliness), consistent with the empirical literature. For a given match (i, j) the optimal joint arrival time t^* can be determined according to the following theorem (of which the proof can be found in the appendix) and illustrated graphically as in Figure 3.3. We have:

Theorem 6. *Consider linear scheduling delay with $\gamma \geq \beta$. Driver i is matched to passenger j with desired arrival times τ_i and τ_j . The optimal arrival time t^* that minimizes the total earliness and lateness penalty as given in Equation (3.4) is given as $t^* = \min(\tau_i, \tau_j)$. In this case, the reduced-form cost of matching driver i to passenger j is $\tilde{\mathcal{C}}(i, j) = 2\alpha \max(x_i - y_j, 0) + \beta|\tau_j - \tau_i|$.*

We emphasize that the reduced matching cost is independent of t^* . Using this property, the problem can be decomposed such that the optimal t^* can be determined for every potential match first and the optimal matching can be determined thereafter using the reduced matching cost. If earliness and lateness are not computed as piece-wise linear functions as in Equation (3.4), Theorem 5.1 does not necessarily hold. However, for many functions a similar closed form solution can be obtained. For any pair of functions satisfying the non-increasing and non-decreasing definition of earliness and lateness, respectively, the following theorem holds.

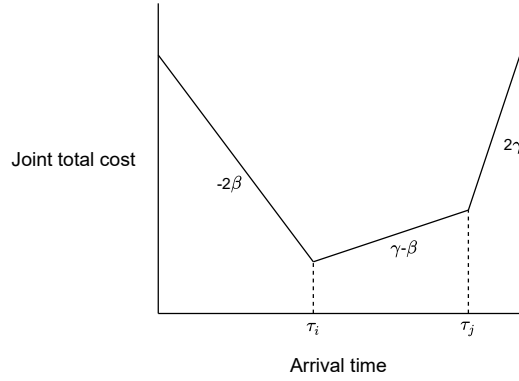


Figure 3.3: Scheduling delay cost for a given match

Theorem 7. *Consider scheduling delay penalties where earliness and lateness are non-increasing and non-decreasing functions of time, respectively. Driver i is matched to passenger j with desired arrival times τ_i and τ_j and $\tau_i \leq \tau_j$, without loss of generality. The optimal arrival time t^* that minimizes the total earliness and lateness is in the closed bounded interval $[\tau_i, \tau_j]$.*

Note that this only holds if travel time is independent of the departure time. In the presence of congestion, travel time and the optimal departure time are not independent and therefore Theorem 5.1 and 5.2 do not necessarily hold.

3.3 Matching in the presence of bottleneck congestion

During the morning commute, the access roads to the CBD are often heavily congested. Travellers cannot always arrive on time at their destination and they may experience both scheduling and travel time delays. The effect of congestion in road networks on departure time decisions is a well-studied problem (e.g. Arnott, Palma, and Lindsey, 1993 and Small, Verhoef, and Lindsey, 2007). Here, we consider a basic Vickrey, 1969 model of road congestion. We incorporate bottleneck congestion and scheduling delay in the matching decisions of the carpooling operator and the departure time decisions of the drivers. Bottleneck congestion is implemented in a similar way as described by Arnott, Palma, and Lindsey, 1993. We assume the road is uncongested except at a single bottleneck at the entrance of the CBD. Here, at most s cars can pass per unit of time, suggesting that a queue is formed if this capacity is exceeded. The queue is served according to a first-come, first-served (FIFO) policy. Similar to Arnott, Palma, and Lindsey, 1993, travel time from home to work is defined as $T(t) = T^f + T^v(t)$, where T^f is fixed travel time including a possible detour caused by matching, T^v is variable travel time and t is the departure

time. Variable travel time depends on the queue length at the time of arrival at the bottleneck, $Q(t')$. A driver's (or carpooler's) queueing time is then equal to:

$$T^v(t') = \frac{Q(t')}{s}. \quad (3.5)$$

The length of the queue can be determined using the cumulative number of arrivals at the bottleneck ($r(t)$) and the departure rate according to the capacity. Let \hat{t} be the last time the queue was empty. The queue length at time $t' > \hat{t}$ is given by:

$$Q(t') = \int_{\hat{t}}^{t'} r(u)du - s(t' - \hat{t}). \quad (3.6)$$

Matching costs in our problem now include possible delay at the bottleneck, detour to pickup a passenger, earliness and lateness. The relationship between travel time, earliness and lateness is similar to that described in Section 3.2.3.

Note that bottleneck congestion does not incorporate congestion on the corridor itself. However, as during the morning commute the traffic leaving the CBD is usually very limited, congestion during the pickup of passengers is unlikely. Congestion on the corridor towards the CBD tends to increase as the bottleneck is approached, due to the increased demand for road space. Therefore, the highest level of congestion is observed at the entrance of the CBD, which is incorporated in our bottleneck. More complex city structures are beyond the scope of this work, but clearly a challenging future direction.

The joint problem of matching and departure time choices is solved in a sequential and iterative way. The departure time choices in equilibrium are determined using an equivalent optimization problem for the bottleneck model, as described by Iryo and Yoshii, 2007. Strictly speaking, Iryo and Yoshii, 2007 determine the equivalent equilibrium arrival times rather than departure times. Therefore, from here onwards we refer to equilibrium arrival times instead. While various heuristics have been developed to solve this sub-problem, we chose this method as it is fast and exact. Thereby, it allows to compute the scheduling delay costs as well as delay at the bottleneck efficiently through a single optimization problem and its dual. By solving an LP, the equilibrium arrival times and corresponding cost functions can be determined. Variable z_{ti} describes the number of drivers in class $i \in I$ (here, all drivers and driver-passenger pairs are treated as separate classes) that choose time $t \in T$ to leave the bottleneck (and also enter it in case there is no congestion). We emphasize that for the congestion model, time is discrete and not continuous. Classes are defined as the sets of drivers with equal desired arrival times. In the case of carpooling, we use the matched couples to identify a class. The total number of drivers in class i is defined as d_i and the capacity at the bottleneck is given as s . The non-bottleneck costs (i.e. scheduling delay costs) of a driver (or match) in class

i that choose to leave the bottleneck at time t is given as p_{ti} . Non-bottleneck costs p_{ti} are defined as the sum of earliness and lateness penalties for both the driver and the passenger, scaled by the value of time α , such that the dual variables directly give the delay. In the case of carpooling, we treat a couple as a single entity and therefore we take the average of their non-bottleneck costs.² This yields the following definition of non-bottleneck costs for a driver i arriving at time t . Note that, as the matching is fixed at this stage, every matched driver has a known passenger j .

$$p_{ti} = \begin{cases} \frac{\beta}{\alpha} \left(0.5(\tau_i - t)^+ + 0.5(\tau_j - t)^+ \right) + \frac{\gamma}{\alpha} \left(0.5(t - \tau_i)^+ + 0.5(t - \tau_j)^+ \right) & \text{if driver } i \text{ is matched to passenger } j \\ \frac{\beta}{\alpha} (\tau_i - t)^+ + \frac{\gamma}{\alpha} (t - \tau_i)^+ & \text{if driver } i \text{ drives alone} \end{cases} \quad (3.7)$$

The formulation is given as follows:

$$\mathbf{P2:} \quad \min \sum_{t \in T} \sum_{i \in I} p_{ti} z_{ti} \quad (3.8a)$$

$$\sum_{i \in I} z_{ti} \leq s \quad \forall t \in T \quad (3.8b)$$

$$\sum_{t \in T} z_{ti} = d_i \quad \forall i \in I \quad (3.8c)$$

$$z_{ti} \geq 0 \quad \forall t \in T, i \in I \quad (3.8d)$$

The objective is to minimize the total scheduling delay costs given that the capacity is satisfied at every time interval and every driver (or match) chooses a time interval. According to Iryo and Yoshii, 2007, the solution to this problem is equivalent to the equilibrium arrival time choices of all drivers if the FIFO constraint is satisfied (FIFO is not enforced by the formulation proposed by Iryo and Yoshii, 2007 and has to be checked separately).

Iryo and Yoshii, 2007 define the dual formulation, to obtain the delay w_t for a given arrival time $t \in T$. We use dual variables w_t and θ_i to solve the following formulation:

$$\mathbf{P3:} \quad \max \sum_{i \in I} d_i \theta_i - \sum_{t \in T} s w_t \quad (3.9a)$$

$$p_{ti} + w_t \geq \theta_i \quad \forall t \in T, i \in I \quad (3.9b)$$

$$w_t \geq 0 \quad \forall t \in T \quad (3.9c)$$

By solving the dual formulation, we can approximate the delay in the equilibrium state. The delay in equilibrium for every discrete time step t is given by w_t . As we consider a matched couple as a single agent, we are able to incorporate scenarios where a part of the drivers are driving alone, while another part is carpooling.

²Equal coefficients ensures optimality for driver and passenger if we assume costs can be redistributed between driver and passenger. The proof hereof is straightforward and left to the reader.

We use an iterative approach to determine the optimal matching decisions in equilibrium. The reason for this is that the joint problem (optimal matching and arrival time choices in equilibrium) is too complex to solve exactly in a one-step process. The iterative procedure is summarized in Algorithm 1. The algorithm is initialized with the uncongested travel time (no delay at the bottleneck). The optimal matching and consequently the equilibrium arrival times given this matching are determined using the aforementioned optimization problem. The delay is updated using the estimated delay from the dual function as a weighted moving average. The predicted delay is updated with the delay from last iteration using a fraction λ^k . After every iteration, λ^k is decreased by multiplying it by a constant δ until λ_{min} is reached. This represents the increasing confidence in the estimated delay as the number of iterations increases. Delay estimates are updated as follows where \hat{W}^k are the predicted delays for iteration k and W^k are the experienced delays for iteration k :

$$\hat{W}^{k+1} = \lambda^k W^k + (1 - \lambda^k) \hat{W}^k \quad (3.10a)$$

$$\lambda^{k+1} = \max(\delta \lambda^k, \lambda_{min}) \quad (3.10b)$$

At convergence, arrival times are in equilibrium following the optimization problem defined by Iryo and Yoshii, 2007 and the matching decisions are optimal given the equilibrium state. The generalized matching costs including bottleneck congestion and tardiness penalties are now as follows for driver i and/or passenger j departing at time t :

$$c(i, j, t) = \begin{cases} \alpha(D(i, j) + W(i, j, t)) + \beta E(i, j, t) + \gamma L(i, j, t), & \text{if regular driver } i \text{ is matched to regular passenger } j \quad (3.11a) \\ b + \alpha W(i, j, t) + \beta E(i, j, t) + \gamma L(i, j, t), & \text{if regular driver } i \text{ is matched to dummy passenger } j \quad (3.11b) \\ c^T, & \text{if dummy driver } i \text{ is matched to a regular passenger } j \quad (3.11c) \\ 0, & \text{if dummy driver } i \text{ is matched to a dummy passenger } j \quad (3.11d) \end{cases}$$

Here, D represents the detour driver i makes to pick up passenger j , W is the delay at the bottleneck, E and L are earliness and lateness penalties respectively. Bottleneck delay, earliness and lateness are experienced by both the driver and the passenger whereas the detour is only experienced by the driver. Equation (3.11a) is the congested variant of (3.1) and (3.4), whereas the other equations represent the matches to dummy variables. If a driver is matched to a dummy passenger, he/she loses his/her subsidy b . Thereby, he/she may still suffer from delay and scheduling delay at the bottleneck, but he/she does not need to incorporate the time preferences of a passenger. For the passenger, the cost of matching to a dummy driver is equal to c^T . In this case, it is assumed the passenger does not suffer from any delay or scheduling delay when using public transport. So far, we have considered that passengers are either carpooling passengers or public transport passengers. Alternatively, we can consider the case where all passengers also own a vehicle and are self-driving if they are not carpooling. In this case, the cost of matching a

passenger to a dummy driver has the same expression as that of matching a driver to a dummy passenger, except for the lost subsidy (which is only added for drivers). Thereby, a passenger incurs a general car cost, such as the cost of gasoline (c_g). The cost of gasoline is assumed to be proportional to the distance from the CBD, but independent of any congestion encountered. Considering this, if passengers own a vehicle, their cost of matching to a dummy driver is defined as follows:

$$\mathcal{C}(i, j, t) = c_g(1 - x_j) + \alpha W(i, j, t) + \beta E(i, j, t) + \gamma L(i, j, t) \quad (3.12)$$

where a dummy driver i is matched to a regular passenger j . Of course, if passengers also drive, they should be accounted for in the bottleneck. We emphasize that Equation (3.12) replaces Equation (3.11c) in case passengers are self-driving instead of using public transport, and the other equations remain unchanged.

To attain convergence, the matching only changes if the improvement compared to the previous matching is higher than $\epsilon\%$. Thereby, we only allow κ matches that we selected in the previous iteration to remain unchosen during the next iteration. This additional restriction can be incorporated easily in Problem P1. We enforce this restriction for the first η iterations.

Algorithm 1: Iterative Matching Approach

- 1 Initialize the waiting time at every time interval to the uncongested waiting time
 - 2 **while** *Stopping criterion is not met* **do**
 - 3 Determine the optimal matching using formulation P1
 - 4 Given the matching, determine the equilibrium arrival times using the equivalent optimization problem defined by Iryo and Yoshii, 2007 using formulation P2
 - 5 Given the matching and the equilibrium arrival times, determine the delay at every time interval using the dual formulation P3
 - 6 Update the matching costs for every potential match given the estimated delays
 - 7 Gather statistics on travel time, delay, earliness and lateness based on current matching and actual waiting time.
 - 8 **end**
-

In accordance with the updated matching costs, we redefine the notion of optimal matching as follows:

Definition 3 (Optimal matching with congestion): *The optimal matching minimizes the sum of detour costs, inconvenience costs (including scheduling delay costs) and delay at the bottleneck for all involved passengers and drivers.*

The updated definition of optimal matching requires to solve a dynamic bottleneck congestion problem. The optimal matching in equilibrium is such that the central operator does not want to change the obtained matching and the individuals do not want to change their arrival times. As such, delay at the bottleneck and the arrival time decisions are in equilibrium and therefore the corresponding optimal matching does not change.

3.3.1 Equilibrium analysis

An analytical solution can be obtained for the case where the desired arrival time t^* is equal for every individual. This solution can be identified using the following theorem:

Theorem 8. *Consider the dynamic carpooling model with n passengers and m drivers, each with $\alpha - \beta - \gamma$ (scheduling) delay penalties and both owning vehicles. If all users have the same desired arrival time t^* and all users prefer carpooling over their alternative, then the equilibrium arrival times can be determined using $\max(n, m)$ non-carpooling individuals as described by Arnott, Palma, and Lindsey, 1993.*

Proof. Arrival time is determined as a couple agreement between driver and passenger. Note that this is independent of the travel time before the bottleneck (including possible detour for pickup) as this is fixed for every arrival time. Therefore, the cost function to determine the arrival time only includes waiting time and scheduling delay. As all desired arrival times are equal, scheduling delay is independent of the matching. The matching and arrival time decisions are therefore independent. Given that every individual has equal desired arrival time, the match can be considered to be a single agent. If everyone prefers to carpool, there will be a total of $\min(n, m)$ matches. The remaining drivers or passengers will drive solo as they all own a vehicle, so the total number of vehicles on the road will be equal to $\max(n, m)$. Using this, the arrival times in equilibrium are similar to those of a non-carpooling setting with $\max(n, m)$ individuals and can be derived analytically as described by Arnott, Palma, and Lindsey, 1993. \square

This result does not necessarily hold if the desired arrival times are distributed. In this case, the cost functions are not equal and therefore driver and passenger may wish to respond differently to congestion. Any other kind of cost can be included, as long as the desired arrival time is equal. If not all users prefer carpooling over their alternative, this is not necessarily true. In this case, some matches may not be optimal under some levels of congestion, while they are optimal under other levels. The number of matches may therefore depend on the level of congestion, thereby losing the independence between the matching model and the bottleneck model. This is discussed in more detail in the next section. Liu and Li, 2017 analyze the

bottleneck model in the homogeneous case (with no spatial distributions of origins, nor of distribution of t^*). They consider three types of commuters: solo drivers, ridesharing drivers and ridesharing riders. Under those assumptions, they show that at equilibrium, with time based and distance based ridership compensations, solo drivers can drive at the middle of the peak or on the wing of the peak. As discussed below, heterogeneity (location and desired arrival time) matter.

Let us elaborate on the difference between homogeneous and heterogeneous individuals with a numerical example. Algorithm 1 is used to obtain the delay at convergence for various t^* profiles. We consider a profile where every individual has identical desired arrival time t^* and a profile where desired arrival times are pseudo-randomly generated. We compare the well known scenario where every individual drives alone to the scenario where everybody carpools. Figure 3.4a displays the delay for all time intervals, with arrival time on the horizontal axis and delay on the vertical axis. The peak delay is at t^* , which is the same for all individuals. We observe that, in accordance with the aforementioned theorem, the shape of delay for carpooling mimics the shape of delay for no-carpooling with $\max(m, n)$ drivers. The slopes are equal for both curves and are equal to $\frac{\beta}{\alpha}$ before t^* and $-\frac{\gamma}{\alpha}$ after t^* , in agreement with Yu, Berg, and Verhoef, 2019. The beginning and end of the rush hour, t_q and $t_{q'}$ respectively, are also equal to those defined in Arnott, Palma, and Lindsey, 1993 and can be defined as follows, where N is the number of vehicles on the road:

$$t_q = t^* - \frac{\gamma}{\beta + \gamma} \frac{N}{s} \quad (3.13a)$$

$$t_{q'} = t^* + \frac{\beta}{\beta + \gamma} \frac{N}{s} \quad (3.13b)$$

Figures 3.4b and 3.4c display the delay at the bottleneck if t^* is not identical. We evaluate the queue on a time horizon between 0 and 20, which is divided into 1000 intervals. The desired arrival times are pseudo-randomly drawn from a uniform distribution³ between $t_s^* = 8$ and $t_e^* = 12$. The peak of congestion is observed at $\hat{t}^* = \frac{\beta}{\beta + \gamma} t_s^* + \frac{\gamma}{\beta + \gamma} t_e^*$, which is at 11.2. This result is in agreement with those defined in Lindsey, Palma, and Silva, 2019. We also observe that the slopes are similar to those when every individual has an identical t^* and the start and the end of the rush hour can be approximated using (3.13a) and (3.13b).

When matching exists, the uniform distribution of the desired arrival times are lost, as they are approximately pooled between the driver and the passenger. Therefore, the shape of the delay curve is lost and no theoretical value for the peak congestion time can be used. However, as the arrival times of a matched couple are determined in a systematic way, we can still identify some descriptive properties. According to

³Desired arrival times are selected such that they are exactly uniformly distributed on the interval and are therefore not truly random

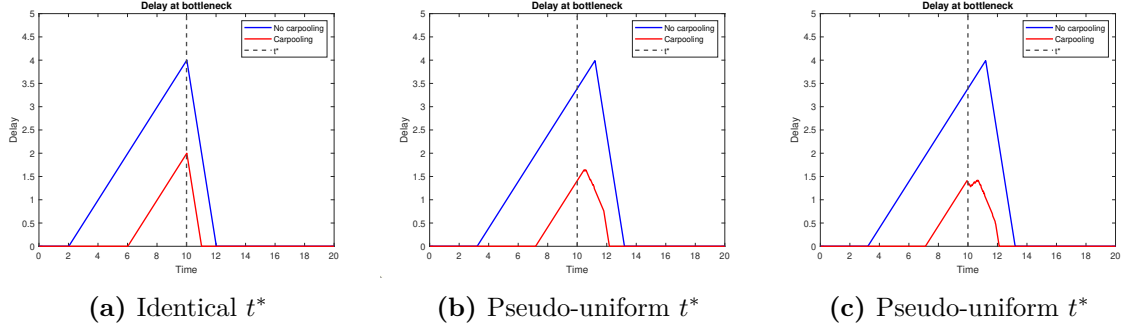


Figure 3.4: Delay at bottleneck

Theorem 5.1, in the absence of congestion, the optimal arrival time of a matched couple is the minimum of their two desired arrival times. We observed that in the presence of congestion, a matched couple still has a tendency towards the earliest arrival time, in agreement with this theorem. Following this, as displayed in Figure 3.4b, the peak of congestion is usually slightly earlier than \hat{t}^* . If the desired arrival times of the matched couples are close to a uniform distribution, there will be a single peak of congestion, as displayed in Figure 3.4b. However, if the uniformity of desired arrival times is lost after matching, the single peak congestion may also be lost. In that case, we can observe multiple peaks in the delay curve as displayed in Figure 3.4c. We emphasize that the delay curves in Figure 3.4a and those without carpooling (blue) in Figure 3.4b and 3.4c are theoretical. The delay curves with carpooling (red) in Figure 3.4b and 3.4c are simulated. A detailed analysis of delay and tardiness penalties is included in Section 3.4.2.

3.4 Numerical results

In this section we provide numerical simulation results based on the methodology described in the previous sections. We first evaluate the theoretical results without congestion. Thereafter we evaluate the optimal matching under congestion and perform extensive sensitivity analysis in Section 3.4.2.

3.4.1 Optimal matching without congestion

As expected, simulation results have shown that in the absence of congestion, the optimal matching is in agreement with Theorems 1 - 5. While this is theoretically proven, it also allows to test that our simulations provide intuitive results. Results have also identified that, especially in large instances, the optimal matching is not necessarily unique. Consider the following example where driver i_1 and driver i_2 are both located earlier on Hotelling's line than passenger j_1 and j_2 . Then, under Assumption A1, the relation in Equation (3.14) holds.

$$\mathcal{C}(i_1, j_1) + \mathcal{C}(i_2, j_2) = \mathcal{C}(i_1, j_2) + \mathcal{C}(i_2, j_1) = 0. \quad (3.14)$$

This indicates that the optimal matching is not necessarily unique, as passengers j_1 and j_2 can be interchanged without changing the objective value.

3.4.2 Optimal matching under bottleneck congestion

In this section we evaluate the optimal matching under bottleneck congestion for various levels of carpooling and bottleneck capacity. We evaluate the convergence of our iterative algorithm in Section 3.4.2. Thereafter, we compare macroscopic statistics such as the total number of matches, average detour, average delay and scheduling delay penalties for various system configurations in Section 3.4.2. Finally, in Section 3.4.2, we provide a more detailed microscopic analysis of the difference between carpoolers and solo-drivers when they face bottleneck congestion.

Convergence of the iterative algorithm

We first evaluate the convergence of our algorithm. Given that we use an exact optimization problem to find the equilibrium arrival times, we only need to evaluate the convergence of the matching decisions. In our analysis, we consider a total of 200 individuals, that can be either drivers or passengers. The length of the corridor is set equal to 10 unit lengths. We set $\lambda^0 = 1.0$, $\lambda_{min} = 0.1$ and $\delta = 0.95$ (for weighted moving average estimates in Equation (15)). Desired arrival times t^* are drawn from a uniform distribution between 8 and 12 time units. We consider the complete horizon between 0 and 20, which is divided into 100 discrete time intervals. The value of time and schedule delay parameters α , β , γ are chosen as 2, 1 and 4, respectively.

We consider the following parameter settings. The maximum number of matches that can be changed κ is equal to 20 for the first $\eta = 30$ iterations, while we consider a total of 50 iterations. The minimal expected matching improvement ϵ is equal to 0.1%. The bottleneck capacity is set to 4 per time interval, while the opportunity cost $b + c^T$ is equal to 3. The initial matching is randomly generated to identify multiple starting points. The iterative process is then repeated 10 times. The convergence result are depicted in Figure 3.5. The first graph displays the actual number of matches. We observe that this number fluctuates during the first iterations but stabilizes after 15 to 30 iterations. This is in line with the expected improvement, which approaches zero after 15 iterations. Although the exact number of matches may differ by at most 3 between different runs (ranging between 68 and 71 in Fig. 5a), we observe that the difference in the average cost per individual is negligible. This can be explained by the criterion that the total cost has to be improved by at least 0.1% to accept the new matching. In addition to this, we

have shown that multiple matching may lead to (roughly) the same total costs. To achieve convergence for other system configurations, the specific parameters $\kappa, \eta, \lambda, \delta$ and ϵ may have to be adjusted. When passengers also drive, the level of congestion depends on the number of matches, which may lead to fluctuations. This is related to the myopic nature of the operator's matching decisions, in the sense that the operator does not incorporate the effect of his/her decisions on congestion. To avoid this, the operator could make decisions in a more strategic way, by approximating the congestion level from the number of matches he/she imposes. The convergence of the algorithm and the equilibrium solution when the number of road users is endogenous has to be studied in more detail.

Although the costs are extremely similar across different runs, the exact matchings are not. Over the 10 different runs with on average approximately 70 matches, a total of 250 matches are used in the optimal solution out of which only 50 are used in at least half of the runs. This is explained by the highly similar costs of different matches, similar to the uncongested case in Section 3.4.1. Although the exact matches are quite different, 97% of the passengers either always or never carpool and 87% of the drivers either always carpool or never carpool⁴.

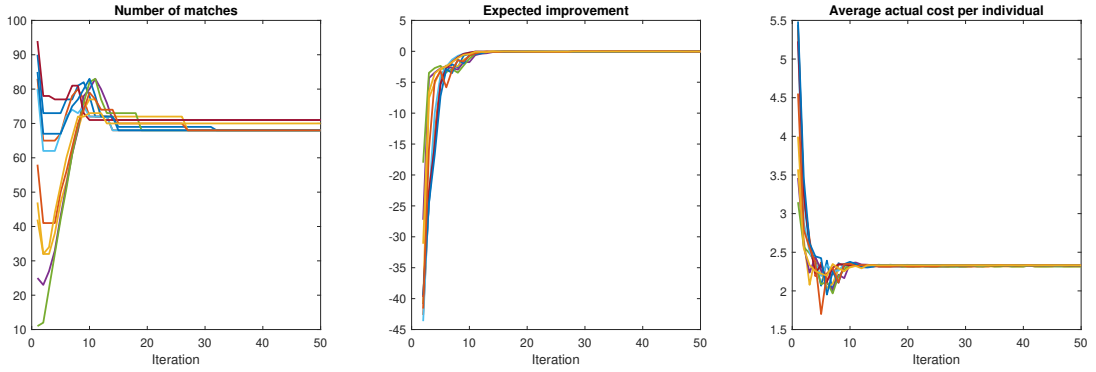


Figure 3.5: Convergence of matching decisions

Macroscopic analysis

The success of carpooling services significantly depends on the quality of the pooling compared to other travel alternatives. In principle, commuters decide to pool if there are savings in their total cost. As the problem described has a large number of parameters to perform sensitivity analysis, our objective is to identify representatives scenarios to shed intuition for critical decisions. We are investigating how the bottleneck capacity, the price of gas and level of subsidy influence the

⁴Note that these numbers are dependent on the chosen simulation settings.

number of matches and different costs associated⁵. With respect to mode choice of passengers, we consider two different options:

Option A: Passengers do not own a car and are either carpooling passengers or public transport passengers.

Option B: Passengers own a car and are either carpooling passengers or self-driving passengers.

We first evaluate the effect of subsidies. Figure 3.6 and 3.7 display the number of matches and cost of scheduling delay, delay, detour and public transport for varying subsidies, while the other parameters are fixed. Figure 3.6 displays option A and Figure 3.7 displays option B. In Figure 3.6, the number of matches increases with the subsidies, as expected. With increasing subsidies to carpoolers, carpooling becomes a more attractive alternative to public transport, thereby attracting more passengers. Average scheduling delay penalties increase, as it is assumed passengers do not experience scheduling delay when using public transport. Overall, it is clear that total cost increase with subsidies, suggesting that when passengers take public transport as an alternative, carpooling subsidies do not have a positive social effect, as expected. This is in contrast with the effect illustrated in Figure 3.7 when passengers also own a car. In that case, by subsidizing carpooling the number of matches increases slightly, thereby reducing the number of cars on the road and therefore reducing scheduling delay and delay penalties. Here, it is clear that subsidizing carpooling does have a positive effect on total cost, due to the positive externality carpooling has by reducing the number of cars that pass the bottleneck. In reality, passengers are likely to be a mix of carowners and non-carowners. An important topic of further research is to use targeted subsidies to those individuals that have the largest positive effect on welfare.

In line with Theorem 6 in the absence of congestion, carpooling induces an increase in earliness. This suggests that even in the presence of congestion, carpoolers have a tendency to depart earlier rather than later. The number of matches first shows a steep increase, then it increases much slower. The reason for this is that the majority of the individuals can be convinced to carpool with relatively small subsidies. Following the intuition of Theorem 5, the most costly matches are left out. For it to be beneficial for the operator to match those individuals, subsidies need to increase substantially, explaining the slower increase in the number of matching for $b > 2$. Next we evaluate the effect of bottleneck capacity on the same statistics, with no subsidy ($b = 0$). When passengers use public transport as an alternative (Option A) in Figure 3.8, the number of matches increases with bottleneck capacity. The reason for this is that congestion decreases, making carpooling more attractive compared

⁵Emissions (in particular CO_2) are roughly proportional to fuel consumption. They are omitted in the current analysis.

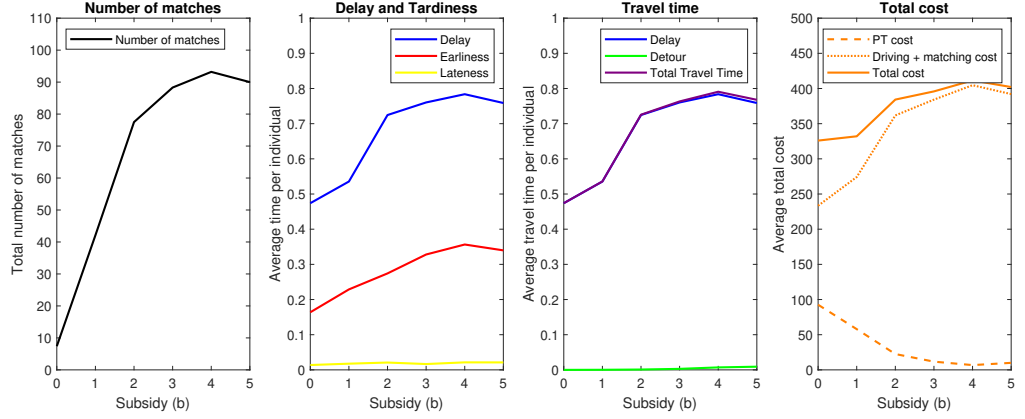


Figure 3.6: Passengers do not own a car and use public transport as an alternative to carpooling. Bottleneck capacity (s) is equal to 4 cars per time interval, public transport cost c^T is fixed at 1 and subsidy varies.

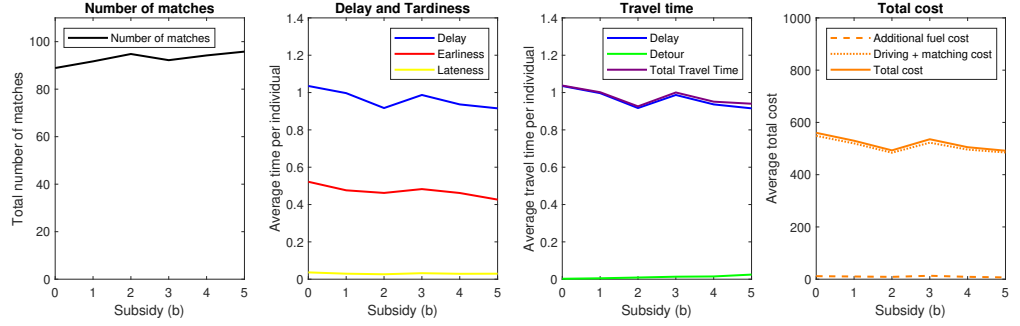


Figure 3.7: Passengers own a car and drive solo as an alternative to carpooling. Bottleneck capacity (s) is equal to 4 cars per time interval, gasoline cost (c_g) is equal to 0.2 per unit of distance and subsidy varies.

to public transport. As an effect, scheduling delay penalties as well as total cost decrease. This is related to the fact that individuals only carpool if it reduces their costs, similar to the matching condition in Equation (3.3). When passengers are self-driving as an alternative to public transport, as in Figure 3.9, the effect of bottleneck capacity on the number of matches is the opposite. If capacity increases, delay at the bottleneck decreases. Delay at the bottleneck gives more flexibility to match with individuals that have different desired arrival times. In equilibrium in the presence of delay, commuters can be indifferent between arriving on time or early/late depending on their desired arrival time t^* . This flexibility disappears as capacity increases and therefore driving solo becomes more attractive for some commuters. We note that as passengers face bottleneck congestion independent of whether they carpool or not, the effect of bottleneck capacity is less substantial when passengers are self-driving (90 to 78 matches) compared to when they use public transport (0 to 85 matches).

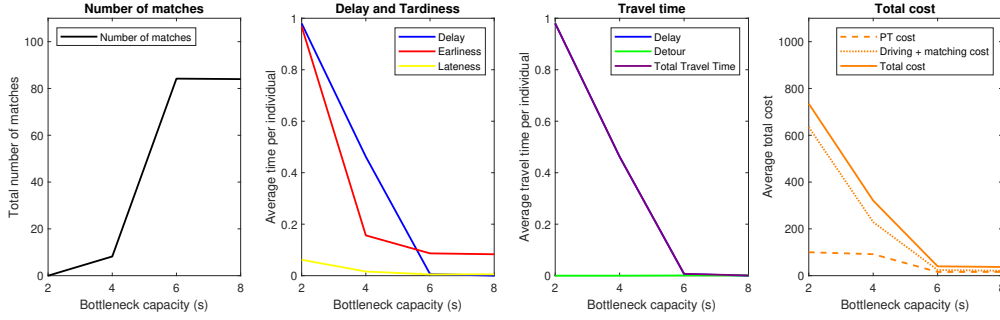


Figure 3.8: Passengers do not own a car and use public transport as an alternative to carpooling. Public transport cost c^T is fixed at 1 and subsidy (b) is fixed at 0. Bottleneck capacity varies between 2 and 8 cars per time interval.

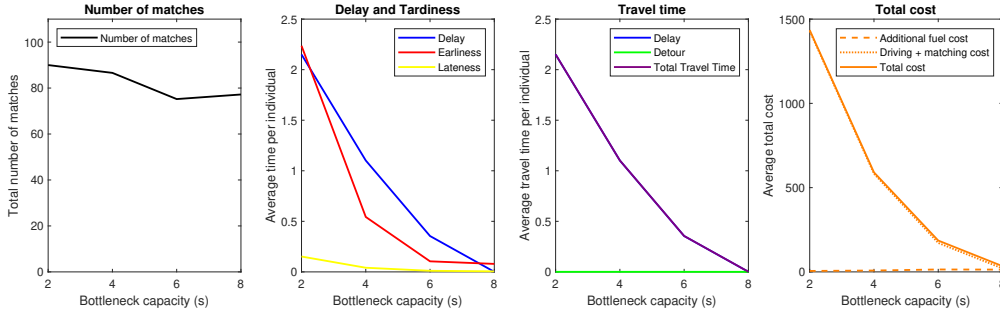


Figure 3.9: Passengers own a car and drive solo as an alternative to carpooling. Gasoline cost (c_g) is equal to 0.2 per unit of distance and subsidy (b) is fixed at 0. Bottleneck capacity varies between 2 and 8 cars per time interval.

A sensitivity analysis on the fuel cost per unit of distance is displayed in Figure 3.10. We note that the fuel cost of 0 is incremented by a value of 10^{-4} , such that carpooling is preferred if all other cost components are equal for driving solo and carpooling. We observe that the number of matches increases with the fuel cost, as the fuel savings of carpooling increase. As a consequence, the total number of cars that have to pass through the bottleneck decreases and therefore congestion decreases. We therefore observe that although fuel costs increase in the right-panel of Figure 3.10, total cost decreases as congestion is reduced. If fuel cost increases even further, it no longer leads to an increase of carpooling participation and therefore leads to increasing total cost.

Lastly, we perform a sensitivity on the composition of the set of drivers and passengers. Specifically, we keep the total number of travellers constant at 200 but vary the number of passengers and drivers. Passengers are self-driving as an alternative to carpooling (Option B) such that the number of cars passing through the bottleneck may vary. This is displayed in Figure 3.11. It is clear that the total cost is minimized when the number of drivers and passengers are equal, as the number of cars on the road is minimized and therefore bottleneck congestion is minimized. We also note that the statistics are approximately symmetric in the

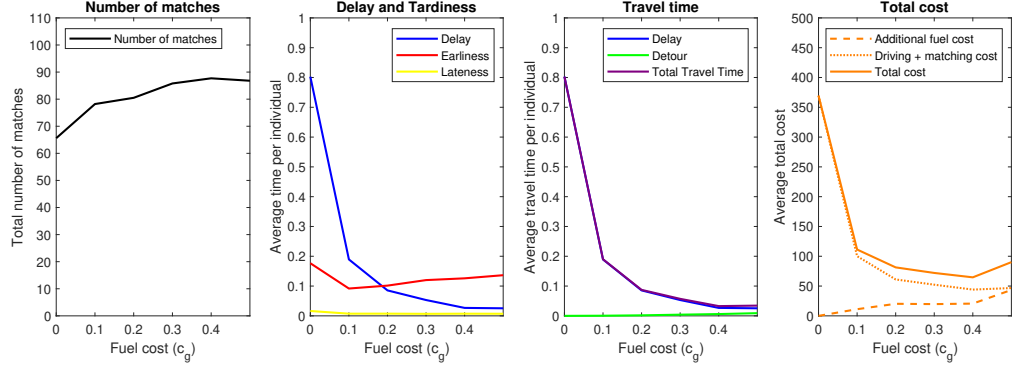


Figure 3.10: Passengers own a car and drive solo as an alternative to carpooling. Bottleneck capacity (s) is equal to 6 cars per time interval and subsidy (b) is fixed at 0. Gasoline costs vary between 0 and 0.5 per unit of distance.

number of drivers and passengers, apart from the fuel cost which is only considered for passengers by construction. When the number of passengers and drivers are unequal (e.g. 60, 80, 120 and 140 passengers), we observe that the number of matches is almost maximized, whereas this is not the case if the number of drivers and passengers are equal. The reason for this is that if there are more drivers or passengers, there exists some flexibility to leave out the commuters that are difficult to match (i.e. can only be matched at high cost) without decreasing the number of matches. We emphasize that when the number of passengers and drivers grows altogether, matching costs in general decrease due to the increased number of matching opportunities.

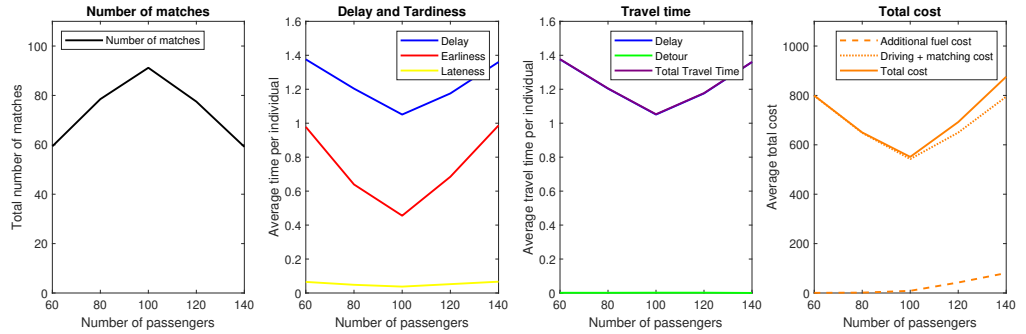


Figure 3.11: Passengers own a car and drive solo as an alternative to carpooling. Bottleneck capacity (s) is equal to 4 cars per time interval, subsidy (b) is fixed at 0 and gasoline cost (c_g) is fixed at 0.2 per unit of distance. The total number of individuals is fixed at 200, but the number of passengers and drivers varies.

Microscopic analysis

Next, we look at the behaviour of the two groups of individuals (i.e. carpoolers and solo drivers) separately. We consider two sets of simulations for options A and B

separately. The simulation settings for both cases are chosen such that there are sufficient solo drivers and carpoolers to compare the two. The graphs are obtained using a total of 50 runs. Note that, when aggregating the two types of commuters, the bottleneck operates at capacity from the moment a queue starts until the queue is completely depleted, in agreement with dynamic congestion equilibrium theory. Also note that the desired arrival times of carpoolers include those of both the driver and the passenger. In the absence of congestion (i.e. for high bottleneck capacities) the distributions of actual and desired arrival times are approximately uniform and similar for carpoolers and non-carpoolers. This serves as a benchmark for our investigation of the effect of bottleneck congestion on carpooling behaviour.

Figure 3.12 displays the actual and desired arrival times for the case where passengers also own a car and are therefore self-driving if they do not carpool (Option B). We observe that carpoolers generally avoid the peak period of the bottleneck, where waiting time is the highest. Delay costs are experienced by both matched individuals, whereas tardiness penalties are split between them. As a consequence, avoiding the bottleneck and moving closer to either one of the desired arrival times is beneficial for a match. Again, we observe that the tendency towards early arrival is higher than the tendency towards late arrival, following the same intuition as in Theorem 6, given that $\gamma \geq \beta$.

Figure 3.13 displays the actual and desired arrival times for the case where passengers do not own a car and use public transport if they are not matched (Option A). We observe a very similar effect to that of the previous case. However, there is one key difference. As the alternative option of passengers does not yield any bottleneck costs (i.e. delay and scheduling delay caused by bottleneck congestion), passengers who are more likely to have high bottleneck costs are less likely to carpool (i.e. those with desired arrival times at the peak congestion period; roughly around time 11 in agreement with the theoretical results without carpooling discussed in the previous section). This makes the distinction between carpoolers and solo-drivers even more clear, but this effect diminishes as the subsidy increases and more carpoolers are attracted.

Carpoolers and non-carpoolers respond differently to congestion. Specifically, carpoolers have a tendency to depart earlier and avoid the peak period of congestion. The optimal matching and the corresponding optimal arrival times are determined accordingly, such that carpoolers arrive in the wings of the travel period, whereas individual drivers arrive somewhat in the middle. The above analysis emphasizes the importance to model congestion in tandem with carpooling/matching.

Finally, we consider two examples of explicit matchings. We use similar settings as before where passengers are assumed not to have access to a car and therefore

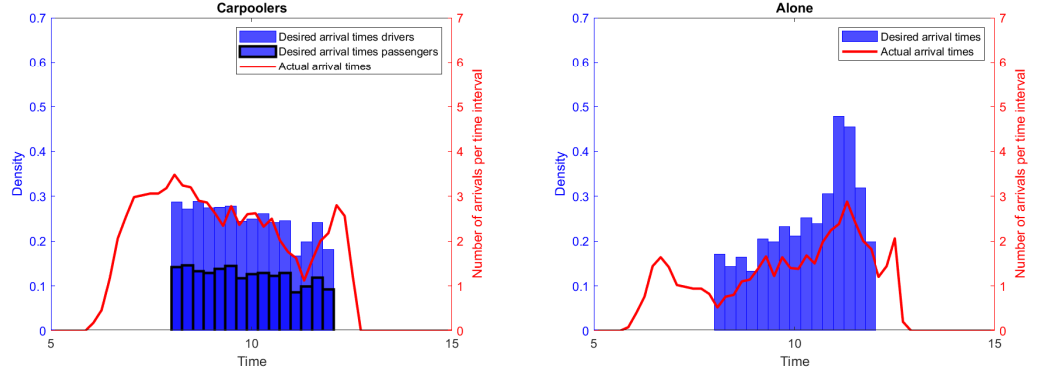


Figure 3.12: Pattern of desired and actual arrival times for carpoolers and individual drivers if passengers own a car

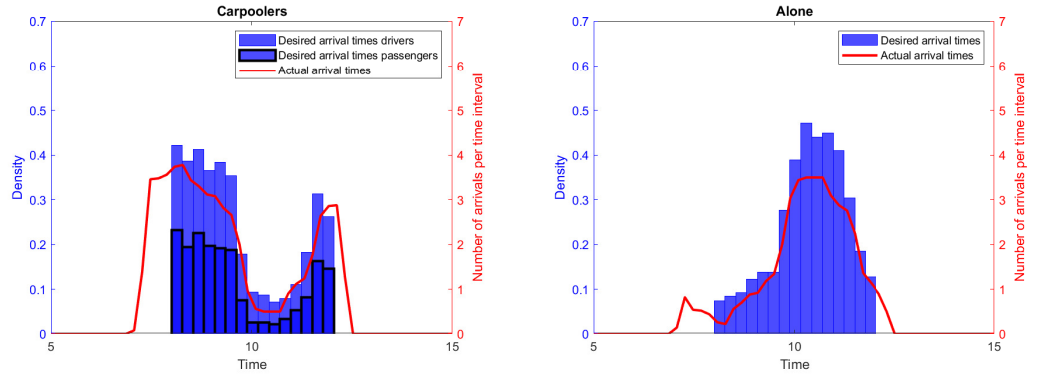


Figure 3.13: Pattern of desired and actual arrival times for carpoolers and individual drivers if passengers do not own a car

use public transport if they are not matched (Option A). In Figure 3.14, the points mark the drivers (blue) and passengers (red) based on their location on the corridor and their desired arrival time. The line segments between the points indicate the matches. The results are obtained through a single run and display 100 drivers, 100 passengers and their corresponding matches. Figure 3.14a considers the case where bottleneck capacity is relatively high and therefore congestion is negligible. We observe, in line with Theorem 4, that some of the left-most passengers and the right-most drivers remain unmatched. Further, drivers are mainly located to the left of their matched passenger, to minimize the detour and desired arrival times are extremely similar. Other than what was stated in Theorem 2, drivers and passengers are not necessarily matched in sequence. This is caused both by non-uniqueness of the solution, as well as the presence of scheduling delay penalties that have to be taken into account when finding the optimal matching. Figure 3.14b displays the explicit matches for a bottleneck capacity of 4. Congestion is more significant here and therefore influences the matching. We make similar observations as for the high capacity case, however, there is one important difference. Again, carpoolers

mainly occur in the tail of the travel period. Those individuals with desired arrival times between 10 and 11 (the peak period of congestion) choose to travel alone or resort to their alternative options for drivers and passengers respectively.

A closer look at the arrival times of matches where one of the two individuals has a desired arrival time that is substantially closer to the peak of delay indicates that these matches generally avoid the peak of delay. Specifically, their actual arrival time tends to the desired arrival time that is furthest from the peak. This explains the behaviour in Figure 3.12 where carpoolers typically avoid the bottleneck more than solo-drivers.

We also note that when desired arrival times are heterogeneous and congestion exists, the optimal matching may not be unique. Consider two drivers that are located before two passengers such that in any matching of these four individuals will lead to zero detour. In equilibrium, individuals can be indifferent between arriving early and arriving on-time (or between late and on-time, depending on their desired arrival time relative to the peak of congestion). If all four individuals have overlapping time intervals where they are indifferent, the two potential matchings lead to identical objective values.

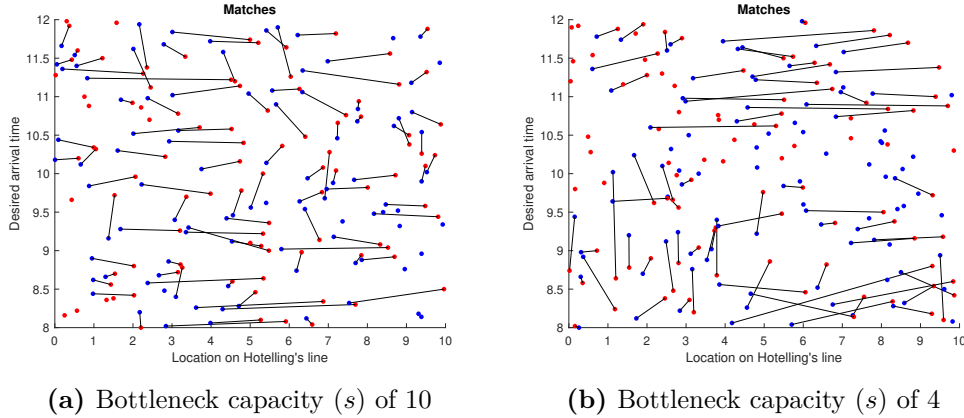


Figure 3.14: Explicit matches of drivers and passengers

3.5 Summary

We developed a framework for matching in carpooling or ride-sharing incorporating dynamic bottleneck congestion. We first described various fundamental properties of matching decisions in carpooling. Our theoretical results showed that, if the only matching inconvenience is a possible detour, a system optimal matching is obtained when the drivers and passengers are matched based on their location in the sequence of drivers and passengers. When individuals differ in their desired

arrival time, matching induces tardiness penalties. For a given match, their jointly optimal desired arrival time is the lowest desired arrival time of the matched couple. Matching therefore induces earliness and reduces lateness.

In order to evaluate the two-way causal effect of dynamic congestion and matching decisions, we developed an iterative approach where matching decisions are adapted based on bottleneck congestion. In the presence of bottleneck congestion, we still observe a tendency towards early arrival for carpoolers, thereby moving the peak of congestion forward in time. Our experimental results show that our algorithm tends to converge to a near-optimal solution within 50 iterations. Thereby, we observe that the optimal matching, and specifically the optimal number of matches, depends heavily on the level of congestion, potential subsidies and the number of available drivers and passengers. A microscopic analysis indicated that carpoolers and non-carpoolers show different behavioural patterns in the presence of congestion, where carpoolers mainly try to avoid the peak of congestion. The optimal matching is also influenced by this behaviour, such that individuals with a desired arrival time closer to the peak of congestion are more likely to travel alone.

We have proposed a first model on how matching models may be adapted in case of congestion. Congestion distorts the matching costs and may therefore distort the optimal matching. In that case, theoretical rules may no longer apply. A central operator that imposes the matching is more likely to know traffic conditions more precisely and in particular what the effect of the proposed matching on traffic congestion will be. As private agents are less likely to know about traffic conditions, in this case the market will lead to another solution, if congestion is not internalized via congestion pricing. This justifies the choice for a central operator that imposes the matching. The implementation of our algorithm requires more testing for a realistic network, which will be the task of future work.

We have assumed so far that any commuter is either a driver or a passenger. This is not necessarily the case, as individuals with a car may be flexible in their choice to be a passenger or a driver. The matching problem can be extended to this more complex setting. This has been explored on a realistic network of the city of Paris by Palma, Javaudin, Stokkink, and Tarpin-Pitre, 2022.

By targeting subsidies to those drivers that have the highest matching costs, more matches can be formed with the same budget. This outlines the benefit of an adaptive pricing policy to stimulate carpooling. The development of such an adaptive (targeted) pricing policy is an important direction of further research. Several adaptive pricing policies have been proposed for the ride-sourcing problem by among others Yang, Shao, Wang, and Ye, 2020 and Zha, Yin, and Xu, 2018,

which bears many similarities to the carpooling problem. Further research can shed light in this direction.

4

Multi-modal ride-matching with transfers and travel-time uncertainty

This chapter is based on the following article:

- P. Stokkink, A. de Palma, and N. Geroliminis (2023). “Multi-modal ride-matching with transfers and travel-time uncertainty”. Available in: THEMA Working Papers (Currently under review in a scientific journal)

4.1 Introduction

One of the reasons that current carpooling systems are not successful is the lack of a central operator who can match riders in a multimodal system with different itineraries in a reliable and efficient way for all parties involved. An important operational limitation of direct ride-sharing is that a pairing of drivers and riders needs to be found with matchable itineraries. This means that a driver needs to be able to pick up and drop off the matched rider without deviating too much from their original route. In addition to this, desired arrival times of the rider and the driver need to be similar. Dissimilar matches increase the costs of drivers and riders, such that ride-sharing is no longer competitive with private or public transport.

A potential solution for this is to allow riders to transfer between drivers and between multiple modes of transportation. In this work, we consider the possibility of a single transfer, since more than one transfer makes ride-sharing a less appealing alternative. By allowing transfers, a larger set of potential matches is available

for drivers and riders since the itineraries only need to be partially similar, as they spend only a part of their trip together. Transfers may also promote family carpooling at least for part of their journey. As families share their origin but do not necessarily share their destination, they can spend the first part of their trip together before one of them transfers. According to Li et al., 2007, family carpooling makes up nearly 75% of all carpools.

Despite this benefit, transfers may impose additional difficulties in case travel time is uncertain. In that case, riders or drivers may arrive late to their match, which may either cause them to miss their ride or to delay the matched individual as well. Thereby, for a transfer between ride-sharing and public transport, uncertain travel times may lead to missed connections. In the presence of uncertainty, transfers can make ride-sharing with transfers less appealing due to their effect on tardiness and uncomfortable and unanticipated waiting times. For that, it is important to develop adaptive matching strategies to maintain the attractiveness of ride-sharing.

In this chapter, we consider a ride-matching framework with transfers and uncertainty in travel time. We consider riders' transfers between drivers and between modes of transport. In our framework, riders are able to match with multiple drivers sequentially and drivers are able to match with multiple riders both sequentially and simultaneously if the capacity of their car allows. Riders can transfer between modes or drivers at designated transfer hubs as depicted in Figure 4.1. Transfer hubs have connections to public transport services and offer parking opportunities for riders that use their own car to reach the transfer hub. By considering travel time uncertainty, schedule delay and waiting time penalties of potential matches can be influenced. Potential matches may be infeasible for some uncertain scenarios, which can therefore affect the optimal matching. We first model the multi-modal ride-matching problem with transfers as a deterministic integer programming problem. Then, we extend this model to a two-stage stochastic programming problem where travel time uncertainty forms the division between first- and second-stage decisions. Matches on the first leg of the trip are made under uncertainty, whereas second-leg matches are made afterwards when information on travel time is gathered.

This chapter is organized as follows. The deterministic matching approach is described in Section 4.2. The stochastic matching approach is described in Section 4.3. Numerical results are provided in Section 4.4 and the chapter is concluded in Section 4.5.

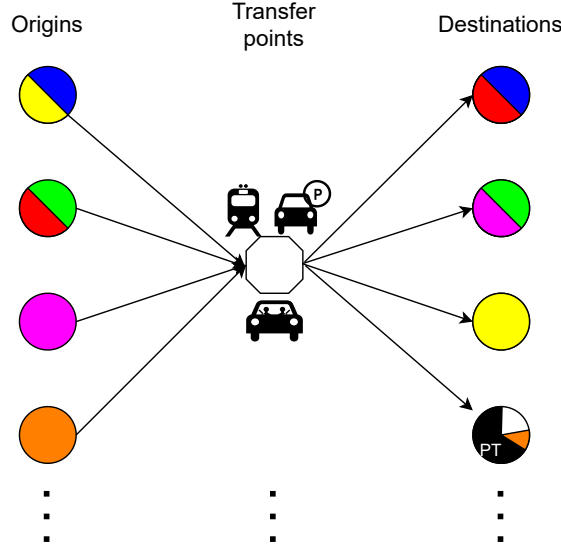


Figure 4.1: Graphic illustration of transfers. Colors represent individual commuters. The lower-left half of a circle represents the driver and the upper-right half represents the rider. If drivers are not matched to a rider, the full circle is colored by the driver.

4.2 Deterministic matching

In this section, we describe the deterministic ride-matching problem. We provide the modelling assumptions and the mathematical formulation in Section 4.2.1. We describe the way the costs and parameters are computed in Section 4.2.2 and describe the departure time choices of riders and drivers in Section 4.2.3.

4.2.1 Assumptions and mathematical formulation

The deterministic matching approach is based on a set of predefined rider paths. Let I be the set of riders, J the set of drivers, and H the set of transfer hubs. The set of riders can be split into two subsets I^c and I^{nc} according to car ownership. Those in I^c own a car which they may use if they are not matched to a driver, whereas those in I^{nc} do not own a car and will therefore take public transport if they are not matched to a driver. Although the number of possible matches can get large, it is still polynomial in the number of riders, drivers, and transfer hubs. Note that, if the number of transfers is not limited to one, the number of potential matches would increase exponentially in the number of transfer hubs. Given that the possible number of matches for riders is polynomial, we generate all possible paths in advance. We let the drivers set the departure times, such that the costs of rider paths are independent and such that the cost of a path is independent of the total matching which allows us to determine the costs à-priori. The problem then reduces to selecting the optimal set of rider paths, taking into account that drivers may carry multiple riders both sequentially and simultaneously, as long as

Table 4.1: Notational glossary

Sets	
H	Set of transfer hubs (indexed h)
I	Set of riders (indexed i)
I^c	Set of riders owning a car
I^{nc}	Set of riders not owning a car
J	Set of drivers (indexed j)
K	Set of rider paths (indexed k)
M	Set of transport modes (indexed m)
T	Set of discrete time intervals (indexed t)
$K(m, t)$	Set of rider paths for which the first-leg mode is $m \in M$ and first-leg departure time is $t \in T$
Ω	Set of uncertain scenarios of travel-time (indexed ω)
Parameters	
a_{jk}^0	Binary parameter indicating if driver $j \in J$ contributes to rider path $k \in K$ through a direct trip
a_{jk}^{1h}	Binary parameter indicating if driver $j \in J$ contributes to rider path $k \in K$ through a first-leg trip to transfer hub $h \in H$
a_{jk}^{2h}	Binary parameter indicating if driver $j \in J$ contributes to rider path $k \in K$ through a second-leg trip from transfer hub $h \in H$
c_k	Generalized cost of rider path $k \in K$
$c_k(\omega)$	Generalized cost of rider path $k \in K$ for scenario $\omega \in \Omega$
d_i	Destination of individual $i \in I \cup J$
e_{ik}	Binary parameter indicating if rider path $k \in K$ corresponds to rider $i \in I$
q_j	Capacity of the car of driver $j \in J$
o_i	Origin of individual $i \in I \cup J$
$tt(\cdot, \cdot)$	travel time between two nodes in the network
α^{car}	Cost per time unit spent in a car
α^{pt}	Cost per time unit spent in public transport
α^{wait}	Cost per time unit spent waiting at a transfer hub
β	Cost per time unit arriving early at the destination
γ	Cost per time unit arriving late at the destination
τ	Maximum detour a driver is willing to make
ϕ_d^{park}	Fixed cost of parking at destination d
ϕ^{fuel}	Cost of fuel per time unit in a car
ϕ^{pt}	Cost of public transit per leg
Decision Variables	
x_k	Binary decision variable indicating if rider path $k \in K$ is selected
$x_k(\omega)$	Binary decision variable indicating if rider path $k \in K$ is selected in scenario $\omega \in \Omega$
y_{jh}	Binary decision variable indicating if driver j travels through transfer hub j
$y_{jh}(\omega)$	Binary decision variable indicating if driver j travels through transfer hub j in scenario $\omega \in \Omega$

their paths are compatible and the capacity of the car is not exceeded.

The deterministic model is based on the following set of assumptions. Here, Assumption (A1) is required for the formulation to hold, whereas Assumptions (A2) - (A5) can be relaxed without significantly influencing the formulation and the solution framework. For Assumptions (A2) - (A5), we consider that they add efficiency to the system and they make the operational platform more appealing

and acceptable to riders and drivers.

- (A1) Travel times are exogenous and time-independent and there is no congestion.
- (A2) Drivers determine their departure time that minimizes their own costs. If matched, a rider must agree to the departure time of the driver.
- (A3) The matching is determined by a central operator but drivers and riders only accept the match if their costs are lower than that of their solo-travel alternatives. The objective of the central operator is to minimize the costs of all riders.
- (A4) (a) Drivers can only perform a pickup in their departure zone or at a transfer hub and only perform a drop-off at a transfer hub or in their arrival zone.
 (b) Riders can only be picked up in their departure zone or at a transfer hub and only be dropped off at a transfer hub or in their arrival zone.
- (A5) Drivers can reach a transfer hub as long as their detour is at most τ time units (riders have no such constraint, as long as their costs are minimized)

Every individual has an origin o_i , a destination d_i and a desired arrival time t_i^* . Let K be the set of rider paths and let binary parameters $e_{ik} = 1$ if rider path k corresponds to rider i , and 0 otherwise. Binary parameters $a_{jk} = 1$ if driver j contributes to rider path k , and 0 otherwise. The cost of rider path k is denoted by c_k . Our model aims to minimize the total costs of riders and does not account for the costs of drivers. Since they determine the departure time, they do not incur any additional scheduling delay costs by sharing a ride. Other than that, drivers are assumed to be fully compensated for the inconvenience of sharing their car with others and the minor detour that may be involved with picking up and dropping off passengers. The design of compensation schemes for drivers is outside the scope of this work. Let decision variable $x_k = 1$ if rider path k is chosen and 0 otherwise. Let q_j be the capacity of the car of driver j , that is, the maximum number of riders driver j is able to transport at the same time. We let the driver only perform pickups at their own origin or the transfer hub and only perform drop-offs at their own destination or the transfer hub. We distinguish between direct trips that take a rider directly from their origin to their destination, and indirect trips that pass through a transfer hub. This means we can identify the following three types of trips, for which the binary parameter a_{jk} is adapted to denote the trip type and the transfer hub that is used.

- Direct trip: $a_{jk}^0 = 1$ if driver j contributes to rider path k through a direct trip.
- First leg of indirect trip: $a_{jk}^{1h} = 1$ if driver j contributes to rider path k through a first-leg trip to transfer hub h .

- Second leg of indirect trip: $a_{jk}^{2h} = 1$ if driver j contributes to rider path k through a second-leg trip from transfer hub h .

We use decision variable y_{jh} to define through which transfer hub driver j is going. Similar to ride-sharing paths, public transport paths and paths where riders use their own private car have a corresponding cost c_k . Since no driver is involved in these paths all a_{jk}^l parameters are equal to 0. For a multi-modal path where one leg is a ride-sharing leg, only the corresponding a_{jk}^l is 1, and the others all remain zero. We formulate the deterministic matching problem as follows:

$$(P1) \text{ minimize } \sum_{k \in K} c_k x_k \quad (4.1a)$$

such that

$$\sum_{k \in K} e_{ik} x_k = 1 \quad \forall i \in I \quad (4.1b)$$

$$\sum_{k \in K} a_{jk}^0 x_k \leq q_j \left(1 - \sum_{h \in H} y_{jh} \right) \quad \forall j \in J \quad (4.1c)$$

$$\sum_{k \in K} a_{jk}^{1h} x_k \leq q_j y_{jh} \quad \forall j \in J, h \in H \quad (4.1d)$$

$$\sum_{k \in K} a_{jk}^{2h} x_k \leq q_j y_{jh} \quad \forall j \in J, h \in H \quad (4.1e)$$

$$\sum_{h \in H} y_{jh} \leq 1 \quad \forall j \in J \quad (4.1f)$$

$$x_k \in \mathbb{B} \quad \forall k \in K \quad (4.1g)$$

$$y_{jh} \in \mathbb{B} \quad \forall j \in J, h \in H \quad (4.1h)$$

The objective (4.1a) is to minimize the cost of all matches. Every rider needs to be matched to exactly one driver, which is enforced by Constraints (4.1b). Feasibility of the solution from the perspective of a driver is enforced through Constraints (4.1c) - (4.1e). The feasibility of the solution from the perspective of a rider is enforced directly on the set of paths K . That is, the set K only contains paths that are feasible for a rider. On every leg, a driver $j \in J$ may have at most q_j riders in his/her car, which is enforced jointly by Constraints (4.1c), (4.1d) and (4.1e). A driver may either serve riders directly from his/her origin to his/her destination or through a transfer hub, but not both. This means that when a driver j makes an indirect trip, he/she can carry q_j riders on the first leg and q_j riders on the second leg. The set of riders on both legs may be partially similar, but it is possible that a driver carries $2q_j$ unique passengers on his/her full trip. Constraints (4.1f) ensure that a driver only makes a stop at one hub. These constraints also ensure that the first and second legs of a driver are compatible. That is, the first leg ends at the same transfer hub as the second leg starts.

A feasible solution to this problem always exists as long as every rider has access to public transport. As public transport capacity is unlimited, every rider will have a corresponding public transport path. The solution in which every rider uses public transport as a direct path between their origin and destination is then always feasible. The optimal solution obtained by solving **P1** is not necessarily unique, as multiple solutions may lead to the same objective value. According to Proposition 3 (see Appendix C.2.1), the optimal solution to **P1** is also a stable solution if riders are aware of their alternative transport modes and their costs but are not aware of alternative matches besides their proposed match. As riders can always choose to take public transport or ride their own car (there is no capacity constraint on these modes), the optimal solution will always select these paths if the costs are lower than other paths. According to Proposition 4 (see Appendix C.2.1), the set of paths K can be reduced by removing strictly dominated paths. A dominated path is a path where, by replacing one or multiple legs with a solo leg (either public transport or solo driving), the cost can be reduced. By reducing the number of paths in K , the computation time to solve **P1** can be reduced.

4.2.2 Computation of costs and parameter values

In this subsection, we describe the costs and parameters of all rider paths. We distinguish between direct paths and indirect paths that go through a transfer hub. We consider three potential modes for riders: solo driving (SD) for those riders that own a private car, public transport (PT), and ride-sharing (RS). Each mode can be used as a direct path, or a combination of two modes can form an indirect path. Given that solo driving is not possible as a second-leg mode after public transport or ride-sharing (because they left their car at home) we have 7 potential mode choice combinations for indirect paths.

We consider the following cost components and the corresponding parameters. We let α^{car} be the value of time spent in a car, β the penalty for every unit of time an individual is early, and γ the penalty for every unit of time an individual is late. Waiting time is penalized by α^{wait} and the value of time spent in a car may be different from the value of time spent in public transport, which is defined as α^{pt} . Thereby, public transport has a fixed cost ϕ^{pt} . Riders that own a car may choose to drive themselves. They incur ϕ^{fuel} fuel cost per time unit, on top of their value of time, and have to pay a parking fee ϕ_d^{park} at destination d . Travel time between o and d is defined as $tt(o, d)$. We highlight that for the sake of notation, these parameters are all homogeneous. However, the formulation allows for fully heterogeneous parameter values among all individuals. In the latter, we use linear functions of earliness, lateness, and waiting time with respect to time. The computation of the cost provided in Appendix C.1 can be generalized to non-linear cost functions without changing the problem formulation in **P1**. Schedule delay

penalties for commuters are incorporated in a way that is consistent with Small, 1982 and Arnott, Palma, and Lindsey, 1993 and has been previously used in a ride-sharing framework by Palma, Stokkink, and Geroliminis, 2022.

The generalized costs consist of in-vehicle costs depending on the mode, possible physical costs such as fuel or a public transport ticket, waiting penalties, and schedule delay penalties. The exact cost formulation depends on the specific type of path and the departure time choice. When driving themselves, riders can leave at any time t . For the sake of tractability, we consider a set of discrete time intervals $t \in T$ at which a rider can leave. The optimal departure times of carpooling drivers are also mapped to the closest discrete time interval $t \in T$. Drivers as well as riders that are using their own cars determine their departure time in advance. A detailed description of departure time choices is provided in Section 4.2.3. For completeness, the exact cost definitions for each type of path is given in Appendix C.1.

We note that in the computation of the costs, we only considered the costs of the riders. The reason for this is that due to the inflexibility of the drivers with respect to departure time, they do not incur any extra scheduling delay costs with a rider. We can neglect the payment of riders to drivers since these are direct money transfers and therefore do not change the solution of the optimization problem. For example, riders that save a percentage of their (expected) costs can share it with the driver. The compensation schemes are outside the scope of this work but deserve future research attention.

4.2.3 Departure time choice

In this work, we assume that drivers choose their departure time such that they minimize their own generalized cost. The reason for this is that coordination of departure times in a complex system where riders match with multiple drivers and drivers match with multiple riders is difficult both theoretically and in practice. However, there are some special cases for which the departure times can be determined optimally. In this section, we discuss those special cases and the jointly optimal departure times of matches.

Without transfers, the optimal departure time has a closed form solution. Consider a direct match where a single driver takes a group of riders directly from their origin to their destination. In case lateness is penalized heavier than earliness, the jointly optimal departure time is the minimal departure time of all matched individuals (See Theorem 9 in Appendix C.2). In this case, everyone is either on-time or early and no one is late. Every rider is matched to at most one driver and therefore the problem can be decomposed over the groups of agents that share a ride altogether. The optimal departure time can be determined independently for every group.

With a transfer, the problem complexifies as more coordination is required. Consider a set of riders with identical destinations d that make a transfer at the transfer hub $h \in H$. Also, consider a driver with an identical destination as the riders that only performs a ride-sharing trip between the same hub h and destination d . According to Theorem 10 in Appendix C.2, the jointly optimal departure time on the second leg is a function of the arrival time of riders at the second leg, as well as the minimal desired arrival time. According to Theorem 11 in Appendix C.2 the optimal departure on the first leg for a driver that takes one rider on the first leg and another rider on the second leg depends on the desired arrival time of all three individuals. The optimal departure time on the first leg also depends on the rider on the second leg, although they are not directly involved.

The results of Theorem 10 and 11 in Appendix C.2 also emphasize the difficulty of coordination in more complex matching systems. If the driver takes another group of riders on his first leg, coordination of departure times with this group influences the departure time of the second group. Similarly, these riders may be matched to a second driver after making a transfer, therefore also influencing his departure time and vice versa. For a large system where many drivers and riders are (indirectly) connected to each other, determining the jointly optimal departure time is a complex problem to solve and difficult to implement both theoretically and in practice. Therefore, in this work, we consider that the driver is in charge of determining the departure times. In current ride-sharing systems such as BlaBlaCar, the driver is also in charge of determining the departure time, and the rider is forced to adapt if they are matched.

4.3 Two-stage stochastic matching

In this section, we describe the two-stage stochastic ride-matching problem. In Section 4.3.1 we provide the assumptions and mathematical formulation of the problem. In Section 4.3.2 we describe how the computation of the costs and parameters is different from the deterministic variant and in Section 4.3.3 we provide two benchmarks on the formulation with respect to uncertainty and information availability.

4.3.1 Assumptions and mathematical formulation

Travel time in transportation systems is often prone to uncertainty. Travel time may change as a consequence of, for example, congestion, weather, or unexpected roadblocks and accidents. As a consequence, riders may suffer from additional scheduling delay penalties or miss a connection at the transfer hub. In case a rider

is too late for the driver to pick him/her up, he/she will be forced to take public transport instead or match to another driver. To adapt to uncertainty, the optimal matching may be (partially) different. For example, a rider could depart earlier or match to a driver that leaves earlier to create a safety margin for unexpected delays. In this way, waiting time at the transfer hub and the associated costs may increase, but the chances of missing a connection are lower.

We adapt our formulation to a two-stage stochastic programming problem to incorporate uncertainty in travel times. The first stage corresponds to the first leg (i.e., origin to transfer hub), and the second stage corresponds to the second leg (i.e., transfer hub to destination). The matching for the first leg is made with uncertainty about the exact travel times, but the probability distribution of travel times, denoted by Ω , is known. These are referred to as first-stage matching decisions. Then, the matches for the second leg are made after the exact travel times are observed. The exact travel times are a realization $\omega \in \Omega$. These are referred to as second-stage matching decisions.

A timeline for the decision-making process of the stochastic matching approach is given in Figure 4.2. First-stage matching decisions are made in advance. Here, all direct matches as well as first-leg matches (pickup from origin and drop-off at hub) are determined. Thereafter, in the second stage, second-leg matches are determined (pickup from hub and drop-off at destination). First-stage decisions are made based on the probability distribution of travel times, whereas second-stage decisions are based on actual travel times. The first-stage decisions can be seen as a contract between the rider and the driver. Although actual travel times may be available before the driver departs, the matching may not be altered after the contract is agreed upon.

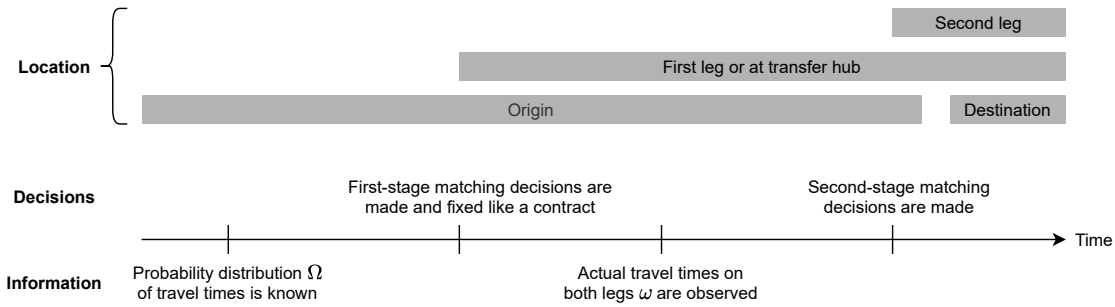


Figure 4.2: Timeline of decision-making in stochastic matching approach. The location bars indicate the potential physical locations of agents when they receive information and make decisions.

Some variables and parameters depend on the scenario ω . This means that instead of x_k we use $x_k(\omega)$ and instead of y_{jh} we use $y_{jh}(\omega)$. We note that strictly speaking

$y_{jh}(\omega)$ is by definition the same across all scenarios, but we still model it as a scenario-dependent variable for one of the benchmarks discussed in Section 4.3.3. In addition to this, the cost of path k also depends on the scenario as it influences travel time and may even make some paths infeasible. Therefore, we change c_k to $c_k(\omega)$, where $c_k(\omega) = \infty$ if path k is infeasible for scenario ω . This may happen, for example, when the rider arrives at the transfer hub after their driver has already departed because of a delay. We denote $p(\omega)$ the probability of scenario ω occurring, such that $p(\omega) \geq 0$, $\sum_{\omega \in \Omega} p(\omega) = 1$ and $p(\omega_1 \cup \omega_2) = p(\omega_1) + p(\omega_2)$ if ω_1 and ω_2 are disjoint.

In each of the scenarios $\omega \in \Omega$, a driver may match to another rider and vice versa. A match that is optimal in at least one of the scenarios $\omega \in \Omega$ is referred to as a *potentially optimal match*, as per Definition 1. Every potentially optimal match may lead to a different departure time for the rider (as before, the driver is free to set his/her departure time and therefore has a unique potentially optimal departure time, independent of the match). The set of optimal departure times for every potentially optimal match is referred to as a *potentially optimal departure time*, as per Definition 2.

Definition 1: Potentially optimal match: A second-stage match between driver i and rider j which is optimal in at least one of the scenarios $\omega \in \Omega$.

Definition 2: Potentially optimal departure time: An optimal departure time of an agent for at least one of their potentially optimal matches.

In addition to the assumptions for the deterministic model, we make the following assumptions for the stochastic model:

- (A6) Travel times on every arc are time-independent and are drawn from a multivariate random distribution Ω
- (A7) Every rider, driver, and the operator perfectly knows the distribution Ω
- (A8) The central operator makes first-stage decisions with full knowledge of the distribution Ω , but with no knowledge of the realization (i.e., actual travel times)
- (A9) First-stage decisions are like a contract that is made before departure. Riders, drivers, and the operator have to stick to them even though they may be aware of suboptimality due to observed conditions.
- (A10) The realization ω is revealed to everyone at the same time. First-stage decisions cannot be changed, because they are fixed like a contract. All second-stage decisions are made after the realization of ω is available.

Given the previous assumptions, all first-stage decisions have to be the same for all scenarios. To enforce this, we extend **P1** with scenario-dependent variables and parameters and impose the equality of first-stage decisions across scenarios through the following constraints. As in the deterministic case, we distinguish between direct and indirect paths. For direct paths, the full path needs to be the same (i.e., the mode, driver, and departure time). For an indirect path, only the first leg needs to be the same. This means that the mode, driver, and departure time on the first leg are fixed, whereas the second leg may be completely different.

For direct paths, we add the following set of constraints. If driver j is involved in path k for scenario ω that is a direct match from origin to destination, he must commit to doing the same direct path in any other scenario. Therefore, as direct paths only have one leg, the chosen paths are identical for every scenario and are used to minimize the expected cost. As the exact same path is chosen, this implies that the mode, driver, and departure time are also identical. This is enforced through:

$$a_{jk}^0 x_k(\omega) = a_{jk}^0 x_k(\omega') \quad \forall j \in J, k \in K, \omega, \omega' \in \Omega. \quad (4.2)$$

For an indirect path that goes through a transfer hub, only the first leg is fixed. First, we consider indirect paths where the first leg is a ride-sharing leg. In this case, the full path need not be the same, as long as the same driver goes to the same hub in both scenarios. As the driver determines his/her expected optimal departure time à-priori and imposes this on the rider, this is automatically forced to be identical across scenarios. In addition to this, we impose that both paths need to correspond to the same rider. By enforcing the matched driver-rider pair as well as the hub at which the rider is dropped to be equal across scenarios, we guarantee that first-leg matches are fixed in advance. To enforce this, we use the following set of constraints:

$$\sum_{k \in K} e_{ik} a_{jk}^{1h} x_k(\omega) = \sum_{k \in K} e_{ik} a_{jk}^{1h} x_k(\omega') \quad \forall i \in I, j \in J, h \in H, \omega, \omega' \in \Omega. \quad (4.3)$$

Finally, for indirect paths where the first leg is not a ride-sharing leg, we impose that the mode and departure time is the same across scenarios. Let $m_k^1 \in M$ denote the chosen mode on the first leg of path k where $M = \{\text{ride-sharing, public transport, solo driving}\}$ and let $t_k^1 \in T$ denote the departure time on the first leg of path k . We define $K(m, t) = \{k \in K | m_k = m, t_k = t\}$. That is, $K(m, t) \subset K$ is the set of paths for which the mode on the first leg is equal to $m \in M$ and the departure time is $t \in T$. Using this definition, we enforce that the modes and departure times on the first leg need to be identical across scenarios for the same rider $i \in I$, through the following set of constraints:

$$\sum_{k \in K(m, t)} e_{ik} x_k(\omega) = \sum_{k \in K(m, t)} e_{ik} x_k(\omega') \quad \forall i \in I, j \in J, h \in H, m \in M, t \in T, \omega, \omega' \in \Omega. \quad (4.4)$$

The full formulation of the stochastic programming problem, to which we refer as **P2** is given as follows:

$$(\mathbf{P2}) \text{ minimize } \sum_{\omega \in \Omega} \sum_{k \in K} p(\omega) c_k(\omega) x_k(\omega) \quad (4.5a)$$

such that

$$\sum_{k \in K} e_{ik} x_k(\omega) = 1 \quad \forall i \in I, \omega \in \Omega \quad (4.5b)$$

$$\sum_{k \in K} a_{jk}^0 x_k(\omega) \leq q_j \left(1 - \sum_{h \in H} y_{jh}(\omega) \right) \quad \forall j \in J, \omega \in \Omega \quad (4.5c)$$

$$\sum_{k \in K} a_{jk}^{1h} x_k(\omega) \leq q_j y_{jh}(\omega) \quad \forall j \in J, h \in H, \omega \in \Omega \quad (4.5d)$$

$$\sum_{k \in K} a_{jk}^{2h} x_k(\omega) \leq q_j y_{jh}(\omega) \quad \forall j \in J, h \in H, \omega \in \Omega \quad (4.5e)$$

$$\sum_{h \in H} y_{jh}(\omega) \leq 1 \quad \forall j \in J, \omega \in \Omega \quad (4.5f)$$

$$a_{jk}^0 x_k(\omega) = a_{jk}^0 x_k(\omega') \quad \forall j \in J, k \in K, \omega, \omega' \in \Omega \quad (4.5g)$$

$$\sum_{k \in K} e_{ik} a_{jk}^{1h} x_k(\omega) = \sum_{k \in K} e_{ik} a_{jk}^{1h} x_k(\omega') \quad \forall i \in I, j \in J, h \in H, \omega, \omega' \in \Omega \quad (4.5h)$$

$$\sum_{k \in K(m,t)} e_{ik} x_k(\omega) = \sum_{k \in K(m,t)} e_{ik} x_k(\omega') \quad \forall i \in I, j \in J, h \in H, m \in M, t \in T, \omega, \omega' \in \Omega \quad (4.5i)$$

$$x_k(\omega) \in \mathbb{B} \quad \forall k \in K, \omega \in \Omega \quad (4.5j)$$

$$y_{jh}(\omega) \in \mathbb{B} \quad \forall j \in J, h \in H, \omega \in \Omega \quad (4.5k)$$

The Objective (4.5a) is to minimize the expected costs, which is a linear function weighted by the probability of each scenario occurring. Constraints (4.5b) to (4.5f) are the same as in **(P1)**, but adapted to the various scenarios, by extending them with the scenario dependency ω . Constraints (4.5g) ensure that the exact same direct paths are chosen for every scenario, which automatically imposes that the rider, mode, and departure time are identical across the scenarios. Constraints (4.5h) enforce the first leg of drivers to be the same on indirect paths and that they carry the same rider. Again, this automatically imposes that the mode and

departure time are identical across scenarios, as these are determined à-priori by the driver. Finally, Constraints (4.5i) enforce that a rider that does not share a ride on the first leg but rather drives alone or uses public transport has the same mode of transportation and the same departure time across scenarios.

Using the formulation **P2** we determine the matching that minimizes the expected costs. The first-stage decisions determine the direct matches and the matches on the first leg, which minimize the expected costs on both legs, taking into account all possible second-stage scenarios. The first leg decisions impose constraints on the decisions that can be made in the second leg, by choosing a transfer hub, a departure time on the first leg, and the mode choice and hence influencing the arrival time at the chosen transfer hub. In the second stage, we then optimize the second-leg matches by taking into account these constraints. By formulating the two stages as a single stochastic optimization problem, first-stage decisions are chosen optimally by considering all possible second-stage scenarios and the constraints that are imposed on the second stage by the first-stage decisions.

4.3.2 Computation of costs and parameter values

The computation of the cost on each path is similar to that described in Section 4.2. Here, instead of using the fixed travel time $tt(\cdot, \cdot)$, we use the scenario-dependent, yet exogenous, travel time $tt(\cdot, \cdot, \omega)$. This then allows computing the costs $c_k(\omega)$ for all paths $k \in K$.

Every driver that is traveling on a direct path or on the first leg determines his/her departure time in advance. A driver determines his/her expected optimal departure time considering the probability distribution of Ω and the expected earliness and lateness they encounter for a given departure time. This is formalized in Theorem 12. Given the expected optimal departure time of drivers, the costs for every scenario $c_k(\omega)$ can be determined. For this it is important to note that second-leg departure times are flexible, but immediate departures are always preferred over waiting for reasonable values of α^{wait} and β , according to Theorem 10.

As opposed to the deterministic case, the optimal departure time for a rider cannot be determined à-priori. As riders may make different second-stage decisions for every scenario, this influences their expected optimal departure time. We compute all potentially optimal departure times for a rider, for all possible paths that he/she can take. Given that the cost is a linear combination of the costs of the individual paths, the optimal departure time has to intersect with one of these potentially optimal departure times. A separate path is constructed for every potentially optimal departure time if this path leads to a feasible solution. This is formally denoted in Theorem 12. For specific shapes of the uncertainty set Ω , for example

when $p(\omega)$ is uniform, the expected optimal departure time can be defined exactly. This is formally denoted in Corollary 1.

Direct paths and indirect paths ending in a public transport leg are always feasible, independent of the scenario ω . Indirect paths ending in a ride-sharing leg, on the other hand, are infeasible if the delay causes an arrival at the transfer hub after the driver has already departed. For such a scenario ω , $c_k(\omega) = \infty$. We also note that if a path $k \in K$ has cost $c_k(\omega) = \infty$ for every $\omega \in \Omega$, this path can be omitted from K . In the stochastic case, the rule described in Proposition 4 to remove strictly dominated paths does not apply. A counterexample to disprove Proposition 4 in the case of stochastic travel times is given in Remark 1. However, a weaker rule can be used in the stochastic case, as described in Proposition 5. Here, we assume a path is dominated by another path if the cost is lower for every scenario, allowing the first path to be combined with the path with minimal cost for all but one scenario. In this way, we can identify that a path will never be used, not even in combination with the best possible path.

4.3.3 Benchmarks

A comparison of the stochastic programming problem to the following two benchmarks can attribute a value to the level of information. The expected-value benchmark has limited information and only assumes the distribution of Ω is known in advance, but no information on the scenario $\omega \in \Omega$ is obtained anywhere throughout the commute. The stochastic programming problem in **P2** also assumes the distribution of Ω is known before the commute, but the exact scenario $\omega \in \Omega$ is observed after the first-stage decisions are fixed but before the second-stage decisions are made. Finally, the wait-and-see benchmark assumes the exact scenario $\omega \in \Omega$ is known at the start of the day and allows one to make full scenario-dependent decisions even for the first stage. It is clear that this benchmark assumes the highest level of information.

Definition 3: Wait-and-see benchmark: Relaxation of the stochastic programming problem where completely independent decisions can be made for every scenario.

The solution to the wait-and-see benchmark describes what would be the optimal solution if the realized scenario was to be identified in advance. In general, the wait-and-see benchmark is obtained by relaxing Constraints (4.5g), (4.5h), and (4.5i) in **P2**. However, in the precomputation of the paths K and corresponding costs $c_k(\omega)$, drivers are assumed to anticipate all scenarios and determine their expected optimal departure time à-priori. For the wait-and-see benchmark, drivers determine their optimal departure time for the exact scenario. Therefore, for this benchmark, we determine the costs $c_k(\omega)$ for every scenario independent of the

other scenarios and relax Constraints (4.5g) and (4.5h). This new formulation is referred to as **P3**. This allows solutions to be completely different for the various scenarios, making it a lower bound to **P2**.

Definition 4: Expected-value benchmark: Restricted version of the stochastic programming problem where completely identical decisions have to be made for every scenario.

The solution to the expected-value benchmark describes what would be the optimal solution if all matches were determined in advance. The expected value benchmark is obtained by replacing c_k in **P1** by $\sum_{\omega \in \Omega} p(\omega) c_k(\omega)$. Here, only one solution is obtained which minimizes the expected value of the costs for the scenarios. This removes the flexibility of adapting second-leg matches based on the observed traffic situation. Therefore, this solution is an upper bound to the stochastic problem in **P2**. We refer to this problem as **P4**. This problem is especially restrictive for indirect paths where the second leg is a ride-sharing leg. As for some paths, a path may be infeasible for one scenario ($c_k(\omega) = \infty$), and therefore $c_k = \infty$ in the expected value problem, ensuring that this path will never be selected. This results in paths only being selected in the expected value problem if the waiting time at the transfer hub is long enough to ensure the timing restriction (drop-off before pickup) is satisfied in every scenario.

4.4 Results

4.4.1 Case study

We evaluate our model on a circular city consisting of 33 nodes, as depicted in Figure 4.3. Every rider and driver has an origin and destination at one of the 33 nodes. Origins are more likely to be in the suburbs (the outer rings) whereas destinations are more likely to be in the city center. Transfer hubs can be at any of the nodes in the network. In our analysis, we use at most 9 hubs that are always added in the same order. The index of the hub is given in red in Figure 4.3. Finding the optimal hubs is an interesting direction of future research, but is outside the scope of this work. Drivers can perform a pick-up or a drop-off at one of the transfer hubs, but only if their shortest path between origin and destination already passes through this hub. Drivers do not make any detours. We consider 500 drivers and 500 riders. Out of those riders, 75% own a car which they may use to drive themselves. Desired arrival times are drawn from a truncated normal distribution with a mean at 8:00 and a standard deviation of 1 hour. The distribution is truncated such that we only allow desired arrival times between 7:00 and 9:00.

The parameter settings are homogeneous among the entire population and are defined as follows. The value of time spent in a car α^{car} is equal to 6.4[\$/h]. The value of time in public transport α^{pt} is higher and is set equal to 12.0[\$/h]. In addition to this, public transport has a fixed cost ϕ^{pt} of 2.0 per trip. Earliness and lateness are penalized with β and γ equal to 3.9[\$/h] and 15.21[\$/h] respectively, independently of the mode of transport. Waiting time is penalized by α^{wait} which is equal to 13.5[\$/h] such that $\beta < \alpha^{\text{car}} < \alpha^{\text{pt}} < \alpha^{\text{wait}} < \gamma$, consistent with the literature (Small, 1982). Fuel costs ϕ^{fuel} are equal to 4[\$/h] and parking costs ϕ^{park} are equal to 1.5\$. The percentage of riders owning a car is set to 75%.

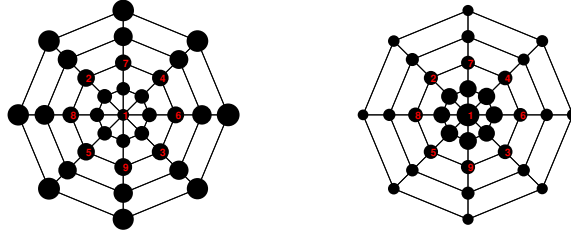


Figure 4.3: Circular city with the distribution of origins (left) and destinations (right) and in red the index of the transfer hub. The size of the node shows the density of trips starting (left) and ending (right) in each node

A lower bound on the average cost per rider in the discrete formulation is found when all riders find their perfect match (i.e., when origins, destinations, and desired arrival times of drivers and riders are identical). In this case, they only incur travel costs α^{car} . Given an average commuting time of 1 hour and 15 minutes in the synthetic data, the lower bound is $\frac{75}{60}\alpha^{\text{car}} = 8\$$. On the other hand, an upper bound is found when all riders use public transport. In this case, they all incur travel costs α^{pt} and the fixed cost ϕ^{pt} . Given an average commuting time of 1 hour and 15 minutes in the synthetic data, the upper bound is $\frac{75}{60}\alpha^{\text{pt}} + \phi^{\text{pt}} = 17\$$. We note that stronger bounds can be found by incorporating the portion of riders owning a car, by incorporating their private transport alternative.

All integer programming problems are implemented in Java with CPLEX version 12.6.3.0. All problems are solved to optimality and can be solved within a matter of seconds or minutes, depending on the exact problem configurations. Solving the stochastic programming problem typically takes more time and time goes up when the number of transfer hubs increases.

4.4.2 Influence of transfers on modal split, costs, and VHT

We consider the influence transfers make on the modal split, the average cost per individual, and the total Vehicle Hours Travelled (VHT). For this, we vary the number of transfer hubs in the system between 0 and 9. The results are an average of 10 randomly simulated instances. The results are displayed in Figure 4.4 where 4.4a displays the modal split of riders, 4.4b displays the average cost of riders, and 4.4c displays the VHT as a percentage of the VHT when ride-sharing is not available. Clearly, when there are no transfer hubs, the only possible mode choices are direct ride-sharing, solo driving, and public transport. By opening transfer hubs, a modal shift to the other modes is observed. Especially the number of riders ride-sharing on two separate legs and the number of riders using their own car on the first leg and ride-sharing on the second leg increases drastically. The reason for this is that by using a transfer, more options exist for matching to a driver with the same destination and a similar desired arrival time, at the cost of waiting at the transfer hub. The number of direct matches may be limited as the origin and destination of the rider and driver need to be identical and the desired arrival time needs to be relatively similar. Figure 4.4c displays that the total VHT by riders in their private car significantly decreases by 30% when allowing transfers, which has a direct influence on emissions.

Figure 4.4b displays how the costs change by opening transfer hubs. By using a single transfer hub in the center of the network, the average cost decreases from 12.80\$ to 12.10\$. Increasing the number of transfer hubs allows for a further decrease in the average cost, but not nearly as substantial as for the first hub in the center. When all 9 hubs are opened, the average cost decreases to 11.70\$. To put these numbers into the right perspective, we compare them to the lower and upper bounds defined in Section 4.4.1. The upper bound is strengthened by using the portion of riders that own a car. The upper bound is 14.00\$ and the lower bound is 8.00\$. We see that when using 9 hubs, the improvement from the no ride-sharing upper bound is doubled compared to when zero hubs are used. Thereby, the objective is almost 20% closer to the lower bound of the cost compared to when zero hubs are used. We emphasize that this lower bound is only attained if every rider can find a perfect match. Therefore, attaining this lower bound is highly unlikely in realistic scenarios where the number of drivers is not infinitely large. For example, when the number of drivers is 2500 (5 drivers for every rider) the costs only decrease to 10.60\$ (the purple line in Figure 4.4b).

Note that as the number of private vehicles used decreases, it is expected to have a further decrease in travel times due to a decrease in congestion. We do not include this effect in our analysis as travel times are exogenous, but in reality, the system could create even higher social benefits.

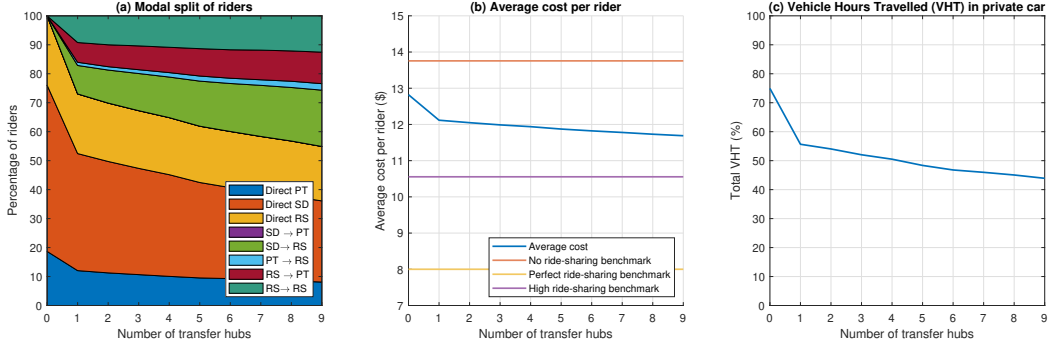


Figure 4.4: Statistics for a varying number of hubs. VHT is given as a percentage of the VHT when ride-sharing is not available as a mode. SD = Solo Drive, PT = Public Transport, RS = Ride-Share.

Figure 4.5 displays how the mode choice changes when the number of hubs changes. Although the majority of the mode choices remain the same, some significant movements can be observed. For example, riders that drove their own car without transfer hubs mostly change to ride-share on both legs or to use their own car on the first leg and ride-share on the second leg. Riders that used to take public transport without transfer hubs, change either to ride-share on both legs or on a single leg while using public transport on the other. Former ride-sharers may change to any of the modes, abandoning their direct ride. As an effect of these changes, we also observe some riders that used their own car or public transport move towards a direct ride-share and vice-versa.

4.4.3 Spatio-temporal distribution of riders

In this section, we evaluate the spatial-temporal distribution of riders. First, we classify riders by their desired arrival time and the mode they use to commute. The results are displayed in Figure 4.6 where the left-hand panel displays the number of riders using every mode and the right-hand panel displays the proportion of riders using every mode (i.e., scaled by the number of riders with that desired arrival time). To obtain these results, 100 simulated instances have been used with 1 hub in the center and 4 on the second ring road (identified by 1-5 in Figure 4.3).

It is clear that the proportion of riders traveling solo is the highest in the tails. The reason for this is that the number of potential matches with identical origins and destinations and similar desired arrival times is low since the number of individuals here is rather low. This effect is more apparent for riders with an early desired arrival time. When these riders match to a driver, it is highly likely that the desired arrival time of the driver is later than that of the rider, and therefore the rider will suffer from lateness. As lateness is penalized heavier than earliness, the effect is more apparent at the start of the morning commute than it is at the end. As the

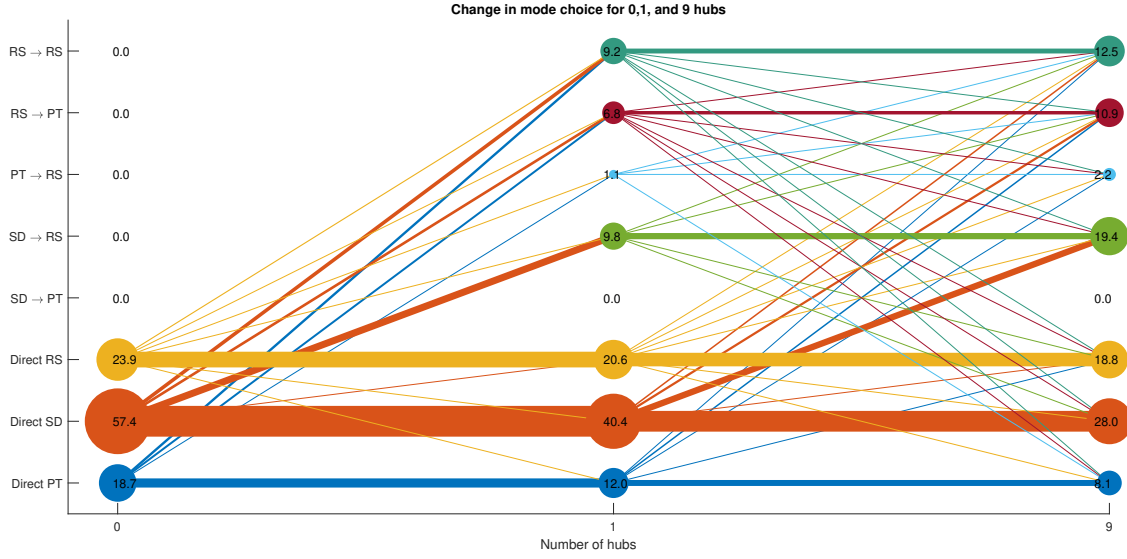


Figure 4.5: Change in mode choice for 1, 2, and 9 transfer hubs. The vertical axis displays the modes, horizontal axis displays the number of hubs. The size of a bubble depicts the number of riders using that mode and the thickness of the lines depicts how many riders change from one mode to another when the number of hubs changes. SD = Solo Drive, PT = Public Transport, RS = Ride-Share.

value of β approaches the value of γ the distribution gets more symmetric. At the peak of the rush hour (i.e., around 8:00 when most commuters have their desired arrival time), the number of ride-sharers is the highest. We see a skewness towards later desired arrival times, which follows the same reasoning as stated before. By changing the number of hubs, the modal share of each mode changes as described in Section 4.4.2. The shape of the distribution on the other hand stays roughly the same while being shifted either up or down depending on the mode.

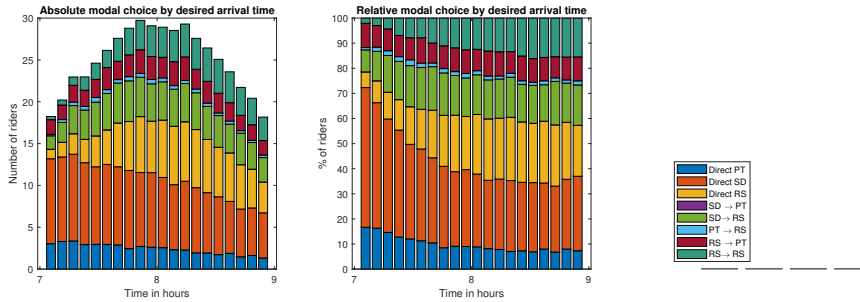


Figure 4.6: Distribution of riders by desired arrival time and mode. SD = Solo Drive, PT = Public Transport, RS = Ride-Share.

For a more detailed analysis of ride-sharing with transfers, we look at the portion of riders that share a ride with a transfer distributed by origin and destination. We consider a network with a single hub in the center of the network. The results

are analyzed in more detail by disaggregating over both origin and destination. Given that the network is symmetric in all interior roads, we only distinguish between the four rings, but not the nodes on the ring. That is, the network can be rotated without changing the distribution. The results are shown in Figure 4.7. Riders that share a ride at a transfer generally have an origin at one side of the transfer hub and a destination on the opposite side, approximately. The reason for this is that the detour imposed by the transfer hub is relatively small for those origin-destination combinations. Furthermore, we observe there is a higher concentration of origins and destinations closer to the center.

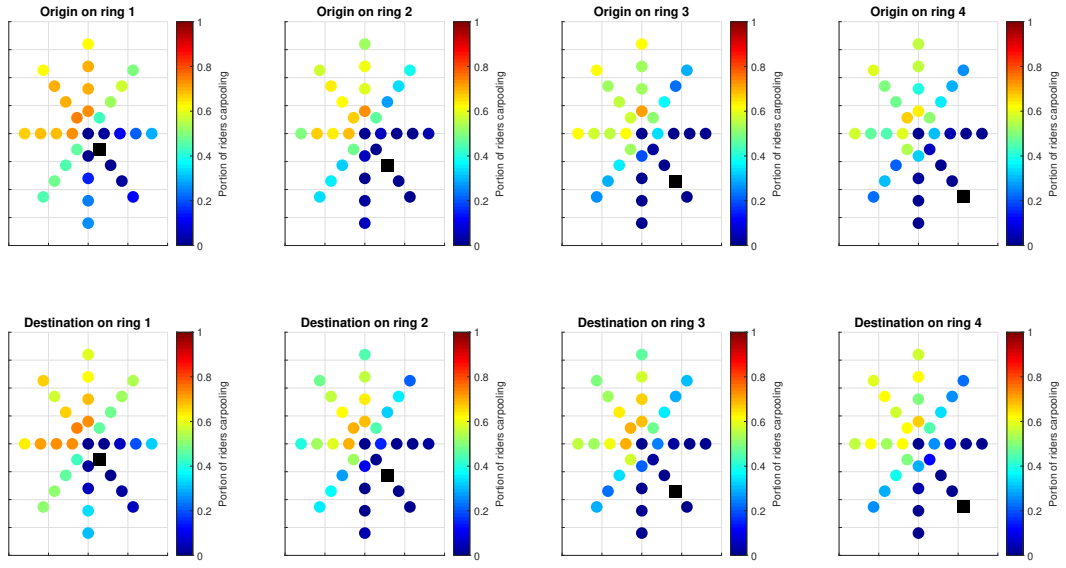


Figure 4.7: Proportion of riders sharing a ride with a transfer for a network with a single transfer hub in the center. The destination and origin, for the top and bottom respectively, are marked by a black square.

4.4.4 Sensitivity analysis

We perform a sensitivity analysis to evaluate the effect of the driver's car capacity ($q_j = [1, 2, 3, 4]$), the maximum detour drivers are willing to make to reach a transfer hub ($\tau_j = [0 \text{ min}, 5 \text{ min}, 10 \text{ min}, 15 \text{ min}]$) and the ratio of drivers and riders ($\frac{|J|}{|I|} = [0.5, 1.0, 1.5, 2.0]$). We evaluate the effect using the base parameters as described before and a total of 5 transfer hubs, changing only the specific parameter to evaluate the sensitivity of the solution, while for the others the base value is used. We obtain the average cost per rider for every possible combination on a set of 10 randomly generated instances. The results are displayed in Figures 4.8 and 4.9 that display the average cost per rider and the VHT in a private car for riders (red) and

drivers (blue) with minimum, maximum and mean over the 10 instances.

The results indicate that the number of drivers is the most influential in decreasing costs. The reason for this is that the higher the number of drivers, the higher the probability for a rider to find a good match. Although each of the considered parameters decreases the cost, the marginal effect is diminishing. For the capacity, the effect is diminishing because the pickup and dropoff locations of all riders that are carried simultaneously need to be identical. This is more difficult to arrange when the number of riders increases. The effect of the maximum detour is diminishing because of the finite size of the network. The number of drivers is the most influential, but whenever most riders have found a good match, the effect of each additional driver will also diminish.

Whereas an increase in the capacity of vehicles and the maximum detour drivers are willing to make decreases the total vehicle hours traveled, this is not the case for the ratio of riders and drivers. Because of the increase in the number of drivers, ride-sharing participation among riders increases which therefore decreases their VHT in a private car. However, by increasing the number of drivers, their VHT increases proportionally. Clearly, the decrease in VHT of riders does not offset the increase in VHT of drivers. Interestingly, whereas the capacity and the maximum detour increase the proportion of riders that use a transfer, an increase in the number of drivers leads to a decrease in this proportion. This means that a higher number of drivers leads to a higher number of direct ride-sharing matches, rather than the number of matches with transfers.

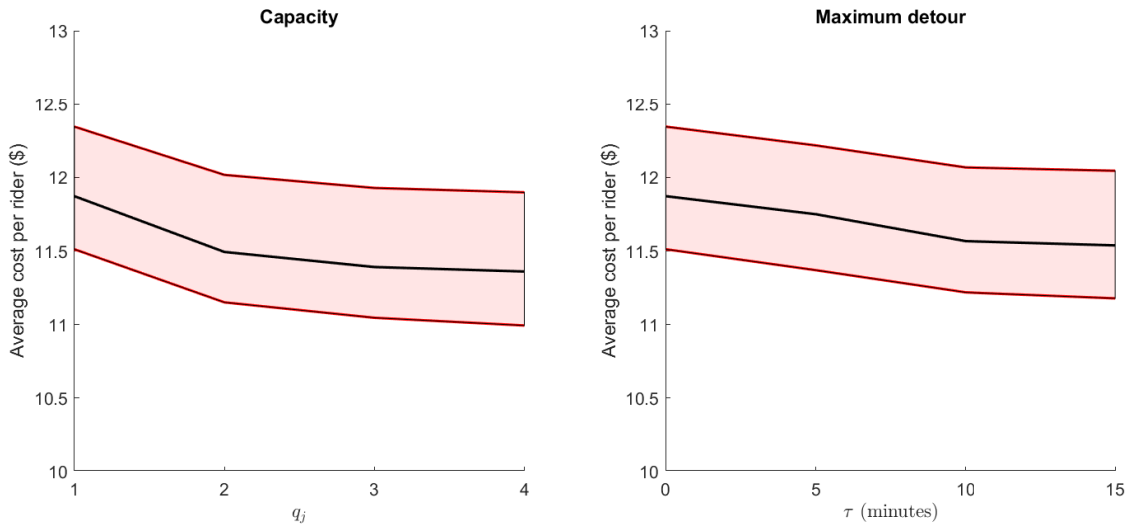


Figure 4.8: Effect of driver capacity (q_j) and maximum detour τ on the average cost per rider. The red shaded area displays the difference between the maximum and minimum average cost among the 10 random instances and the black line displays the mean.

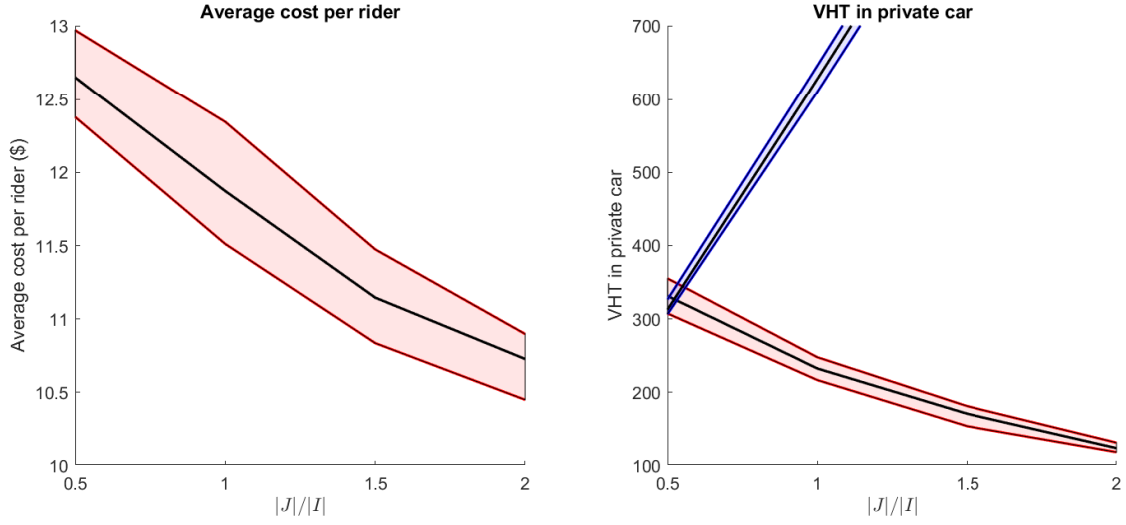


Figure 4.9: Effect of the ratio of drivers and riders on the average cost per rider and the total VHT. The shaded area displays the difference between the maximum and minimum average cost/vht among the 10 random instances and the black line displays the mean. The blue area corresponds to the driver and the red area corresponds to the rider.

4.4.5 Stochastic programming results

In this section, we analyze the results of the stochastic programming problem. Three scenarios have been used ($|\Omega| = 3$) for five different ranges of the uncertainty set. We choose $\Delta_\omega \in \{0.0, 0.1, \dots, 0.5\}$ where the scenarios have the following variations: $[tt(1 - \Delta_\omega), tt, tt(1 + \Delta_\omega)]$. In this way, we can evaluate the influence of increasing uncertainty in travel times on the average cost, modal share, and similarity across scenarios. Thereby, we evaluate the effect of the availability of information on the solution by comparing it to two benchmarks. The results are evaluated on the network with a single hub in the center of the network and parameter settings as described in Section 4.4.1.

A comparison of the expected value, stochastic programming, and wait-and-see problem gives more insight into the importance of information. The expected value problem **(P4)** does not allow to change the solution based on information about the realized scenario, the stochastic programming problem **(P2)** allows to change second-stage decisions only and the wait-and-see problem **(P3)** allows to adapt all decisions to the observed scenario. Based on this, **(P4)** can be seen as a limited information regime (note that it is limited but not zero, as the distribution of travel times is still known), **(P2)** as a partial information regime, and **(P3)** as a full information regime. Figure 4.10a) shows the effect of information on the objective, where consistent with the theory in Section 4.3.3 the solution to **(P4)** forms an upper bound and **(P3)** forms a lower bound. The difference between the solution to the stochastic programming problem and the bounds increases as the variance

increases. Whereas the objective of **(P2)** and **(P4)** increases with the variance, the objective of **(P3)** decreases with Δ_ω . The reason for this is the independence of public transport to the scenario ω that causes asymmetry in the effect of positive and negative travel time changes in the wait-and-see problem. As the public transport option has constant costs, once the cost of the alternative exceeds the cost of public transport, riders will abandon their previous choice and use public transport instead. Overall, a comparison of these three problems indicates the benefit of information and flexibility to change decisions at the transfer hub.

Figure 4.10b) displays the modal split of the riders for the 10 considered mode combinations. In general, increasing the variance has two main effects. The first is that it becomes less attractive for drivers to drive in their own car, because they may face high travel times. The second effect is that those modes that are by definition suboptimal in the deterministic case ($PT \rightarrow PT$ and $SD \rightarrow SD$) may now be optimal because they are combined with other modes in other scenarios. That is, passing through a transfer point gives the rider additional flexibility to change their decision after the travel times are known. A solo-driving rider that passes through the transfer point may choose to continue with public transport or ride-sharing if travel time is high, or continue solo-driving if travel time turns out to be low. This flexibility makes indirect modes more attractive than direct modes.

By increasing the variance, the similarity across decision-making in the various scenarios also decreases. Clearly, direct trips and the first leg of indirect trips are fixed (the modal share indicates that approximately half of the trips are direct trips). The second leg of a trip that includes a transfer can be changed according to the observed scenario. In Figure 4.10c), the blue bars depict the percentage of riders that use the same mode in each scenario. The red bar displays the number of riders who share a ride with the same driver in all scenarios as a percentage of those that use the same mode in all scenarios and share a ride on their second leg but not necessarily with the same driver. Without uncertainty, 100% of the matches are the same. This decreases to approximately 40% for a Δ_ω of 50%. In that case, half of the riders who share a ride on the second leg use a different driver for some scenarios. The results show the flexibility the riders use, especially when the variance of the uncertainty set is higher. For the mode choice, over 20% of the riders change their mode choice when Δ_ω is 50%.

We compare in more detail the stochastic programming solution to the solution of the two benchmark models and the deterministic alternative. We specifically focus on the model for which $\Delta_\omega = 30\%$. Figure 4.11 compares the modal share of each mode and describes how the riders change mode for different levels of information. The size of the bubble describes the percentage of the modal share of each mode. A large shift is observed from solo driving to public transport between

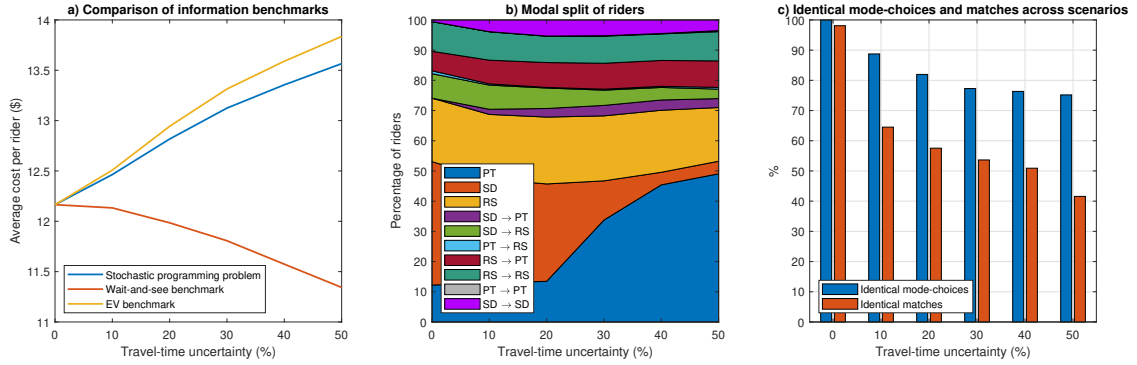


Figure 4.10: Comparison of stochastic programming solutions to two benchmarks for $\Delta\omega$ between 0% and 50%. Panel a) compares the average cost per rider of the stochastic programming problem to the wait-and-see benchmark and the expected-value benchmark. Panel b) displays the modal split of riders for the stochastic programming problem and panel c) displays the percentage of riders that have identical mode choices and matches across the three tested scenarios of the stochastic programming problem.

deterministic and stochastic. The reason for this is that public transport is not prone to uncertainty like the other modes. This also causes an increase in the number of riders that make a transfer (or at least pass through the transfer hub). Clearly, the more information riders have, the more likely they are to use their own car for a part of their trip. Figure 4.12 presents the percentage of time spent in each of the three modes. The biggest difference can be observed between deterministic and stochastic. We also observe that ride-sharing is the most resilient with the percentage changing the least across the four settings.

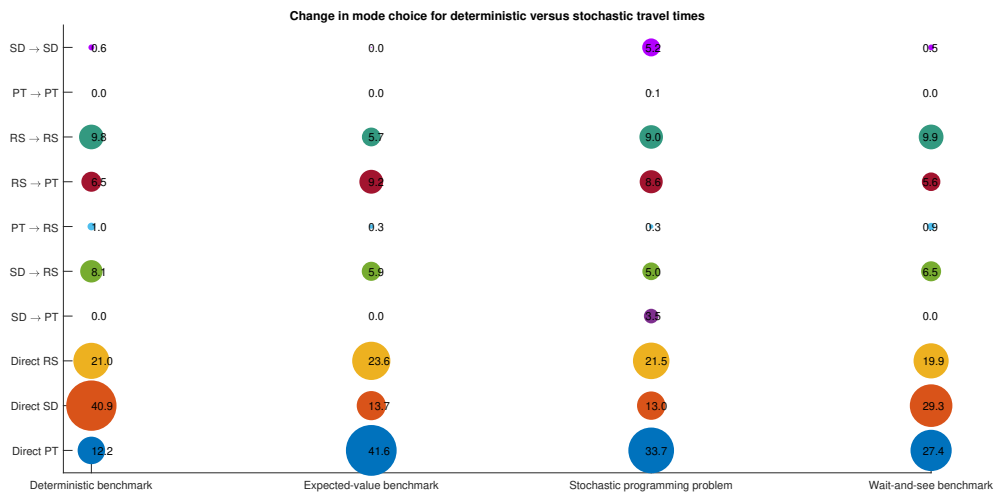


Figure 4.11: Change in mode choice between deterministic and stochastic travel times.

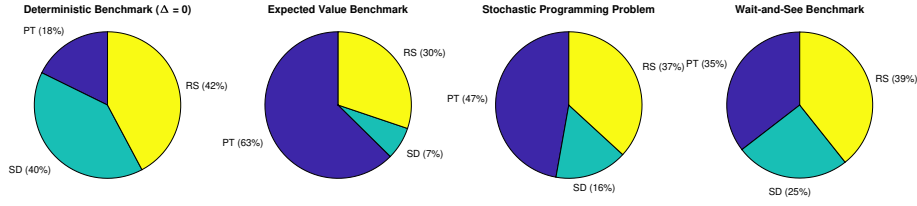


Figure 4.12: Percentage of rider-VHT spent in each mode for deterministic and stochastic travel times.

4.5 Summary

In this chapter, we introduced the multi-modal ride-matching problem with transfers and travel time uncertainty. Ride-sharing, public transport, and private cars were modeled as complementary first- or last-mile modes, as well as competitive modes. Riders can change between two modes as well as between two drivers at designated transfer hubs. These transfer hubs have connections to public transport and have sufficient parking spaces for those reaching the transfer hub with their private car. Travel time uncertainty can affect waiting time and schedule delay penalties and with that the optimal matching. By allowing riders to partially change their decisions at the transfer hub in order to anticipate the observed travel times, significant improvements to their costs can be made.

The deterministic problem with fixed travel times is modeled as a path-based integer programming problem. This problem is extended to a two-stage stochastic programming problem to incorporate uncertain travel times. In the second stage, we allow riders to adapt their path to the observed conditions, while their first-stage decisions remain unchanged.

The results show that with a limited number of transfer hubs, both the average cost per rider and the vehicle hours traveled can be reduced by more than 20%. Contrary to previous studies, our results show that ride-sharing does not only attract riders that were previously using public transport, but it also reduces private car usage by 20%. Especially when travel times are uncertain we observe a large shift from private car usage to public transport, ride-sharing, or multi-modal transport. Due to the flexibility of changing their decisions at the transfer hub, riders can further reduce their costs. Our results indicate that riders frequently change between modes and drivers after observing the actual travel times, ranging between 10 and 60% depending on the variability of the uncertainty set.

Part II

Last-mile logistics systems

5

A continuum approximation approach to the depot location problem in a crowd-shipping system

This chapter is based on the following article:

- P. Stokkink and N. Geroliminis (2023). “A continuum approximation approach to the depot location problem in a crowd-shipping system”. In: *Transportation Research Part E: Logistics and Transportation Review* 176, p. 103207

5.1 Introduction

The success of a crowd-shipping system heavily relies on the availability of crowd-shippers and the potential to match them to demand requests without large detours. The potential pool of crowd-shippers is considered to have a planned personal trip and an associated individual trajectory which is not necessarily near to a parcel trajectory. In this work, we focus on the few-to-many delivery problems of small parcels that are transportable by foot or by bike. If the pickup locations of parcels are poorly accessible by potential crowd-shippers, few parcels can be delivered by crowd-shippers. This can form a major problem for crowd-shipping systems that use in-store pickups. For this reason, we focus on the problem of determining optimal depot locations that function as origins of parcels. Depots are built at central locations in the network such that they are well accessible by crowd-shippers and such that a large portion of the parcel requests can be delivered.

In a depot-based crowd-shipping system, various decisions have to be made to construct a profitable system. These decisions can be divided into strategic, tactical, and operational decisions. A schematic representation of the decision process is illustrated in Figure 5.1. In the first stage, the locations of the depots, where parcels are stored for crowd-shippers to pick them up, are determined. This is a strategic decision that has to be made before the system is operational and is therefore made without full knowledge of the trajectories of potential crowd-shippers and parcels. In the second stage, the assignment of parcels to these depots is determined. This is a tactical decision that is made under full knowledge of the set of parcel requests but only expectations of the trajectories of potential crowd-shippers, fed by historical data. Then, crowd-shippers announce themselves, usually in a dynamic fashion, and the parcels are assigned to them in the third (final) stage. These operational decisions are made daily based on full knowledge of parcels and either full or partial knowledge of crowd-shippers' itineraries. One of the complexities of the considered problem is the uncertainty in the requests for parcels and the availability of crowd-shippers. As many companies offer next-day delivery, demand for parcels is generally only known a day in advance. Crowd-shippers may announce their availability only upon departure from their origin.

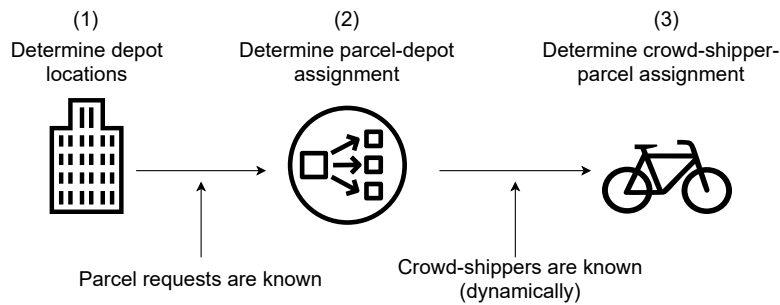


Figure 5.1: Schematic representation of decision process

In this chapter, we develop a framework to determine the best depot locations for a crowd-shipping system in a large urban area. This problem is especially difficult because of the dependency on lower-level decisions and costs on upper-level decisions. To track these interactions, we solve the lower-level assignment problem of parcels to potential crowd-shippers through a Continuum Approximation (CA) approach, allowing us to determine the lower-level costs efficiently in a short time. The reader is referred to Ansari, Başdere, Li, Ouyang, and Smilowitz, 2018 for a recent review of the advancements of CA models for logistics and transportation systems. CA approaches have been widely used for the design of large-scale networks. For example, for the design of vehicle routing problems (Daganzo, 2005; Ouyang, 2007), integrated package distribution systems (Smilowitz and Daganzo, 2007) and pickup-and-delivery problems (Lei and Ouyang, 2018). Thereby, CA has been used for various variants of the FLP, such as the reliable FLP (Cui, Ouyang, and Shen,

2010; Li and Ouyang, 2010) and the competitive FLP (Wang and Ouyang, 2013). Although we note the similarities between the FLP and the depot location problem discussed in this chapter, we note that the main difference is that our problem introduces an additional layer of complexity due to the underlying assignment problem of parcels to crowd-shippers. This complexity does not allow us to evaluate all the lower-level costs for every potential set of depot locations, as is commonly done a priori in an FLP.

Building on the approximated lower-level costs, we use a large neighborhood search heuristic to solve the depot location problem that minimizes the total cost of the crowd-shipping system. We use performance metrics to efficiently search the neighborhoods of solutions. Due to the fast CA approximation, we can find the optimal depot locations in a reasonable time. In addition to this, the CA estimates are used as input to a smart dynamic assignment strategy, that outperforms existing dynamic strategies by leveraging the expectations of future crowd-shippers. A discrete event simulator is used to evaluate the performance of the continuum approximation, the depot-location algorithm, and the dynamic assignment strategy.

The remainder of this chapter is organized as follows. The proposed methods to efficiently solve this multi-level problem, including the continuum approximation of the assignment problem, the depot-location algorithm, and a smart dynamic assignment strategy, are discussed in Section 5.2. In Section 5.3 we describe the discrete event simulator and various assignment strategies for the second and third-stage assignment problems. The results are discussed in Section 5.4, where we evaluate the proposed methodology with comparisons to static and dynamic benchmarks. We compare the performance of our CA-based algorithm against solving a discrete formulation using CPLEX and perform various sensitivity analyses. Performance is evaluated using a discrete event simulator based on a part of the city of Washington DC. The chapter is concluded with a summary in Section 5.5.

5.2 Methodology

In this section, we describe the methodological contributions of this work. In Section 5.2.1 we give a more detailed description of the problem and formulate the problem as an integer stochastic programming problem. In Section 5.2.2 we describe the continuum approximation approach used to approximate the cost in the third-stage assignment problem. In Section 5.2.3 we explain the methods used to identify the depot locations based on the CA results. A notational glossary of the sets, parameters, and variables is provided in Table 5.1. For the sake of readability, a division has been made for parameters that are introduced in the problem description, continuum approximation approach, and large neighborhood search algorithm.

Table 5.1: Notational Glossary

Problem description	
Sets	
C	Set of potential crowd-shippers (indexed c)
D	Set of opened depots (indexed d)
D^p	Set of potential depots (indexed d)
P	Set of parcels requested (indexed p)
Parameters	
q_c	Capacity of crowd-shipper c
D^{\max}	Maximum number of depots that can be built
f_{pcd}	Parameter indicating if crowd-shipper c can feasibly pick up and deliver parcel p , with an origin at depot d
t_{od}	Travel time between node o and node d
ϕ_d^{depot}	Fixed costs of opening a depot
ϕ_p^{cs}	Reward given to crowd-shippers to pick up and deliver parcel p , with an origin at depot d
ϕ_p^{reg}	Cost of regular delivery of parcel p
ϕ_p	Cost of regular delivery of parcel p
$\psi_k(\cdot)$	Function describing the costs of the k^{th} stage for a given set of inputs
$\xi_P(\xi_C)$	Random variable of which the realizations are vectors of parcel requests (potential crowd-shippers)
Decision variables	
x_{pc}	Decision variable indicating whether parcel p is matched to crowd-shipper c
y_{pd}	Decision variable indicating whether parcel p is assigned to depot d
z_d	Decision variable indicating whether depot d is opened
Continuum Approximation Algorithm	
Sets	
\tilde{C}	Set of crowd-shipper classes
R	Set of regions
Parameters	
e_{rcd}	Binary parameter indicating if crowd-shippers of class c can feasibly pick up a parcel at depot d and deliver it to region r
\tilde{e}_{rc}	Binary parameter indicating if crowd-shippers of class c can feasibly pick up a parcel at <i>at least one</i> depot and deliver it to region r
$or(c)$	Origin of crowd-shippers in class c
$dest(c)$	Destination of crowd-shippers in class c
μ_r	Number of parcels with destination in region r
$\hat{\mu}_r$	Expected number of parcels with destination in region r
λ_c	Actual number of potential crowd-shippers of class c
$\hat{\lambda}_c$	Expected number of potential crowd-shippers of class c
τ_c	Maximum detour crowd-shippers of class c are willing to make to pick up and deliver a parcel
ϕ_p^{cs}	Reward given to crowd-shippers to perform a pickup at depot d and delivery to region r
ϕ_p^{reg}	Cost of regular delivery of a parcel to region r
ϕ_r	Cost of regular delivery of a parcel to region r
Variables	
a_{rd}	Number of parcels with final destination r stored at depot d
$\mathcal{C}(D)$	Total expected cost of opening the set of depots D
l_r	Leftover parcels with a destination in region r in iterative CA procedure
u_r	Intermediate estimate for the expected number of parcels delivered by crowd-shippers to region r
v_r	Expected number of parcels delivered by crowd-shippers to region r
$v_r(D)$	Total expected number of parcels delivered by crowd-shippers to region r for a set of depots D
x_{rcd}	Decision variable indicating how many parcels with a destination in region r are assigned to depot d and crowd-shipper class c
$\tilde{\mu}_r$	Remaining expected number of parcels with destination in region r in iterative CA procedure
$\tilde{\mu}_c$	Total number of parcels that can potentially be served by crowd-shippers of class c
$\tilde{\lambda}_c$	Remaining expected crowd-shippers of class c in iterative CA procedure
Large Neighborhood Search Algorithm	
sd_1d_2	Similarity of depots d_1 and d_2
m_d	Quality metric for opening single depot d
α, β, γ	Tuning parameters
η	Number of initial solutions in depot-location algorithm
κ	Iteration limit of depot-location algorithm
Ω	Current solution (i.e., set of depots) in the depot-location algorithm

5.2.1 Discrete formulation

The problem as described in Figure 5.1 can be formulated as an integer stochastic programming problem. In this way, we can incorporate the two types of uncertainty by dividing the problem into three levels. The formulation is similar to that of Mousavi, Bodur, and Roorda, 2022. The main difference is that we consider an additional layer of uncertainty (uncertainty in demand) which makes our problem a three-stage stochastic programming problem, compared to the two-stage stochastic programming problem proposed by Mousavi, Bodur, and Roorda, 2022. In addition to this, crowd-shippers may carry multiple parcels in our formulation, as opposed to them carrying only one parcel in the formulation by Mousavi, Bodur, and Roorda,

2022.

We consider a set of demand requests for small parcels $p \in P$, which is a realization drawn from random variable ξ_P . Similarly, we consider a set of crowd-shippers $c \in C$, which is a realization drawn from random variable ξ_C . Thereby, we consider the set of potential depots D^p and binary decision variables z_d for all $d \in D^p$ equal to 1 if depot d is opened and 0 otherwise. The set of opened depots is hereafter referred to as D and we continue to use D^p for the set of potential depots, that are not necessarily opened. A depot can be opened at a cost ϕ^{depot} and a maximum of D^{max} depots can be opened. The value of D^{max} is a modeling choice related to the maximum capital investment a crowd-shipping operator is willing to make. The fixed cost of opening a depot mainly consists of the daily rental costs of a location and the maintenance costs of parcel lockers. Acquiring the parcel lockers is a one-time investment and is therefore neglected. The costs involved with the second and third stages depend on earlier decisions, as well as the realization of parcels and crowd-shippers, and are denoted by $\psi_2(z, P)$ and $\psi_3(z, y, P, C)$ respectively. The first-stage objective is to minimize the sum of the costs of opening depots and the expected costs of the second and third stages. We denote with $\mathbb{E}_\xi[\cdot]$ the expected value function over random variable ξ . The first stage can be formulated as follows:

$$\text{minimize } \phi^{\text{depot}} \sum_{d \in D^p} z_d + \mathbb{E}_{\xi_P}[\psi_2(z, P)], \quad (5.1)$$

$$\text{s.t. } \sum_{d \in D^p} z_d \leq D^{\text{max}}, \quad (5.2)$$

$$z_d \in \mathbb{B} \quad \forall d \in D^p. \quad (5.3)$$

The objective (5.1) is to minimize the total costs consisting of fixed costs for every opened depot and the operational costs. The operational costs are an expected value of the second- (and indirectly third-) stage costs. Constraint (5.2) enforces that at most D^{max} depots can be opened.

In the second stage, we decide which parcel to assign to which depot. For this, we introduce binary decision variables y_{pd} which is equal to 1 if parcel $p \in P$ is assigned to depot $d \in D^p$, and 0 otherwise. We assume there are no costs involved with the second stage (at least, there is no direct cost difference between assigning to different depots) other than the expected costs of the third stage. We consider small and portable parcels, such that depot capacity can be disregarded. This

leads to the following formulation of the second stage:

$$\psi_2(z, P) = \mathbb{E}_{\xi_C}[\psi_3(z, y, P, C),] \quad (5.4)$$

$$s.t. \sum_{d \in D^p} y_{pd} \leq 1 \quad \forall p \in P, \quad (5.5)$$

$$y_{pd} \leq z_d \quad \forall p \in P, d \in D^p, \quad (5.6)$$

$$y_{pd} \in \mathbb{B} \quad \forall p \in P, d \in D^p. \quad (5.7)$$

The second stage costs only consist of the expected costs of the third stage. Every parcel is assigned to at most one depot, which is enforced by Constraints (5.5). Constraints (5.6) ensure that a parcel is only assigned to an opened depot.

Finally, in the third stage, the parcels are assigned to crowd-shippers. For this, we introduce binary decision variables x_{pcd} , which is equal to 1 if parcel $p \in P$ is assigned to crowd-shipper $c \in C$ and depot $d \in D^p$, and 0 otherwise. The crowd-shipper fee for every parcel is denoted by ϕ_{pd}^{cs} and depends on the distance between the origin depot d and the destination of parcel p . Specifically, $\phi_{pd}^{cs} = \phi_p^{cs,1} + \phi_p^{cs,2} \cdot t_{d,dest(p)}$ where the first term is a fixed compensation per delivery and the second term is a variable compensation depending on the distance between the origin and destination of the parcel. The cost of regular delivery of parcel $p \in P$ is defined as ϕ_p^{reg} and comprises all costs associated with last-mile delivery such as fuel cost, driver salary, and cost of maintenance and repair. It is clear that for a parcel p , crowd-shipping is only favored over regular delivery if $\phi_{pd}^{cs} < \phi_p^{reg}$. A parcel can be assigned to only one crowd-shipper and a crowd-shipper can be assigned at most q_c parcels. Although a crowd-shipper can carry multiple parcels, we assume these parcels need to have identical origins and destinations, to limit the inconvenience of pickup and delivery that a crowd-shipper encounters. Thereby, a parcel can only be assigned to a crowd-shipper if they can feasibly pick up and deliver this parcel, given the depot the parcel was assigned to in the second stage. We use a binary parameter f_{pcd} which is equal to 1 if crowd-shipper $c \in C$ can feasibly pickup parcel $p \in P$ from depot $d \in D^p$ and deliver it to the final destination of the parcel. A potential crowd-shipper $c \in C$ is assumed to have a maximum detour τ_c he/she is willing to make to pick up and deliver a parcel, which defines this feasibility parameter. The third stage can then be formulated as:

$$\psi_3(z, y, P, C) = \sum_{p \in P} \phi_p^{reg} + \sum_{p \in P} \sum_{c \in C} \sum_{d \in D^p} x_{pcd} (\phi_{pd}^{cs} - \phi_p^{reg}), \quad (5.8)$$

$$\sum_{p \in P} \sum_{d \in D^p} x_{pcd} \leq q_c \quad \forall c \in C, \quad (5.9)$$

$$\sum_{c \in C} \sum_{d \in D^p} x_{pcd} \leq 1 \quad \forall p \in P, \quad (5.10)$$

$$x_{pcd} \leq y_{pd} f_{pcd} \quad \forall c \in C, p \in P, d \in D^p, \quad (5.11)$$

$$x_{p_1 c d_1} + x_{p_2 c d_2} \leq 1 \quad \forall c \in C, p_1, p_2 \in P, d_1, d_2 \in D^p : \quad (5.12)$$

$$(d_1 \neq d_2 \parallel \text{dest}(p_1) \neq \text{dest}(p_2)),$$

$$x_{pcd} \in \mathbb{B} \quad \forall c \in C, p \in P, d \in D^p. \quad (5.13)$$

The operational costs of the third stage in Equation (5.8) are made up of penalties for regular delivery and compensations awarded to crowd-shippers. Without crowd-shipping, all parcels would be delivered by regular delivery vehicles, therefore incurring penalties $\sum_{p \in P} \phi_p^{\text{reg}}$. Every parcel that is delivered by crowd-shippers then costs ϕ_{pd}^{cs} for compensation, but reduces the penalties by ϕ_p^{reg} . Every shipper can be assigned at most q_c parcels and every parcel can be assigned to at most one crowd-shipper, which is enforced by Constraints (5.9) and (5.10) respectively. Thereby, through Constraints (5.11), we ensure that a parcel should be picked up from the depot to which it was assigned in the second stage and that the match is feasible. Constraints (5.12) enforce that only parcels with identical origins and destinations are allowed to be carried by the same crowd-shipper.

The difficulties of solving this problem in (5.1) - (5.13) are three-fold. First, we are dealing with two separate layers of uncertainty. The first layer of uncertainty is in the parcels; with next-day delivery being extremely common, the number of parcel requests in every region is uncertain up to a day before delivery. The second layer of uncertainty is in the crowd-shippers. The number of crowd-shippers and their itineraries are generally uncertain up to shortly before the departure of the crowd-shipper. Whereas some crowd-shippers may know their schedule well in advance, others may only make themselves available a few minutes before departure. Thereby, exact schedules may be prone to last-minute changes. The second difficulty is that decisions in each of the three stages are heavily intertwined. Decisions in the first and second stages are made based on expected costs and actions in the third stage, whereas the optimal third-stage decisions and corresponding costs depend on the decisions that were made in the first and second stages. This illustrates the importance of solving the problem as a whole and the inability to decompose it. The third difficulty is that the large size of the problem in urban areas causes a computational burden.

These difficulties combined make it impossible for the problem to be solved exactly for a realistic case study. Therefore, in this work, we approximate the second and third-stage operational costs using a continuum approximation approach. Based on

the approximated operational costs, we optimize the first-stage strategic decisions. By using an approximation of the second and third stages, we can evaluate many potential depot combinations within a reasonable amount of time and therefore explore a large search space.

5.2.2 Continuum approximation

To approximate the third-stage costs, as well as estimate the served parcels in every region of the network, we use a CA approach. Rather than using a formulation based on individual crowd-shippers and parcels like in Section 5.2.1, we reformulate the second and third stages to a region-based formulation. We consider a network split into R regions. Daily demand for small parcels in every region $r \in R$ is equal to μ_r . To allow for heterogeneity among crowd-shippers, potential crowd-shippers are divided into classes. We denote \tilde{C} as the set of crowd-shipper classes, not to be confused with the discrete set of crowd-shippers C . Every class $c \in \tilde{C}$ corresponds to a homogeneous group of crowd-shippers with origin $or(c)$, destination $dest(c)$, and a maximum detour τ_c . We note that additional heterogeneity may be added, such as maximum distance traveled or value of time, but this is omitted in this work. The number of crowd-shippers in class c is denoted by λ_c . For the sake of approximation, a crowd-shipper with capacity q_c is counted as q_c separate crowd-shippers in λ_c .

We define parameter e_{rcd} which is equal to 1 if a crowd-shipper of class c can pick up a parcel at depot d and deliver it to the final destination in region r , and 0 otherwise. When $e_{rcd} = 1$, this is referred to as a feasible assignment. For all feasible assignments for which crowd-shipping is more expensive than regular delivery ($\phi_{rd}^{cs} > \phi_r^{reg}$) we set $e_{rcd} = 0$. We remark the relation between the parameter e_{rcd} and the parameter f_{pcd} in the discrete formulation. One of the main advantages of this reformulation is that the computation of the variables e_{rcd} relies only on the size of the network (that is, the number of regions) and the level of heterogeneity and not on the number of crowd-shippers nor the number of parcels. Thereby, it is independent of the realizations of parcels and crowd-shippers, whereas f_{pcd} depends on the realizations of the uncertain sets P and C .

Geometrically, the matching problem and the definition of e_{rcd} can be interpreted through two ellipses, as depicted in Figure 5.2. The original route from the origin to the destination of a crowd-shipper is depicted by the black line. The crowd-shipper has to make a detour by performing a pickup at d and delivery at the final destination of the parcel r . For this to be feasible within the maximum detour τ_c , the following should hold. The depot location d should lie within the ellipse with focus points $or(c)$ and $dest(c)$, where the distance between the focus and the closest vertex is equal to τ_1 . Thereby, the final destination r should lie within a second ellipse with focus points d and $dest(c)$, where the distance between the focus

and the closest vertex is equal to τ_2 . If $\tau_1 + \tau_2 \leq \tau_c$, the pickup and delivery can be made within the maximum detour. The difficulty of this problem is marked by the variability in the values of τ_1 and τ_2 under the constraint $\tau_1 + \tau_2 \leq \tau_c$. In addition to this, there is a dependency between the size of the two ellipses. As the point d has to lie within the black ellipse, it restricts the size of the blue ellipse. We also note that we cannot omit the second ellipse by simply considering the line segment between d and $dest(c)$ inside the black ellipse instead of just point d , because of the influence of the direction of this segment on the total detour. Due to these difficulties, we resort to numerical approximations.

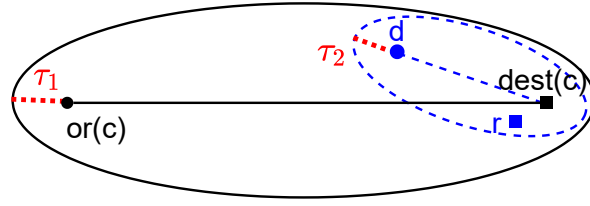


Figure 5.2: Geometric interpretation of matching problem

LP approximation

We use a region-based approximation of the discrete formulation in Section 5.2.1. We consider region-based cost parameters ϕ_r^{reg} for regular delivery to region $r \in R$ and ϕ_{rd}^{cs} for crowd-shipped delivery to region $r \in R$ from depot $d \in D$. We define decision variables $x_{r cd}$ as the number of crowd-shippers of class $c \in \tilde{C}$ that perform a delivery from depot $d \in D$ to region $r \in R$. The approximation can then be formulated using the following Linear Programming (LP) problem:

$$\text{maximize } \sum_{c \in \tilde{C}} \sum_{r \in R} \sum_{d \in D} (\phi_r^{\text{reg}} - \phi_{rd}^{\text{cs}}) x_{r cd} \quad (5.14)$$

$$\sum_{c \in \tilde{C}} \sum_{d \in D} x_{r cd} \leq \hat{\mu}_r \quad \forall r \in R \quad (5.15)$$

$$\sum_{r \in R} \sum_{d \in D} x_{r cd} \leq \hat{\lambda}_c \quad \forall c \in \tilde{C} \quad (5.16)$$

$$x_{r cd} \in \mathbb{R} \cap [0, e_{r cd} \min(\hat{\mu}_r, \hat{\lambda}_c)] \quad \forall c \in \tilde{C}, r \in R, d \in D \quad (5.17)$$

The objective is to minimize the total operational costs. Here, we simplified the objective function in (5.14) to only account for the delivered parcels. All parcels that are not delivered by crowd-shippers incur a cost for regular delivery. Constraints (5.15) and (5.16) ensure that not more deliveries are made than there are parcels requested and not more crowd-shippers are used than there are available. Constraints (5.17) define the range of the decision variables and ensure that only

feasible deliveries are made.

Although this drastically simplifies the second and third layer of the discrete problem in Section 5.2.1 computation times and memory consumption for constructing and solving the LP problem can still be problematic for large networks. Thereby, this formulation ignores the level of uncertainty between hub assignment (second-stage decisions) and crowd-shipper assignment (third-stage decisions). This generally leads to an underestimation of the costs in a more realistic dynamic setting with uncertainty. Therefore, we next propose an algorithmic approximation that encompasses these two aspects.

Algorithmic approximation

For the sake of approximation, we treat the second and third-stage decisions in reverse order. First, we merge all depots and approximate the third-stage assignment. Second, based on the third-stage approximations we approximate the second-stage parcel-depot assignments. The second-stage approximations are needed because of their influence on the costs ϕ_{rd}^{cs} .

When multiple depots are opened, the main difficulty is that both parcels and crowd-shippers have to be split over the various depots. Only when the expected number of parcels delivered by a specific depot is independent of the other depots, this problem can be separated into subproblems. However, especially when two potential depots are close together, it is clear that this independency is not true. When multiple depots are opened, each parcel can originate from multiple depots and each crowd-shipper can pick up parcels from multiple depots as long as they are within their maximum detour τ_c . We first relax the second stage assignment of parcels to depots. We define \tilde{e}_{rc} which is equal to 1 if a crowd-shipper with origin at $or(c)$ and destination at $dest(c)$ can pick up a parcel *from at least one open depot* and deliver it to the final destination in region r , and 0 otherwise. Specifically:

$$\tilde{e}_{rc} = \min(1, \sum_{d \in D} e_{r cd}) \quad \forall c \in \tilde{C}, r \in R. \quad (5.18)$$

We want to approximate the number of parcels delivered from a given set of depots to all regions separately. To approximate this, we let $\hat{\mu}_r$ be the expected number of parcels with a destination in region r and let $\hat{\lambda}_c$ the expected number of crowd-shippers in class $c \in \tilde{C}$. Then, we define $\bar{\mu}_c = \sum_{r \in R} \tilde{e}_{rc} \hat{\mu}_r$ as the total number of parcels that can potentially be served by crowd-shippers of class c . For the sake of the approximation, we assume that a crowd-shipper is equally likely to choose any of the parcels they can feasibly deliver. Following from this, the probability that he picks a parcel with destination region r is equal to $\frac{\hat{\mu}_r}{\bar{\mu}_c}$ if $\tilde{e}_{rc} = 1$ and 0

otherwise. We can then consider all potential crowd-shippers to obtain the following estimation of the number of served parcels:

$$u_r = \sum_{c \in \tilde{C}} \tilde{e}_{rc} \hat{\lambda}_c \frac{\hat{\mu}_r}{\bar{\mu}_c} \quad \forall r \in R. \quad (5.19)$$

Here, we sum over all classes of potential crowd-shippers. A class is only considered if the assignment is feasible. Every crowd-shipper in $\hat{\lambda}_c$ is considered to make the delivery with the same probability $\frac{\hat{\mu}_r}{\bar{\mu}_c}$. If no crowd-shippers can be assigned to a region r , that is either $\hat{\lambda}_c = 0$ or $\tilde{e}_{rc} = 0$ for every origin-destination pair, then the number of delivered parcels will always be zero. Similarly, if the expected number of parcels $\hat{\mu}_r$ is zero, u_r will also be zero. We also note that it is possible for $\bar{\mu}_c = 0$, in which case Equation (5.19) is undefined. In this case, $\frac{\hat{\mu}_r}{\bar{\mu}_c}$ is naturally set to 0.

If $\hat{\mu}_r$ is non-zero, crowd-shippers with different origin-destination pairs may be assigned to the same parcel-destination region r . As this could lead to an overestimation of the number of served parcels in that region ($u_r > \hat{\mu}_r$), we take into account that at most $\hat{\mu}_r$ parcels can be delivered to a region r . Therefore, we define v_r which is the minimum of those two.

$$v_r = \min(\hat{\mu}_r, u_r) \quad \forall r \in R. \quad (5.20)$$

Especially if the number of crowd-shippers is high, by overestimating u_r in region r (i.e., $u_r > \hat{\mu}_r$), it is likely that $u_{r'}$ for another region $r' \neq r$ will be underestimated. Therefore, we use an iterative process to ensure that this unavoidable overestimation is accounted for in the other regions. We consider the leftover number of parcels $l_r = \max(0, u_r - \hat{\mu}_r)$ and split it evenly over the potential crowd-shippers. Similar to the assignment of parcels to crowd-shippers, we assume that every crowd-shipper that can be feasibly assigned to a region r (that is, every crowd-shipper of class c for which $\tilde{e}_{rc} = 1$), is equally likely to be assigned to one of the leftover parcels in l_r . Therefore, the l_r leftover parcels are split over the origin-destination pairs proportional to the number of crowd-shippers that could be feasibly assigned to region r . We define the leftover crowd-shippers as follows:

$$\hat{\lambda}'_c = \sum_{r \in R} l_r \frac{\tilde{e}_{rc} \hat{\lambda}_c}{\sum_{c \in \tilde{C}} \tilde{e}_{rc} \hat{\lambda}_c} \quad \forall c \in \tilde{C}. \quad (5.21)$$

Thereby, we define the undelivered parcels $\hat{\mu}'_r = \hat{\mu}_r - v_r$. All parcels that are already expected to be served by previously assigned crowd-shippers no longer need to be considered and are therefore disregarded. We then compute u_r according to Equation (5.19), but now using $\hat{\lambda}'_c$ and $\hat{\mu}'_r$ as inputs instead of $\hat{\lambda}_c$ and $\hat{\mu}_r$. Using

these we find an additional portion of parcels that can be delivered and we update the estimated number of delivered parcels and the leftover parcels as follows:

$$v_r = \min(\hat{\mu}_r, v_r + u_r) \quad \forall r \in R. \quad (5.22)$$

$$l_r = \max(0, u_r - \hat{\mu}'_r) \quad \forall r \in R. \quad (5.23)$$

This iterative process can be repeated until the number of leftover parcels l_r is zero for all regions $r \in R$. Intuitively, if $l_r = 0$, all previously overestimated delivered parcels have been compensated for. However, we emphasize that the simplifying assumptions of proportional assignment in Equation (5.19) and (5.21) can lead to suboptimal assignments.

As the costs ϕ_p^{cs} depend on the distance between the depot and the final destination of the parcel p , we make an approximation of how many parcels are at which depot through the previously approximated total deliveries. That is, we approximate the second-stage decisions that we previously relaxed to approximate the operational costs. Let $v_r(D)$ be the number of parcels delivered to region r if depots D are opened. We recall ϕ_{rd}^{cs} as the compensation of crowd-shippers making a delivery between depot d and destination region r . Thereby, we define a_{rd} the number of parcels with final destination $r \in R$ that are stored at depot $d \in D$ as follows:

$$a_{rd} = \hat{\mu}_r \frac{v_r(\{d\})}{\sum_{d \in D} v_r(\{d\})}. \quad (5.24)$$

This approximation assumes that a parcel is not necessarily assigned to the closest depot, but may be assigned to a further depot. In Section 5.4.6 we show that such an assignment is substantially better than a closest-depot assignment. The reason for this is that the flow of crowd-shippers is often not homogeneous across the network. In that case, storing a parcel at a depot further away may increase the likelihood of it being delivered by crowd-shippers if the depot and the destination of the parcel (in that order) are on a route that is a more common itinerary for potential crowd-shippers.

The total cost can be obtained directly from the results of the approximation. The total approximated cost, similar to that defined in Section 5.2.1, is as follows. The first term comprises the fixed costs of constructing and maintaining depots. The second term approximates the crowd-shipper compensations by using the total number of delivered parcels derived by the iterative procedure and using the parcel-depot assignment from Equation (5.24). The third term approximates the costs of regular delivery. As the costs of regular delivery are assumed to be equal for all parcels we do not distinguish between depots.

$$\mathcal{C}(D) = \phi^{\text{depot}} |D| + \sum_{r \in R} \left[\sum_{d \in D} \phi_{dr}^{\text{cs}} \frac{a_{rd}}{\sum_{d \in D} a_{rd}} \right] v_r(D) + \phi^{\text{reg}} \sum_{r \in R} (\hat{\mu}_r - v_r(D)). \quad (5.25)$$

This approximation can be easily extended for a probability distribution rather than a single expected value. For example, the distributions ξ_P and ξ_C in Section 5.2.1 can be used to generate multiple scenarios for which the approximation is repeated. The average value can then be used as an approximation of the costs.

5.2.3 Determining depot locations

The cost for a given set of depot locations can be approximated efficiently using the methods discussed in the previous section. Despite this, in large urban networks, the number of options for depot locations to consider can still be extremely large. Specifically, the number of possible combinations grows exponentially with the number of possible depot locations and therefore with the size of the network. Enumerating all options is impossible for large networks and therefore we design an efficient heuristic to determine the best depot locations.

We propose a Large Neighborhood Search (LNS) heuristic to solve this problem. LNS heuristics explore a complex neighborhood to find better candidate solutions (Pisinger and Ropke, 2010). We efficiently explore the neighborhood by using metrics for the quality of solutions, that allow us to select candidates to *destroy* and *repair* in a smart way. The advantage of this algorithm is that we do not need to evaluate the full neighborhood at every iteration but we can use a quality and similarity metric to select a single candidate which is evaluated. Thereby, the high level of randomness allows for diversification of the search and therefore finding robust solutions. The general structure of the algorithm is described in Algorithm 2. The remainder of this subsection describes the initialization of the algorithm and the details of the destroy and repair operators, that exploit the specific features of our problem.

Initialization

To initialize the heuristic, we compute several components that are input to the algorithm. First, we compute the 3-dimensional matrix E with elements e_{rcd} for every potential depot. Thereby, we compute the single-depot objectives for each depot $d \in D^p$ which will be used as a quality metric for the depots. This metric will be referred to as m_d , which is defined as $m_d = \mathcal{C}(\emptyset) + \phi^{\text{depot}} - \mathcal{C}(\{d\})$. The intuition behind this is that a depot that performs well on its own is more likely to perform well in combination with other depots. Nevertheless, depots that are serving parcels with similar destinations and attract crowd-shippers with similar itineraries might perform poorly if they operate together, as one depot has little added value over the other. For this reason, we construct a similarity measure s_{d_1, d_2} for how similar two depots are in terms of the service area of crowd-shippers using that depot. Specifically, s_{d_1, d_2} is determined as follows:

$$s_{d_1, d_2} = \frac{[\sum_{r \in R} \sum_{c \in \tilde{C}} \min(e_{rcd_1}, e_{rcd_2}) \lambda_c]^2}{[\sum_{r \in R} \sum_{c \in \tilde{C}} e_{rcd_1} \lambda_c][\sum_{r \in R} \sum_{c \in \tilde{C}} e_{rcd_2} \lambda_c]}. \quad (5.26)$$

Clearly, two depots that are very similar in terms of the service area are likely to have a lower gain in performance when they are combined. For this reason, this similarity measure will be used to select dissimilar depots to be combined.

We use a multi-start heuristic so that we randomly determine η initial solutions. Every initial solution is generated according to a simple construction heuristic. Every depot is randomly chosen with a probability proportional to the quality of the depot in the single-depot solution. By using a multi-start heuristic we aim to increase the search space and therefore decrease the likelihood of ending up at a local optimum.

Body of algorithm

We terminate the LNS algorithm after a fixed number of κ iterations. In every iteration, we consider the following operations on the current solution Ω and obtain the corresponding objective value. A newly generated solution is always accepted if it is an improvement over the previous solution and a worse solution is never accepted.

1. O_1 - *Repair operator*: For every depot d that is not in the current solution Ω , we compute the following metric: $\frac{m_d^\alpha}{[\sum_{\omega \in \Omega} s_{d, \omega}]^\beta}$ where α and β are tuning parameters that determine the relative importance of single-depot performance and inter-depot similarity. This metric determines the best depot to be added to the current solution, taking into account the quality of the depot in the single-depot solution as well as the similarity to the other depots in the current solution. We use the sum of the similarity with all depots in the current solution Ω , such that a depot that is similar to two depots rather than only one should have a lower metric value. Alternatively, the maximum similarity across all depots in the current solution Ω can be used in the denominator to replace the sum. A new depot is added randomly with a probability proportional to the value of this metric.
2. O_2 - *Destroy operator*: For every depot d that is in the current solution Ω , we determine the same metric as in O_1 and randomly drop a depot with a probability proportional to the inverse of the metric in O_1 . By taking the inverse, depots that are very similar to other depots and depots with relatively low single-depot performance are most likely to be removed.
3. O_3 - *Swapping operator*: A sequential combination of O_1 and O_2 where a depot in the current solution Ω is replaced by another depot that is not in the current solution. We first destroy a depot $\omega \in \Omega$ using destroy operator O_2 and then add a depot using repair operator O_1 , based on the similarity with the remaining depots $\Omega \setminus \{\omega\}$.

Algorithm 2: Large Neighborhood Search Algorithm

```

1 Input: For every depot  $d$  the quality metric  $m_d$  and for every pair of depots
   ( $d_1, d_2$ ) the similarity metric  $s_{d_1, d_2}$ 
2 for  $n \in [1, \eta]$  do
3   Generate an initial solution  $\Omega_n^0$ 
4   for  $k \in [1, \kappa]$  do
5      $\Omega_n^k \leftarrow \Omega_n^{k-1}$ 
6     Destroy a depot in  $\Omega_n^k$  according to  $O_2$ 
7     Repair a depot in  $\Omega_n^k$  according to  $O_1$ 
8     if  $C(\Omega_n^k) > C(\Omega_n^{k-1})$  then
9        $\Omega_n^k \leftarrow \Omega_n^{k-1}$ 
10 return  $\arg \min_{\Omega \in \{\Omega_1^\kappa, \dots, \Omega_\eta^\kappa\}} C(\Omega)$ 

```

5.3 Discrete event simulator

We develop a discrete event simulator to simulate the dynamic operational process. Considering the decision process in Figure 5.1, the depot locations are determined using the methods described in Section 5.2, and the second and third-stage decisions are simulated. We consider various assignment strategies both for the second and third stage decisions, of which we evaluate the performance in Section 5.4.6.

The simulator is initialized by generating a set of parcel requests consisting of only a destination region (the origin is at one of the depots and will be determined in the second stage) and a set of potential crowd-shippers consisting of an origin and destination region and a starting time of the trip. For the sake of this simulation, crowd-shippers are assumed to make themselves available at the start of their trip. All generated parcel requests are assigned to a depot based on one of the strategies described in Section 5.3.1. All potential crowd-shippers are sorted in ascending order of their start times. The generation of parcels and crowd-shippers is performed using a pseudo-random number generator, such that simulations using various policies can be directly compared.

Upon the arrival of a crowd-shipper, a parcel or a set of parcels is assigned to this crowd-shipper based on one of the strategies described in Section 5.3.2. After the assignment, the assigned parcel(s) is/are reserved for the crowd-shipper for pickup and the crowd-shipper departs from his origin to the origin of the parcel(s). A new pickup event is scheduled, taking into account the travel time between the origin of the crowd-shipper and the origin of the parcel. As soon as a parcel is assigned to a crowd-shipper, it is no longer available to be assigned to other crowd-shippers, even

when it is not yet picked up.

For every pickup event, a delivery event is scheduled taking into account the travel time between the origin and destination of a parcel. As only parcels with identical origins and destinations are assigned to the same crowd-shipper, pickup events that correspond to different parcels but the same crowd-shipper occur simultaneously. The same is true for delivery events. For every delivery event, the number of served parcels and total costs are updated and the detour made by the crowd-shipper is stored. The simulation ends when all parcels have been delivered or when all crowd-shippers have either completed a delivery or have failed to be assigned to a parcel. We emphasize that the simulator allows us to consider time synchronization constraints. Crowd-shippers are considered in order of their arrival time and the system is constantly updated such that only the parcels that are available at the arrival time of the crowd-shipper are considered for pickup.

5.3.1 Stage 2: parcel-depot assignment

After the depots are determined in stage 1, parcels have to be assigned to depots on a day-to-day basis. At this stage, parcels are assumed to be known exactly (all orders of parcels have been collected), but crowd-shippers can announce their availability last minute and are therefore unknown. By simply assigning parcels to the closest depot in terms of distance, the importance of the flow of potential crowd-shippers is neglected. For the sake of comparison, we consider a distance-based metric that assigns all the parcels of a region to the opened depot that is closest to that region. Consider the set of opened depots D and consider μ_r parcels with destinations in region R . We recall that the travel time between depot d and region r is defined as t_{dr} . We define a_{rd} the number of parcels with final destination $r \in R$ that are stored at depot $d \in D$ as follows:

$$a_{rd} = \mu_r \mathbb{1}_{\left[t_{dr} = \min_{d' \in D} t_{d'r}\right]}. \quad (5.27)$$

Although a depot can be close in terms of distance, if very few crowd-shippers can feasibly deliver a parcel from that depot to the final destination, such a parcel-depot assignment can perform poorly. Therefore, we develop an assignment strategy based on the CA-estimates obtained using the algorithm as described in Section 5.2.2. We solve the single-depot-approximation for every depot $d \in D$ to obtain the expected number of parcels delivered to region $r \in R$, $v_r(\{d\})$. To obtain the single-depot approximations, we ignore the capacity of crowd-shippers to be slightly more conservative, which has shown to perform better. We then assign the

parcels proportional to the expected number of parcels delivered from a specific depot location. In this case, we define a_{rd} as follows:

$$a_{rd} = \mu_r \frac{v_r(\{d\})^\gamma}{\sum_{d \in D} v_r(\{d\})^\gamma}. \quad (5.28)$$

The tuning parameter γ can be used to give extra weight to larger depots (depots with high expected deliveries) and less weight to smaller depots (depots with lower expected deliveries). We emphasize the similarity of this assignment and parcel-depot assignment in the CA estimation in Equation (5.24).

5.3.2 Stage 3: parcel-crowd-shipper matching

In the third stage of our problem, when the depots are known and parcels are distributed over these depots, parcels have to be assigned to crowd-shippers. Generally, parcels are matched to crowd-shippers dynamically, upon arrival of the crowd-shippers. A parcel can only be matched to a crowd-shipper if they can pick up and deliver the parcel within their maximum detour, this is referred to as a *feasible match*. If a crowd-shipper can be feasibly matched to multiple parcels, the operator has to decide which parcel to assign to the crowd-shipper to maximize the total number of delivered parcels over the entire planning horizon. We consider the following three alternative matching approaches. The static matching approach relies on solving an integer linear programming problem, whereas the minimal-detour and CA-based matching use a simple decision rule. Furthermore, the static matching approach uses information about future crowd-shippers, whereas the other two approaches, more realistically, only consider one crowd-shipper at a time.

Static matching

The static matching approach assumes complete knowledge of all future crowd-shippers that will arrive. The static matching can be obtained by solving an ILP problem, that can be taken from the third stage of the stochastic programming formulation (5.8) - (5.13). Here, the depots are fixed and the parcel-depot assignment has been made. Therefore, the variable y_{pd} can be fixed to 0 or 1 according to the previously made assignments. This then simplifies the right-hand side of Constraints (5.10), while the rest of the formulation remains unchanged.

As the static matching is made without uncertainty about the future crowd-shippers, this forms a lower bound to the other matching approaches. In reality, this could be very unrealistic if crowd-shippers make themselves known only shortly before departing. Despite this, it forms a useful benchmark to compare the dynamic assignment strategies as well as the continuum approximation.

Minimal-detour matching

The minimal detour matching matches the crowd-shipper to the parcel for which the crowd-shipper minimizes the detour. This matching strategy, therefore, does not use any knowledge of future crowd-shippers. This strategy minimizes crowd-shipper inconvenience but is likely to be suboptimal as it does not use any information on future crowd-shippers. We consider that the set P only contains parcels that are not yet assigned to other crowd-shippers, and we consider $c \in C$ the current crowd-shipper. We let $or(\cdot)$ and $dest(\cdot)$ be the origin and destination location, respectively, for a crowd-shipper or parcel. Then the minimal detour matching chooses the parcel that minimizes the detour a crowd-shipper makes to pick up and deliver a parcel. This is computed as follows:

$$p_{min} = \arg \min_{p \in P} [t_{or(c)or(p)} + t_{or(p)dest(p)} + t_{dest(p)dest(c)} - t_{or(c)dest(c)}]. \quad (5.29)$$

If multiple parcels with the same origin and destination have the lowest detour, up to q_c parcels are assigned to the current crowd-shipper c .

CA-based matching

The minimal-detour matching only uses information on the crowd-shipper that is currently available. In this way, part of the information about potential crowd-shippers arriving in the future remains unused, although this can be useful. Generally, historic data is available that provides insights into the expected number of crowd-shippers. In practice, expectations of demand based on historic data are used to make strategic decisions. Actual demand is usually known at least one day in advance and is therefore used for operational decisions. The CA-based matching approach exploits the information on the approximated number of delivered parcels to improve the matching quality. An arriving crowd-shipper is assigned the parcel with the destination region that has the lowest expected number of delivered parcels relative to the total demand in that region ($\frac{v_r}{\mu_r}$). The intuition behind this assignment strategy is that we favor parcels that are less likely to be delivered in the future. By doing so, we increase the total expected number of parcels delivered over the entire planning horizon. If the chosen destination region has multiple parcels available, up to q_c parcels are assigned to the current crowd-shipper c . To obtain the estimates v_r , we disregard the capacity of crowd-shippers to be slightly more conservative. This shows an improvement in the performance of the CA-based matching. Compared to static matching, CA-based matching only uses a simple metric to determine the assignment, rather than solving an ILP problem. Therefore, the match can be determined extremely fast, similar to minimal-detour matching, which forms a major advantage for our real-time application. Here, we do not take into account fluctuations in the arrival rates of crowd-shippers during the day. This is marked as an interesting direction for future research.

5.4 Results

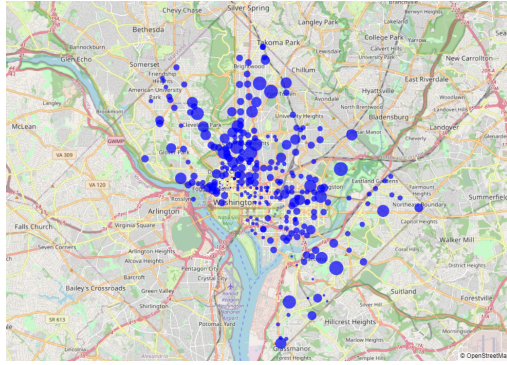
In this section, we evaluate the performance of the developed CA approach to find the optimal depots, as well as the potential of depot-based crowd-shipping. Our results are obtained through a case study based on the city of Washington DC, of which the details are described in Section 5.4.1. In Section 5.4.2 we evaluate the accuracy of our continuum approximation by comparing it to two exact benchmarks. In Section 5.4.3 we further evaluate the performance of our CA approach by comparing our solution to the solution obtained by solving the discrete formulation using a CPLEX solver. In Section 5.4.4 we evaluate the results on the network and in Section 5.4.5 we perform a sensitivity analysis on the optimal number of depots. Finally, in Section 5.4.6 we evaluate the effect of incorporating historic information about crowd-shippers and parcels in the three levels of decision-making on the results.

5.4.1 Case study

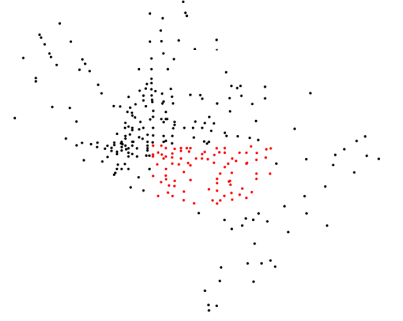
We use the city of Washington DC and the surrounding metropolitan area as a case study. The city of Washington DC has around 700,000 inhabitants and the entire agglomeration has around 7 million inhabitants. Washington DC hosts one of the biggest Bike-sharing platforms in the USA: Capital Bikeshare. Capital Bikeshare has over 500 stations and 4500 bikes (Capital Bikeshare, 2020). Such a bike-sharing platform with a large number of users forms a good base for a crowd-shipping service. We consider the locations of the bike-sharing stations as potential depot locations and use them as approximations for regions making up the entire service area.

Based on the station names, approximate coordinates of the locations are extracted from Google Maps, 2020. An approximation of the surrounding population has been made using Census Reporter, 2021 data, which in turn has been used as a proxy for demand for small parcels. We note that the actual coordinates of the stations may slightly deviate from the approximated coordinates due to misinterpreted station names. Nevertheless, the obtained network is used as a representation of an actual network. A bike-sharing station is used as a potential demand region with the expected number of parcels proportional to the population around that station. Historic system data from the Capital Bikeshare, 2020 database has been used to identify origin-destination pairs for the crowd-shippers. The stations that are used as regions are displayed in Figure 5.3a. This figure displays a bubble chart of those regions, where the size of the bubble is determined relative to the population around the corresponding station. Whereas the most used stations in terms of origins and destination of crowd-shippers are around Union Station, the Mall, and the center of Washington DC, demand is higher in the suburbs. This shows the large asymmetry in crowd-shippers' origin and destination locations on the one hand and parcel

destinations on the other that is usually present in crowd-shipping systems in real urban areas, making this case study especially realistic and interesting. We consider 330 stations in the center of Washington DC and the nearest suburbs, as displayed in Figure 5.3a. In Sections 5.4.2 and 5.4.3 we consider a smaller network of only 90 nodes in the center, as the benchmark approaches used in those sections are computationally intractable for the larger instance. The network is displayed in Figure 5.3b with the smaller subnetwork in red. For the sake of computation time, we only use crowd-shippers that have their origin and destination within the selected area. In our experiments, the number of parcels has been fixed at the expected value such that crowd-shippers are the only uncertain variable in the problem.



(a) Bubble chart of bike-sharing stations, where the size of the bubble is determined by the population in the area.



(b) Abstract network used as input to the optimization and simulation, with smaller subnetwork in red

Figure 5.3: Network of Washington DC used for the case-study

We use the following parameter values to get a realistic interpretation of the results. We assume that the cost of opening and operating a depot ϕ^{depot} is equal to 1000\$ per day. This is based on the average rental price in Washington DC to place the storage lockers and the maintenance cost involved in operating the depots. The cost of regular delivery is set to be 15\$, similar to Le, Stathopoulos, Van Woensel, and Ukkusuri, 2019. A part of these costs is still made to serve the depots, therefore, we consider a slightly lower value for ϕ_p^{reg} equal to 10\$. Our baseline scenario assumes a homogeneous set of crowd-shippers that are willing to make a detour (τ) of at most 500 meters and wish to receive (ϕ_p^{cs}) 5\$ to make a delivery plus 1\$ for every kilometer traveled with a parcel. The daily demand for parcels is assumed to be roughly 20,000 per day, proportional to the population in a region. The number of potential crowd-shippers is set to approximately 25,000. Of these crowd-shippers, 60% can carry only 1 parcel, 30% can carry 2 and 10% can carry up to 3 parcels. The demand scenario is fixed to the expected value, whereas 10 different supply scenarios are generated based on a Poisson process.

The LNS parameters are tuned in order to find a good objective value within

a reasonable time. The algorithm uses 5 initial solutions (η) and 500 iterations (κ). The repair and destroy operators take parameters α equal to 4.5 and β equal to 8. These values were chosen based on a grid search over a large range of parameter values. The parameters were chosen such that they maximize the number of multi-start initial solutions for which the best objective was found and at the same time minimize the number of iterations needed to find this objective. The tuning parameter γ , used for the parcel-depot assignment, is chosen equal to 1. CPLEX version 12.6.3.0 is used in Java to solve all MILPs.

5.4.2 Comparison of continuum approximation to static and dynamic assignment strategies

To evaluate the quality of the continuum approximation, we compare the estimated total costs to two benchmarks. The first benchmark assumes full knowledge of crowd-shippers *before* parcels are distributed to the depots. This means that we use the formulation of the second and third-stage as given by Equations (5.4) - (5.13), but for a single realization of parcels and crowd-shippers which is known with certainty. This eliminates the expected value in Equations (5.4) and (5.8) and allows us to solve the problem to optimality using a standard Mixed Integer Programming (MIP) solver, such as CPLEX. The objective is to minimize the operational costs which are given in (5.8). The second benchmark is a realistic dynamic assignment procedure. We use the CA-based parcel-depot assignment and the CA-based matching of parcels and crowd-shippers, which outperforms other assignment strategies (see Section 5.4.6).

For the sake of computation time, we consider a subset of the full network of only 90 regions and we set the capacity of every crowd-shipper to 1. In this way, we eliminate Constraints (5.12), which significantly reduces the computation time of the static assignment method. A large sensitivity analysis is performed for the expected number of potential crowd-shippers ($\hat{\lambda}$), the maximum detour (τ), and the number of depots ($|D|$). We use daily demand for parcels $\hat{\mu}$ equal to approximately 4000 and varying $\frac{\hat{\lambda}}{\hat{\mu}}$ from 0.5 to 2. We vary τ from 250 meters to 1000 meters and we consider 1, 3, 5, and 7 depots, which is appropriate given the small network. Given the lower number of parcels, we also change the depot cost ϕ^{depot} to 100\$. We compare the objective obtained through the CA approach as well as the LP approximation. The results are visualized in Figure 5.4 to 5.7, that display the percentage difference between the predicted objective and the actual (simulated) objective. We compare the LP-approximation and the algorithmic approximation to a static and dynamic assignment strategy. A set of 10 demand and supply instances is used for all simulations.

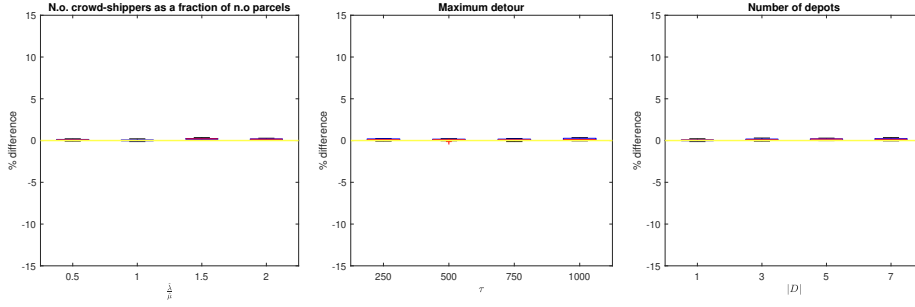


Figure 5.4: Percentage difference between the predicted objective by the LP approximation and the actual objective simulated by the static assignment strategy

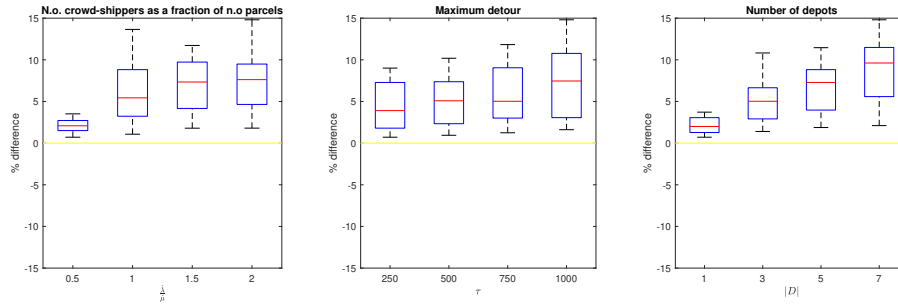


Figure 5.5: Percentage difference between the predicted objective by the LP approximation and the actual objective simulated by the dynamic assignment strategy

It is clear that the LP approximation performs well compared to the static benchmark, with near 0 differences between the approximation and the actual objective. The reason for this is that both the benchmark and the approximation disregard uncertainty and are therefore able to reach a lower bound on the total costs. However, compared to the more realistic dynamic benchmark the LP approximation underestimates the actual total costs by on average 5% and can go up to 13%. Clearly, with increasing uncertainty, the performance of the LP approximation will decrease. Thereby, the performance also deteriorates with the number of hubs and slightly deteriorates with the number of crowd-shippers and the maximum detour. We emphasize that the objective generally decreases when $\frac{\hat{\lambda}}{\hat{\mu}}$, τ , and $|D|$ increase and therefore the percentual difference between the actual and approximated objective is amplified.

The CA approximation accounts for uncertainty between the two lower levels of decision-making whereas the static benchmark ignores this uncertainty. Therefore, the CA approximation overestimates the objective of the static benchmark and performance deteriorates especially when the number of hubs increase as the static benchmark can optimally make the parcel-hub assignment, whereas the CA approximation cannot. Compared to the more realistic dynamic benchmark, the approximated objective differs on average 2% from the actual objective with ranges between 0 and 5%. A comparison of Figures 5.5 and 5.7, shows that

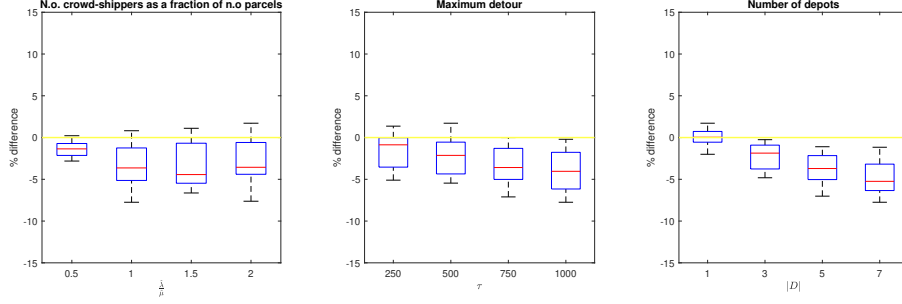


Figure 5.6: Percentage difference between the predicted objective by the CA approximation and the actual objective simulated by the static assignment strategy

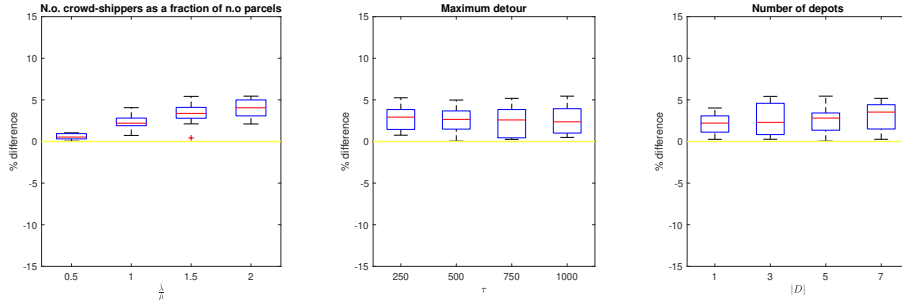


Figure 5.7: Percentage difference between the predicted objective by the CA approximation and the actual objective simulated by the dynamic assignment strategy

incorporating the uncertainty between the two lower-level decisions significantly improves the performance of the approximation. With respect to computational time, the CA approximation is on average 100 times faster on the small network. Thereby, the number of variables and constraints will increase for the full-size network, making it even more computationally demanding for the problem to be constructed and solved by CPLEX.

5.4.3 Comparison of CA approach to discrete formulation

In this section, we further evaluate the performance of our algorithm by comparing the CA-based solution algorithm proposed in this work to solving the discrete formulation in Section 5.2.1 using CPLEX. For the discrete formulation to be solvable within a reasonable amount of time, we consider a subset of the network with 90 nodes, limit the number of scenarios in ξ_P and ξ_C to 1 and the capacity of the crowd-shippers to 1. In this way, the expected values in the objectives in Equations (5.1) and (5.4) are eliminated and due to the full knowledge of supply and demand, the variables y_{pd} can be omitted as well. Thereby, the capacity of 1 allows us to eliminate Constraints (5.12). We emphasize that this significantly simplifies the discrete formulation, as for larger networks with more realistic settings (such as those considered in Sections 5.4.4 - 5.4.6), CPLEX fails to find an optimal or

feasible solution or even fails to construct the model due to the size of the problem. We also note the discrete formulation requires integer inputs. In the 90×90 network, this may lead to a relatively high difference between the two estimates which could potentially influence the solution.

The results are displayed in Figure 5.8. The left-hand panel displays the objective of the CA method relative to the objective of the discrete formulation. The objective values were computed as an average of 10 simulations using a CA-based dynamic assignment strategy for the parcel-depot and parcel-crowd-shipper assignments. Rather than directly comparing the objective values, we subtract a baseline of €5 for every parcel to properly quantify the percentual difference in the objective value. The right-hand panel displays the computation times of both methods for various settings, where we note the log scale of the y-axis. A time limit of 1 hour has been implemented and CPLEX solves the problem up to a 5% optimality gap (without this, the solver may continue looking for negligible improvements, creating a biased comparison of CPU times). For this experiment, demand for parcels $\hat{\mu}$ and the number of crowd-shippers $\hat{\lambda}$ vary between 1000 and 2000 and the maximum detour is either 250 meters or 500 meters. Compared to the previous experiment, smaller values have been chosen such that the discrete formulation is solvable within a reasonable amount of time.

For the comparison of the objective values, we emphasize that the hub locations for both the discrete formulation and the CA-based algorithm were determined using only a single scenario, which can lead to suboptimality in the dynamic simulation setting. We observe that the CA-based method generally obtains better solutions, which can be partially explained by the chosen 5% optimality gap for the discrete formulation. Clearly, the results obtained by the CA-based algorithm are more robust and find good solutions even when using only a single scenario. The discrete formulation, on the other hand, does not find robust solutions and requires a higher number of scenarios to adapt the depot locations. Thereby, the current simplification ignores the uncertainty which is captured by the CA-based algorithm, which may therefore also lead to suboptimality. We also observe that for settings for which the discrete formulation cannot be solved within the 1-hour limit, the improvement of the CA-based algorithm over the discrete formulation is significantly higher.

Computation times are significantly higher for solving the discrete formulation than for the CA-based method. Computation time for both methods increases with the number of hubs. However, the CA-based approach is only marginally influenced by the number of crowd-shippers, parcels, and maximum detour. Averaged over all tested settings, the CA-based algorithm is almost 150 times faster than solving the discrete formulation. For the high-demand case with 2000 parcels, this even

goes over 400 times faster, despite computation time being limited to one hour. Thereby, we emphasize again that for larger networks with more realistic settings such as multiple scenarios, the discrete formulation cannot be used at all whereas the CA-based method only requires a couple of minutes to find high-quality solutions.

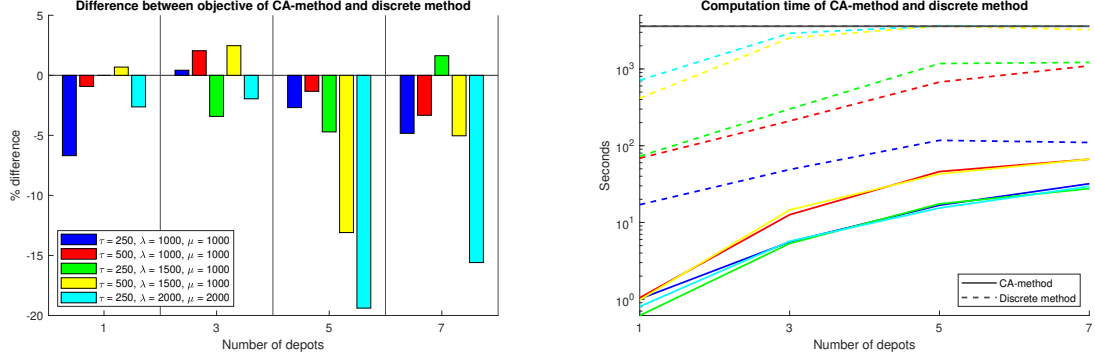


Figure 5.8: Comparison of the CA-based method and solving the discrete formulation

Another alternative method is to use a simulation-optimization approach in combination with the described large neighborhoods search algorithm. Here, a simulation is used to evaluate the objective rather than the CA-based approximation. The performance of this method is highly dependent on the details of the simulator and the efficiency of the implementation. Thereby, to obtain a good estimate, the simulation has to be repeated many times to obtain an average. For large-scale systems, such as the one considered in the remainder of this chapter, evaluating the objective by simulation is computationally too time-consuming to obtain results within a reasonable amount of time or will also run into memory issues like the discrete formulation. Note that even the initialization of the LNS algorithm requires strong computational effort for large systems.

5.4.4 Results on the network

To obtain managerial insights regarding the exact location of depots, we evaluate the depot locations in the network. We emphasize that from this section onwards we use the 330 node network with the baseline parameters outlined in Section 5.4.1. The bubble chart in Figure 5.9 displays the considered network where a blue bubble represents a regular demand region and a red bubble represents a demand region that was chosen to have a depot. The size of the bubble represents the fraction of demand that was served by crowd-shippers. That is, a full bubble implies that all parcels in the region are expected to be delivered by crowd-shippers, according to the continuum approximation, whereas a smaller bubble implies that only a fraction of the parcels is expected to be delivered by crowd-shippers. Figure 5.9a displays the approximation for three depots and Figure 5.9b displays the approximation for five depots.

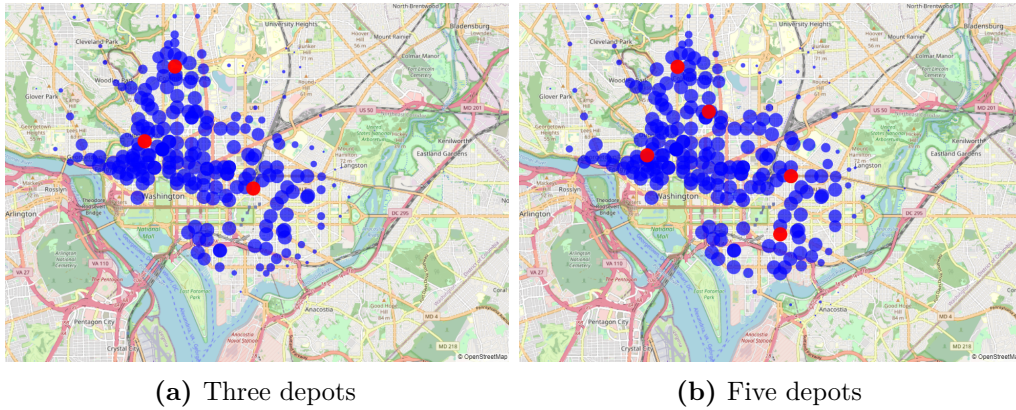


Figure 5.9: Bubble chart of parcels served by crowd-shippers in the network

We observe that depots are located in the city center of Washington DC where the number of potential crowd-shippers is the highest. As we are using bike-sharing users to approximate the commuting patterns of crowd-shippers, these are mainly in the city center. We notice that most depots are in the northwest of the city center, where demand is the highest as depicted in Figure 6.4. In addition to this, depots are spread sufficiently to attract more crowd-shippers and to have a broader service area. Many of the chosen depots are either at popular origins of potential crowd-shippers, such as a train station, or at popular intersections where many crowd-shippers pass by either directly or within their maximum detour.

The strong inter-dependency between the depots is shown in Figure 5.10, which displays from which depots the parcels delivered to each region originate. Figure 5.10a shows the spider chart for three depots and Figure 5.10b shows the spider chart for five depots. The majority of the destination regions are served by multiple depots. Only the regions in the outskirts of the network are served by a single depot. To quantify the results in this figure, we calculate how many regions are served by more than one depot. Specifically, we count the number of regions for which at most 90% of the delivered parcels originate from one depot. For three depots, 37% of the regions where at least one parcel is delivered are served by more than 1 depot. For five depots, this is as high as 70%. Typically we observe that the higher the number of depots, the lower the number of regions served by a single depot and thus the higher the inter-dependency. Intuitively, this corresponds to the fact that when more depots are constructed, the distance between depots is lower, and therefore their similarity (see Section 5.2.3) increases.

5.4.5 Optimal number of depots

In this section, we evaluate the optimal number of depots for τ varying between 500 and 1000 meters. Figure 5.11 displays the costs for a varying number of depots, as

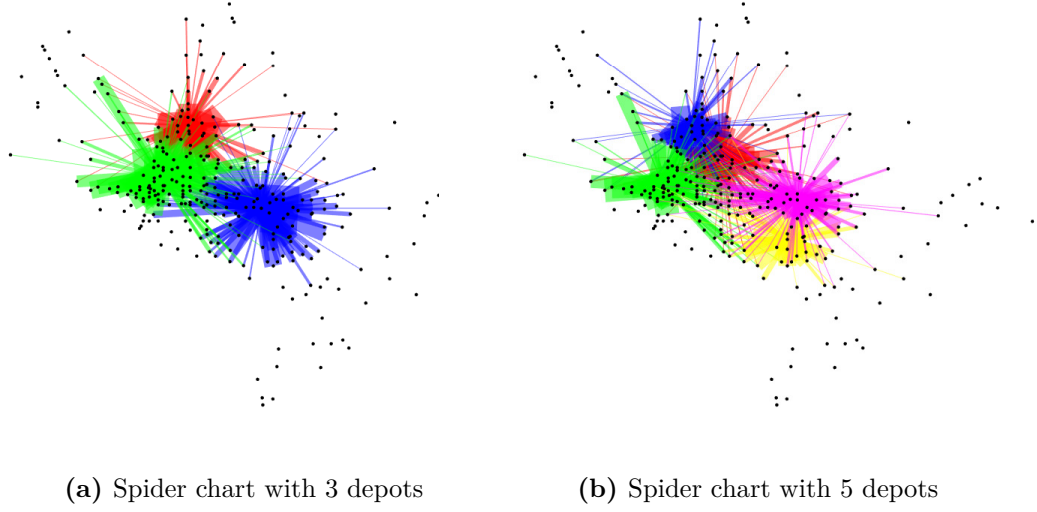


Figure 5.10: Spider chart linking the origins and destinations of delivered parcels

approximated by the CA approach, for a set of depots that is obtained through the CA-based heuristic. The total costs are split into the three components that make up the objective function in Equation (5.25) and consist of costs of maintaining depots, crowd-shipper compensations, and penalties for undelivered parcels.

Increasing the value of τ has two opposing effects on the total costs. On the one hand, potential depot locations that previously were not able to serve sufficient demand to be profitable may now be able to serve more demand because the maximum detour is higher, thereby increasing the optimal number of depot locations. On the other hand, a depot may be able to reach more demand regions due to the increase of τ , making other depots obsolete, thereby reducing the optimal number of depots. A similar effect is true for an increase or decrease in the total expected number of crowd-shippers, which may either lead to an increase or decrease in the optimal number of depots. We emphasize that this is specific to the chosen scenario and costs.

The marginal percentage of demand served due to the addition of one more depot is diminishing. The first depots are the most profitable and can serve a relatively large portion of the demand. As we showed in Section 5.4.4, these depots are built at central locations where the number of potential crowd-shippers is high. Afterward, additional depots can be opened at less busy locations to further increase served demand, but the effect is substantially lower.

5.4.6 Comparison to non-predictive strategies

As described in Section 5.2, the proposed solution approaches to all three stages contain a predictive component to incorporate the influence of parcel and crowd-shipper

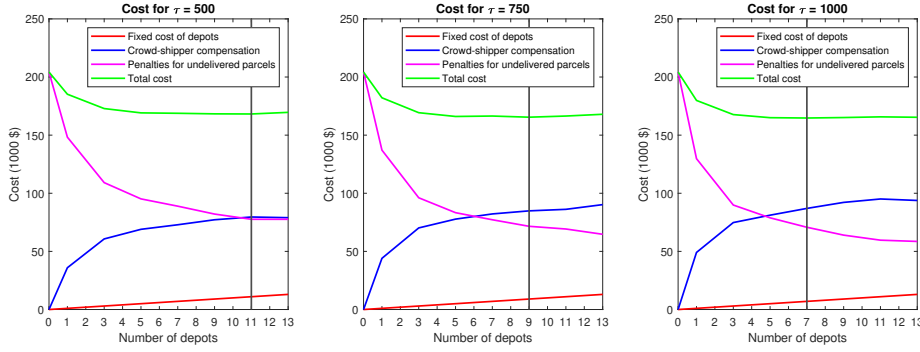


Figure 5.11: Decomposition of costs for varying number of depots and maximum detour τ

patterns and their interaction in decision-making. We use historic information on parcels and crowd-shippers to enhance the decision-making on all three layers. In this section, we compare the predictive CA-based strategies to those that do not use such a predictive component and only base the decisions on geographical distances. The first stage decisions are compared to an FLP minimizing the total distance of every region to the closest depot. The second stage parcel-depot assignment is compared to an assignment strategy where each parcel is assigned to the closest opened depot. The third stage matching is compared to a minimal-detour strategy for crowd-shippers. These non-predictive approaches ignore the connection between the three stages and solve every stage separately, basing their decisions solely on distance rather than estimates of the pattern of parcels, crowd-shippers, and the interaction of the two. The service levels are displayed in Table 5.2. For every scenario, the percentage of parcels served by crowd-shippers is an average of 10 simulation runs. We consider a base case with 5 depots, roughly 20,000 parcel requests, and 25,000 crowd-shippers (with varying capacities that are 1.5 on average). The maximum detour for crowd-shippers is 500 meters.

Using predictive methods that incorporate expected patterns and interactions of parcels and crowd-shippers significantly improves the performance of the depot-based system. Especially the use of our algorithm to determine the optimal depot locations compared to a simple distance-based algorithm that chooses depots at central locations increases the service level by more than 10%. Additionally, using predictive strategies to make second and third-stage decisions can improve the objective by up to 5%. Overall, the results indicate that for this specific set of configurations (i.e., number of depots, maximum detour, etc.) using predictive components in all three levels of decision-making can improve the service level by 15%. We emphasize that results may vary for other configurations. For example, if we increase the number of depots to 10, the fully non-predictive service level

is 44%, whereas the fully predictive service level is 56.1%. Typically, we observe that higher service levels are more difficult to improve.

Table 5.2: Comparison of predictive CA-based strategies to non-predictive strategies

First-stage	Second-stage	Third-stage	
		Non-predictive	Predictive
Non-predictive	Non-predictive	35.4	38.4
	Predictive	36.5	38.7
Predictive	Non-predictive	46.1	48.1
	Predictive	48.1	50.9

The reported values are the number of parcels delivered by crowd-shippers as a percentage of the total demand for parcels.

For the same settings as before, we compare the service level and profit of the predictive and non-predictive approach to randomly generated sets of 5 depots. This provides additional insights into the performance of our methods as well as into the influence of the exact depot locations on service level and profit. We randomly generate 100 sets and evaluate them all on the same 10 instances. This then yields 1000 observations of profit and service level that are displayed in Figure 5.12. The spread of the histograms shows that the exact locations of depots significantly influence the service level and profit of the crowd-shipping system. Clearly, the non-predictive FLP method is outperformed by some randomly generated sets. The predictive CA-based approach, on the other hand, outperforms all randomly generated sets.

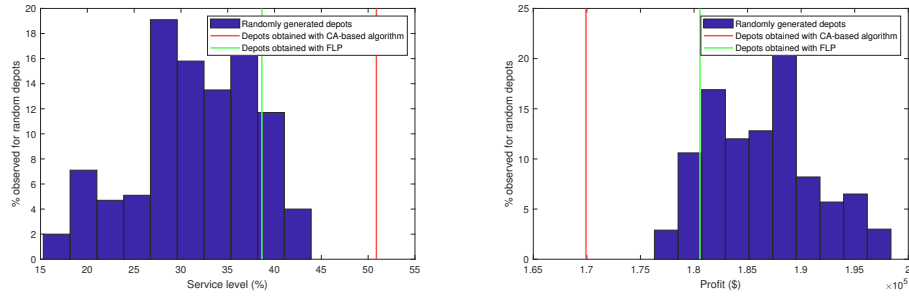


Figure 5.12: Histogram of service level and profit for random sets of 5 depots

5.5 Summary

Crowd-shipping is a promising alternative to traditional last-mile delivery methods that can help to reduce congestion, and pollution and improve the overall performance of the delivery system. One of the main drivers is the availability of sufficient

suppliers. In this chapter, we proposed a depot-based crowd-shipping system where crowd-shippers pick up parcels from depot locations and deliver them to the final destination. By constructing depots at strategic locations, more crowd-shippers can be attracted to serve more demand.

We approximate the lower-level decisions of assigning parcels to depots and later to crowd-shippers using a Continuum Approximation (CA) approach. The approximation shows to provide an accurate estimation of the actual objective using dynamic assignment strategies, on average being within 2% of the optimal dynamic assignment problem, with deviations being at most 5% on the tested instances. We developed a heuristic approach to efficiently determine the optimal depot locations. The heuristic uses CA to approximate the lower-level operational decisions and a large neighborhood search heuristic to make the strategic upper-level decisions (i.e., find the best set of depots). Using quality and similarity metrics, the search space is explored efficiently and a solution is found within a reasonable amount of time. We compare the performance of our algorithm to solving a discrete formulation. This comparison indicates that, on a set of smaller instances of 90 regions, our algorithm is on average almost 200 times faster on the tested, going up to 1000 times for the largest tested instance. In terms of solution quality, our algorithm generally finds slightly better solutions for instances where the CPLEX solver converges with the one-hour time limit, but significantly better solutions for instances where the solver does not terminate. For realistic instances of 330 regions, our algorithm can find good depot locations within a reasonable amount of time whereas the MILP formulation cannot even be constructed, let alone find feasible solutions. A comparison of our suggested approach where all three levels of decision-making incorporate historic information of crowd-shippers and parcels outperforms distance-based methods by 15%.

6

A column and row generation approach to the crowd-shipping problem with transfers

This chapter is based on the following article:

- P. Stokkink, J.-F. Cordeau, and N. Geroliminis (2023). “A column and row generation approach to the crowd-shipping problem with transfers”. Available in: HEC Working Papers (Currently under review in a scientific journal)

6.1 Introduction

One of the main operational challenges in crowd-shipping is matching crowd-shippers to parcels that need to be delivered. The quality of such a match is influenced by the detour that the crowd-shipper needs to make to pick up and deliver the parcel, as well as potential time windows that need to be satisfied. Especially when the number of parcels and the number of crowd-shippers is high, finding the optimal matching is challenging, yet important to optimize the service level. Another major challenge that can complicate matching problems is stochasticity in demand (i.e., uncertainty in destination, quantity, and time window) as well as supply (the full itinerary of crowd-shippers is uncertain until they communicate it).

When the origins and destinations of parcels are further apart than those of potential crowd-shippers, finding matches that can directly take the parcels from their origin to their destination can be difficult. Especially in bike-based or pedestrian-based crowd-shipping, the two forms that are considered among the least polluting and

with the highest potential (i.e. lower value of time), crowd-shipper trips are usually short whereas distances across the city can be long. In this chapter, we consider multi-stage deliveries where parcels can be transported from their origin to their destination in multiple stages and with multiple crowd-shippers.

In this chapter, we propose a general framework that allows the incorporation of both time-synchronized transfers as well as transfers with intermediate storage at transfer points. To the best of our knowledge, this is the first model that can capture both types of transfers simultaneously. In addition to this, we consider the original itinerary of crowd-shippers including their departure times, but we consider some flexibility in their routing decisions. This makes our crowd-shipping system more realistic than those generally considered in the literature and makes crowd-shipping accessible to daily commuters. On top of this, we consider a detailed compensation scheme for crowd-shippers, which includes rewards for stops, detours, and the inconvenience of carrying a parcel for a longer distance. Furthermore, we consider heterogeneous crowd-shippers and parcels. We propose a column-generation approach to solve our problem. This method is highly scalable and allows solving larger instances than those previously considered in the literature for similar problems. Our results are evaluated on a realistic large-scale case study in the city of Washington DC.

This chapter is organized as follows. A formal problem description and formulation are given in Section 6.2. The methods used to solve this problem are given in Section 6.3. Simulation results are provided in Section 6.4 and the chapter is concluded in Section 6.5.

6.2 Problem description and formulation

In Section 6.2.1 we introduce the main concepts and notation used in the chapter before providing a mathematical formulation of the problem in Section 6.2.2.

6.2.1 Concepts and notation

We consider a set P of parcels that make up the considered demand requests. Every parcel $p \in P$ has an origin o_p , a destination d_p and a delivery time window $[e_p, l_p]$, where e_p is the earliest delivery time and l_p is the latest. Every delivered parcel p generates revenue, which can vary between parcels, and is denoted by ρ_p . The set C contains all (potential) crowd-shippers. Every crowd-shipper $c \in C$ has an origin o_c , a destination d_c , and a trip starting time at t_c . Crowd-shippers may be willing to deviate from their path with a maximal detour of τ_c . The detour can be measured either in units of distance or units of time.

Based on their maximum detour, a crowd-shipper c is able to execute a set of delivery segments S_c . A crowd-shipper always executes at most one segment. Figure 6.1 illustrates a network with a crowd-shipper traveling from A to D with its original path, marked in green, being $A \rightarrow B \rightarrow D$. The crowd-shipper can also travel through the blue path $A \rightarrow B \rightarrow C \rightarrow D$ within his maximum detour. Based on these two paths, the list of segments for this crowd-shipper is: $[AB, AC, AD, BC, BD, CD]$. Based on the crowd-shipper's start time, we can compute the time at which the crowd-shipper starts the segment, which is given by t_s . A segment also has an origin o_s and a destination d_s . A crowd-shipper $c \in C$ is rewarded w_{cs} for traversing a segment $s \in S_c$. This cost is made up of three components:

1. A fixed compensation α_c^1 for the inconvenience of pickup and delivery;
2. A variable compensation based on the detour crowd-shipper $c \in C$ makes to perform the delivery on segment $s \in S_c$, denoted by $\alpha_c^2 \cdot \text{det}_{cs}$;
3. A variable compensation based on the time/distance spent carrying the parcel, which is equal to the length of the segment and denoted by $\alpha_c^3 \cdot \text{len}_s$.



Figure 6.1: Illustration of a crowd-shipper travelling from A to D that can perform segments: AB, AC, AD, BC, BD, CD

A parcel can be transferred between crowd-shippers at a set H of transfer points or transfer hubs. After the parcel is dropped off at the transfer point by a crowd-shipper, the next crowd-shipper can pick up the parcel at least Δ^{\min} time units later (a safety margin) and at most Δ^{\max} time units later (to avoid the parcel staying at the hub for too long). We note that by choosing the set H of points to be arbitrarily large and Δ^{\max} arbitrarily small, this corresponds to direct transfers where parcels are handed directly from one crowd-shipper to another. Otherwise, parcel lockers need to be present at transfer hubs for crowd-shippers to temporarily store the parcels. Generally, this may differ across transfer points $h \in H$ and we allow Δ_h^{\min} and Δ_h^{\max} to vary.

The objective is to maximize the profit consisting of the revenue for delivered parcels minus the costs of paying crowd-shippers. For this, we determine the

optimal matching of parcels to crowd-shippers. Specifically, for the multi-stage delivery problem, we determine the exact path a parcel traverses from its origin to its destination. This path may be direct or through transfer points and by using multiple crowd-shippers. To this end, we define the concept of a parcel path.

Definition 1. *A parcel path is the trajectory a parcel traverses to get from its origin to its destination. A parcel path is made up of one or more segments that a parcel travels with a crowd-shipper. Between segments, a parcel is stored at a transfer point.*

In the next section, we give a formulation of the problem based on this concept of parcel paths. The approach we take to solve the problem is described in Section 6.3.

6.2.2 Mathematical formulation

We first give a full formulation of the problem described above. This is a path-based formulation that maximizes the revenue collected by parcel deliveries minus the costs of crowd-shipper compensation. The full set of parcel paths is denoted by K , where K_p is the set of parcel paths that correspond to parcel $p \in P$. Only feasible parcel paths (i.e., paths that are fully connected and time-synchronized) are included in the set K . The binary decision variable x_k is equal to 1 if parcel path $k \in K$ is selected and 0, otherwise. We define a_{ck} as a binary parameter that is equal to 1 if crowd-shipper $c \in C$ is involved in parcel path $k \in K$. For completeness, we also introduce binary parameter b_{csk} , which is equal to 1 if crowd-shipper $c \in C$ contributes to parcel path $k \in K$ by performing segment $s \in S_c$ and 0 otherwise. Although this parameter is only indirectly part of the formulation of the problem through the defined profit of a parcel, it is required for the solution approach. Clearly, following the definition of a segment, $a_{cs} = \sum_{k \in K} b_{csk}$.

The profit of a parcel path $k \in K_p$ is defined as π_k and is defined as follows:

$$\pi_k = \rho_p - \left[\sum_{c \in C} a_{ck} \alpha_c^1 + \sum_{c \in C} \sum_{s \in S_c} b_{csk} (\alpha_c^2 \det_{cs} + \alpha_c^3 \text{len}_s) \right]. \quad (6.1)$$

Here, the first term captures the revenue obtained by delivering the parcel p corresponding to the column $k \in K_p$. The second term is the fixed price paid to a crowd-shipper for making a delivery. This does not depend on the segment and therefore only uses parameter a_{ck} . The third term is a variable cost paid to a crowd-shipper which depends on the segment and is therefore based on b_{csk} . This term captures the cost per unit of detour and cost per unit travelled with a parcel.

The formulation of the problem is as follows:

$$\max \sum_{p \in P} \sum_{k \in K_p} \pi_k x_k \quad (6.2)$$

$$\sum_{k \in K_p} x_k \leq 1 \quad \forall p \in P \quad (6.3)$$

$$\sum_{k \in K} a_{ck} x_k \leq 1 \quad \forall c \in C \quad (6.4)$$

$$x_k \in \mathbb{B} \quad \forall k \in K. \quad (6.5)$$

The objective (6.2) is to maximize the total profit. By substituting Equation (6.1) we observe the dependency on parameters a_{cs} and b_{csk} . Constraints (6.3) ensure that every parcel is delivered at most once and therefore only one parcel path can be selected among those associated with that parcel. Constraints (6.4) ensure that a crowd-shipper is used at most once.

6.3 Methodology

We solve the problem using a column generation approach, where every column is a unique parcel path. Figure 6.2 schematically illustrates this approach. In the master problem, parcel paths from the current set of columns \bar{K} are selected to maximize revenue and minimize operational costs, by solving the LP relaxation of the Restricted Master Problem (RMP). In the pricing problem, new columns are generated that improve the current solution, based on the dual variables of the constraints of the last iteration of the LP. Finally, when no more columns with positive reduced cost are found the optimal solution to the LP is obtained. We then obtain an integer solution by solving the IP with the last set of obtained columns. We note that this does not guarantee the optimality of the IP solution. An exact method would require embedding the column generation in a branch-and-price framework. However, in our computational experiments, the optimality gap of the IP and LP objectives indicates that the obtained solutions are (near) optimal.

The master problem is described in Section 6.3.1 and the pricing problem is described in Section 6.3.2. The shortest path problem that is used to solve the pricing problem is described in Section 6.3.3.

6.3.1 Master problem

The formulation of the master problem closely resembles the formulation in Section 6.2.2. In the master problem, we select the best columns from the current set \bar{K} that maximize the obtained revenue from delivering parcels and minimizes the costs of crowd-shippers. In addition to the total set of columns, we define \bar{K}_p as

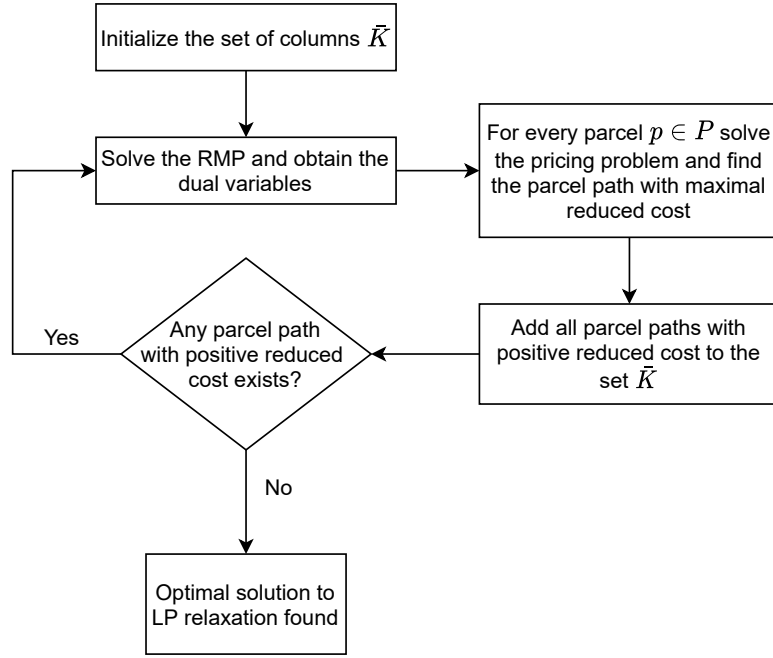


Figure 6.2: Column Generation Approach

the set of columns that correspond to parcel paths of parcel $p \in P$. It follows that $\bigcup_{p \in P} \bar{K}_p = \bar{K}$. The formulation of the master problem is as follows, with the dual variables of the constraints in parentheses.

$$\max \sum_{p \in P} \sum_{k \in \bar{K}_p} \pi_k x_k \quad (6.6)$$

$$\sum_{k \in \bar{K}_p} x_k \leq 1 \quad \forall p \in P \quad (v_p) \quad (6.7)$$

$$\sum_{k \in \bar{K}} a_{ck} x_k \leq 1 \quad \forall c \in C \quad (u_c) \quad (6.8)$$

$$x_k \in \mathbb{B} \quad \forall k \in \bar{K}. \quad (6.9)$$

6.3.2 Pricing problem

We extend the set of columns in the RMP by finding columns with positive reduced cost. The reduced cost for a new column $k \in K \setminus \bar{K}$ is defined as r_k and it can be computed as:

$$r_k = \pi_k - v_p - \sum_{c \in C} u_c a_{ck}. \quad (6.10)$$

We can rewrite this by substituting π_k from Equation (6.1), as follows:

$$r_k = \rho_p - \sum_{c \in C} a_{ck} \alpha_c^1 - \sum_{c \in C} \sum_{s \in S_c} b_{csk} (\alpha_c^2 \det_{cs} + \alpha_c^3 \text{len}_s) - v_p - \sum_{c \in C} u_c a_{ck}. \quad (6.11)$$

We can then rewrite this by grouping together similar terms:

$$r_k = (\rho_p - v_p) - \sum_{c \in C} a_{ck}(u_c + \alpha_c^1) - \sum_{c \in C} \sum_{s \in S_c} b_{csk}(\alpha_c^2 \det_{cs} + \alpha_c^3 \text{len}_s). \quad (6.12)$$

Recall that $a_{cs} = \sum_{s \in S_c} b_{csk}$ and that the total compensation paid to a crowd-shipper is denoted by w_{cs} . We can further simplify the definition of the reduced cost as follows:

$$r_k = (\rho_p - v_p) - \sum_{c \in C} \sum_{s \in S_c} b_{csk}(u_c + \alpha_c^1 + \alpha_c^2 \det_{cs} + \alpha_c^3 \text{len}_s) \quad (6.13)$$

$$r_k = (\rho_p - v_p) - \sum_{c \in C} \sum_{s \in S_c} b_{csk}(u_c + w_{cs}). \quad (6.14)$$

From Equation (6.14) it is clear that finding a column with positive reduced cost can be decomposed over the parcels. For every parcel, we search the parcel path with the highest reduced cost (if any column with positive reduced cost exists). This is done by finding the best crowd-shippers and segments to constitute a feasible path from origin to destination. This path has to satisfy basic flow constraints as well as timing restrictions to ensure that a parcel can only be picked up after it is delivered. As the problem is separated over parcels, the term $\rho_p - v_p$ is fixed. Finding a path with maximal reduced costs is then equivalent to minimizing $\sum_{c \in C} \sum_{s \in S_c} b_{csk}(u_c + w_{cs})$. This means that finding the positive reduced cost path is equivalent to solving the shortest path problem.

We consider a layered procedure for the pricing problem where direct, indirect paths with a single transfer, and indirect paths with multiple transfers are considered separately. This procedure is presented in Algorithm 3. First, direct paths are generated. Direct paths constitute a simple match of a crowd-shipper to a parcel. Here, the feasibility with respect to time windows and location needs to be verified and the costs are computed. Thereafter, indirect paths are generated. Although slightly more difficult due to time and location synchronization at the transfer, this can still be done by simply enumerating for every parcel all crowd-shippers that can pick up and all crowd-shippers that can deliver the parcel. Finally, we consider multi-stage deliveries by solving a shortest-path problem. As the number of transfers is not fixed, this is more complicated and discussed in detail in the remainder of this section.

This layered procedure has two main benefits. First, solving the pricing problem for direct delivery and indirect delivery with one transfer is computationally much faster. For a direct delivery, finding a column with a positive reduced cost only requires going over all feasible matches of crowd-shippers and parcels, which can be done in $\mathcal{O}(|P||C|)$. For an indirect delivery with one transfer, a similar approach is used where every crowd-shipper is considered twice (once for pickup and once for delivery),

which can be done in $\mathcal{O}(|P||C|^2)$. Therefore, the column generation algorithm can be warm-started first for direct deliveries and then also for indirect deliveries with one transfer, before considering the computationally more expensive multi-stage deliveries. The second benefit is that, by considering multi-stage deliveries separately, the shortest path problem and the corresponding graph can be fully adapted to this type of delivery and therefore improve the speed of the algorithm.

Algorithm 3: Layered procedure for pricing problem

```

1 for every parcel  $p \in P$  do
2   | Generate a direct path with positive reduced costs.
3 Compute  $\bar{r}$ ; the maximum reduced cost across all generated paths
4 if  $\bar{r} \leq 0$  then
5   | for every parcel  $p \in P$  and every pickup segment  $s \in N_p$  do
6   |   | Generate an indirect path with one transfer with positive
6   |   | reduced costs.
7   | Compute  $\bar{r}$ ; the maximum reduced cost across all generated paths
8   | if  $\bar{r} \leq 0$  then
9   |   | for every parcel  $p \in P$  and every pickup segment  $s \in N_p$  do
10  |   |   | Generate an indirect path with positive reduced costs, by
10  |   |   | solving the shortest path problem.
11 Add all generated paths with positive reduced costs to  $\bar{K}$ 

```

6.3.3 Shortest path algorithm - Graph construction

To solve the shortest path problem, a graph is constructed based on the movement of crowd-shippers through the road network. An example of such a graph is given in Figure 6.3 and will be described below. The shortest path problem is solved on a directed graph where nodes correspond to segments. Whenever a node is part of the shortest path, the variable b_{csk} is equal to 1 and it is equal to 0 otherwise. The cost of such a node is equal to $u_c + w_{cs}$, such that the length of the shortest path corresponds to the second term of the reduced cost in Equation (6.14). An arc between two nodes exists if the two segments are compatible, in the sense that one segment can be executed right after the other. An arc between two nodes n_1 and n_2 exists if all of the following conditions hold:

- The crowd-shipper of node n_1 is different from the crowd-shipper of node n_2 .
- The segment of node n_1 ends at the same transfer point where the segment of node n_2 starts.
- The segment of node n_1 finishes at least Δ^{min} time units before and at most Δ^{max} time units after the segment of node n_2 starts.

All existing arcs have a cost of zero, which means that the only cost components are on the nodes. For the multi-stage delivery problem we consider three types of nodes each corresponding to a type of segment: pickup nodes/segments, dropoff nodes/segments and transfer nodes/segments. We describe the properties of these nodes in detail below, with the set of nodes of each type in parentheses. A feasible parcel path starts with a pickup segment and ends with a dropoff segment, possibly with one or more transfer segments in between. A segment describes a part of the parcel path for which the parcel is travelling with the same crowd-shipper.

1. **Pickup nodes/segments (N_P):** A pickup segment represents the initial pickup of the parcel from the origin location and its delivery to a transfer point. A pickup node exists if the origin of the segment coincides with the origin of the parcel and the destination of the segment coincides with a transfer point. Thereby, it only exists if the start time of the segment is later than the earliest availability time of the parcel. A pickup node has no incoming arcs.
2. **Dropoff nodes/segments (N_D):** A dropoff segment represents the final delivery of the parcel from the last transfer point to the destination of the parcel. A delivery node exists if the destination of the segment coincides with the destination of the parcel and the origin of the segment coincides with a transfer point. Thereby, it only exists if the time window of the parcel is satisfied. A dropoff node has no outgoing arcs.
3. **Transfer nodes/segments (N_T):** A transfer segment represents the transfer of any parcel from one transfer point to another. There are no restrictions on location or time for the existence of a transfer node.

We emphasize that although pickup and dropoff nodes are parcel-specific, due to origins, destinations, and time windows, transfer nodes are not. Therefore, transfer nodes are only added once, whereas pickup and dropoff segments may be repeated for multiple parcels that are similar.

6.3.4 Modified Dijkstra's algorithm

To find the column to add to the master problem for every parcel, we aim to find the shortest path between any pickup segment and any dropoff segment. We do this by applying a modified version of Dijkstra's shortest path algorithm tailored to fit well the specifics of our problem. As Dijkstra's algorithm can find the shortest path from a source node to any node in the graph, we apply the shortest path problem $|N_P|$ times. The column with the highest reduced cost (if any column with positive reduced cost exists) is added to the master problem and this is repeated for every parcel.

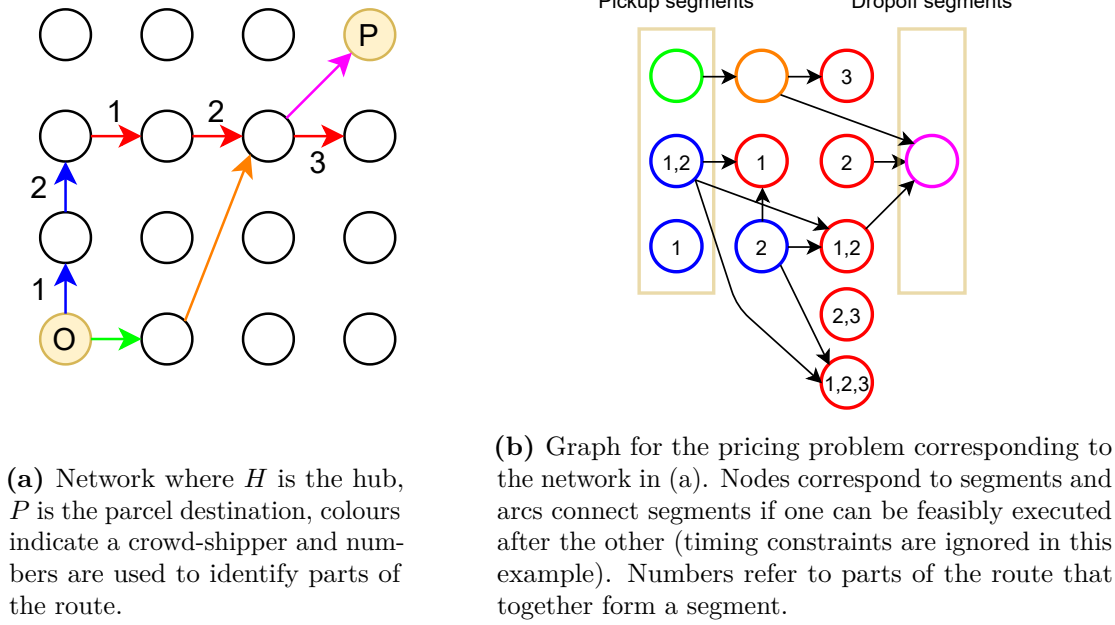


Figure 6.3: Conversion from network with 1 parcel and 5 crowd-shippers, each with a maximum detour of 0, to a graph for the pricing problem

Dijkstra's algorithm takes as an input a set of nodes and an adjacency matrix which defines the arcs between the nodes. We observe that the full graph does not change between iterations and can therefore be pre-computed once. Then, at each call to the pricing problem, only the costs on the nodes are updated according to the dual variables. The details on the algorithm are described in Algorithm 4. The algorithm enforces all constraints that hold between nodes through the adjacency matrix, as these constraints are transitive. The only exception to this is that two segments belonging to the same crowd-shipper may be included in the shortest path, as long as at least one other segment is in between. This constraint is not enforced as a hard constraint as this would make the problem resource-constrained. However, by construction of our problem, such paths are never feasible if $\Delta^{min} > 0$. As a crowd-shipper will leave directly after dropping off the parcel, whereas a parcel can only be transferred after Δ^{min} time units, the crowd-shipper will arrive at the next transfer point at least Δ^{min} time units before the parcel arrives with another crowd-shipper. As crowd-shippers never wait for a parcel to become available in our framework, these paths are implicitly eliminated. In Line 11, the algorithm is terminated because there exist no remaining *unvisited* nodes that can be visited through a feasible path from the source node. In Line 13, we skip the for-loop in Lines 14-16 whenever the current node is in N_D as this is by definition the last node on a path and therefore can not be on the shortest path to another node.

In addition to the modifications to Dijkstra's algorithm, more computational enhancements are made to improve the speed of the algorithm. We consider

Algorithm 4: Modified Dijkstra's Algorithm

```

1 Input: A set of nodes  $N = N_P \cup N_D \cup N_T$  with their costs  $c(n)$  for all
    $n \in N$ 
   For every node  $n \in N$ , a set of neighboring nodes  $A(n)$ 
   A source node  $s$ 
4 Output: A set of shortest paths from source node  $s \in N_P$  to all nodes in
    $N_D$ 
5 Mark all nodes as unvisited:  $visit(n) \rightarrow \mathbf{false}$ 
6 Set the shortest distance to each node at  $dist(n) \rightarrow \infty$ , except for the
   source node which is set to  $dist(s) \rightarrow c(s)$ 
7 while Not all nodes are visited do
8   Find node  $q^-$  as the unvisited node with minimal  $dist(q^-)$ 
9   Set  $visit(q^-) \rightarrow \mathbf{true}$ 
10  if  $dist(q^-) = \infty$  then
11    return shortest paths
12  if  $q^- \in N_D$  then
13    continue to next node
14  for  $q^+ \in A(q^-)$  do
15    if  $visit(q^+) = \mathbf{false}$  and  $dist(q^-) + c(q^+) < dist(q^+)$  then
16       $dist(q^+) \rightarrow dist(q^-) + c(q^+)$ 
17 return shortest paths

```

three enhancements that allow to retain the optimality of the algorithm and one enhancement that does not guarantee optimality.

Removing suboptimal nodes and arcs

For some nodes and arcs, we can immediately see that they will not be part of the shortest path because the cost on the node is too high or the joint costs of two nodes connected by an arc is too high. Propositions 1 and 2 identify several of these cases where nodes and arcs can be eliminated from the graph. This also leads to identifying parcels for which the pricing problem does not need to be solved because no column with positive reduced cost exists for that parcel. By eliminating nodes and arcs, the size of the graph can be reduced, which improves the speed of the shortest path algorithm.

Proposition 1 (Disregarding nodes). *Let $\underline{\rho} = \min_{p \in P} \rho_p$ and $\underline{w} = \min_{c \in C, s \in S_c} w_{cs}$. A parcel $p \in P$ can be disregarded if $\rho_p - v_p - 2\underline{w} \leq 0$. The corresponding pickup and dropoff segments (nodes) can then also be disregarded. A segment (node) $s \in S_c$ of crowd-shipper $c \in C$ can be disregarded if $\underline{\rho} - u_c - w_{cs} - \underline{w} \leq 0$ or $\underline{\rho} - u_c - w_{cs} - 2\underline{w} \leq 0$ if s is a transfer segment (node).*

Proposition 2 (Disregarding arcs). *Let $\underline{\rho} = \min_{p \in P} \rho_p$ and $\underline{w} = \min_{c \in C, s \in S_c} w_{cs}$. An arc between two nodes $s_1 \in S_{c_1}$ of crowd-shipper $c_1 \in C$ and $s_2 \in S_{c_2}$ of crowd-shipper $c_2 \in C$ can be disregarded if $\underline{\rho} - u_{c_1} w_{c_1 s_1} - u_{c_2} w_{c_2 s_2}$ or $\underline{\rho} - u_{c_1} w_{c_1 s_1} - u_{c_2} w_{c_2 s_2} - \underline{w}$ if either s_1 or s_2 is a transfer segment.*

As we consider multi-stage deliveries, a parcel path consists of at least two segments (i.e., a pickup and a dropoff segment). In case the considered segment is a transfer segment, there are at least two other segments involved. Using this property, the proof of these propositions is straightforward.

Constructing several smaller subgraphs

The nodes of the considered graph of the shortest path problem are partitioned into three categories: pickup nodes (N_P), dropoff nodes (N_D) and transfer nodes (N_T). Transfer nodes are independent of the specific parcels and only depend on crowd-shippers' itineraries. Pickup and dropoff nodes, for their part, depend on the specific parcel through the origin, destination and delivery time window. Constructing separate graphs, hereafter referred to as subgraphs, that only include a part of the pickup and dropoff nodes can solve memory issues, at the cost of a slight increase in computation time. Transfer nodes, finally, need to be included in every subgraph to guarantee the optimality of the solution.

We consider a fraction $\eta \in (0, 1]$ of the parcels that are included in a subgraph. This means that $1/\eta$ subgraphs are constructed for which the pricing problems are solved separately. Basically, the value of η forms a trade-off between time-savings and memory-savings, as well as the number of subgraphs and the size of those subgraphs. When η is small, subgraphs are small and therefore do not lead to memory issues, but many subgraphs need to be constructed at the cost of extra computation time. When η is large, subgraphs are larger, which may lead to memory issues, but fewer subgraphs need to be constructed which is generally faster.

Randomly removing highly similar nodes

Whereas the aforementioned enhancements improve the speed of the algorithm and reduce the memory consumption without jeopardizing optimality, we now turn to a method that can very successfully reduce the size of the graph but can no longer guarantee optimality. Due to the nature of our problem, many of the segments (and therefore nodes in the graph) are highly similar and therefore likely unnecessary. For example, many transfer segments between the same two transfer hubs may exist, but with different crowd-shippers at slightly different times. For this reason, many of those nodes can be removed without influencing optimality. However, as we do not know in advance whether such a node will be in a shortest path or not, optimality can no longer be guaranteed. We maintain a fraction $\zeta \in [0, 1)$

of the nodes in the graph and remove the other $1 - \zeta$ (and the arcs connected to those nodes). These nodes are selected randomly and with equal probability. As this is repeated at every iteration of the column generation algorithm, different nodes can be removed across iterations. This limits the influence on optimality, yet maintains the goal of reducing the size of the graph.

6.3.5 Locker and shipper capacity

So far, we have assumed that lockers have an infinite capacity and that crowd-shippers can only carry a single parcel. In this section, we relax those assumptions and extend the formulation accordingly. This will come at the cost of increased complexity in both the master problem and the pricing problem but will lead to more realistic solutions.

Shipper capacity

Instead of assuming that a crowd-shipper can only make a single delivery, we now relax this assumption and allow crowd-shippers to make multiple deliveries. We assume that crowd-shippers only perform multiple pickups and deliveries if they involve the exact same itinerary. That is, a crowd-shipper may carry multiple parcels at the same time, but only if they are picked up and delivered at the exact same stations. The reason for this is that significant effort is involved with every pickup and delivery (i.e., stopping at a locker, collecting or storing the parcel and continuing the journey). Whereas this can be largely consolidated if the pickup and drop-off locations are the same for the different items, this is not the case if these locations are different. We denote the capacity of a crowd-shipper $c \in C$ by Q_c .

To efficiently model the capacity without using a simultaneous column and row generation approach, we duplicate every segment in S_c a total of Q_c times. We redefine the set S_c by introducing S_c^q with $1 \leq q \leq Q_c$ as the q^{th} copy of the set of segments and $S_c = \cup_{q=1}^{Q_c} S_c^q$. For the sake of notation, let $s_1 \sim s_2$ denote the property that segments s_1 and s_2 are copies of each other and $s_1 \not\sim s_2$ the fact that they are not copies. Then, we reformulate problem (6.2) - (6.5) by replacing Constraints (6.4) by the following set of constraints, which ensures the capacity of a crowd-shipper:

$$\sum_{k \in K} a_{ck} x_k \leq Q_c \quad \forall c \in C \quad (u_c). \quad (6.15)$$

In addition to this, we add the following set of constraints to enforce that only segments that are duplicates of each other are performed by the same crowd-shipper. A pair of segments that are not duplicates of each other are deemed *incompatible* and columns that cannot be used together because of such an incompatibility are

part of an incompatible set I . The full collection of incompatible sets is denoted as $\mathcal{I}^{\text{crowd}}$ with $I \in \mathcal{I}^{\text{crowd}}$. Let b_{Ik} be a binary parameter taking value 1 if parcel path k uses a segment that is part of incompatible set I , and 0 otherwise. Basically, the set I contains all paths that are incompatible because they contain one of two incompatible segments s_1 and s_2 for which it holds that $s_1 \in S_c$ and $s_2 \in S_c$ for some crowd-shipper $c \in C$ and $s_1 \approx s_2$. Every set I , therefore, corresponds to a pair of incompatible segments (s_1, s_2) . The following set of constraints is added to exclude incompatibilities, with dual variable δ_I for every constraint $I \in \mathcal{I}^{\text{crowd}}$:

$$\sum_{k \in K} b_{Ik} x_k \leq 1 \quad \forall I \in \mathcal{I}^{\text{crowd}} \quad (\delta_I). \quad (6.16)$$

Instead of adding all constraints, which is computationally impossible due to the large number of segments, we only add those constraints that are violated in the current solution. We can still guarantee optimality as satisfied constraints do not influence the solution or the objective function. Given that they are inactive, their dual variable is by definition equal to 0 and therefore this also does not influence the pricing problem. The procedure to identify violated constraints is as follows. We define in advance all possible combinations of segments that would constitute a violation. That is, we identify all possible $I \in \mathcal{I}$. Then, every time the master problem is solved, for all the newly added columns we verify whether they contain a segment that is in any $I \in \mathcal{I}$. For every violation $I \in \mathcal{I}$ we maintain the set of columns that contain any segment in this set. We note that the violation $I \in \mathcal{I}$ is only added to the master problem if the corresponding set of columns contains more than one column.

The new reduced cost then looks as follows, where we identify if parcel path k contains a segment that makes it part of any of the incompatible sets $I \in \mathcal{I}$:

$$r_k = \pi_k - v_p - \sum_{c \in C} u_c a_{ck} - \sum_{I \in \mathcal{I}^{\text{crowd}}} b_{Ik} \delta_I. \quad (6.17)$$

The pricing problem remains the same apart from an extra cost δ_I that is subtracted whenever the new column is part of an incompatible set. We emphasize that the computational complexity of the pricing problem remains unchanged after the addition of the capacity constraint. Even though we duplicate the number of segments by the capacity, only the duplicate segment s with the lowest value of $\sum_{I \in \mathcal{I}^{\text{crowd}}} b_{Ik} \delta_I$ is considered in the pricing problem. The reason for this is that the duplicate segments are identical. Therefore, a segment s' with a higher sum of dual variables can never be in the shortest path, as replacing it with segment s would always reduce the cost of the path.

Locker capacity

In our framework, we allow parcels to be stored in parcel lockers at the transfer point. So far, we assumed that parcel lockers had infinite capacity. Here, we limit the number of parcels that can be stored at a transfer point $h \in H$ to be \bar{Q}_h . Similar to the capacity of the crowd-shippers, we identify sets of columns that are incompatible because the capacity of a locker is exceeded at some point in time. The full collection of incompatible sets is denoted as $\mathcal{I}^{\text{locker}}$. We adapt the master problem by adding the same set of constraints as in (6.16), but for the new collection:

$$\sum_{k \in K} b_{Ik} x_k \leq 1 \quad \forall I \in \mathcal{I}^{\text{locker}} \quad (\delta_I). \quad (6.18)$$

Again, we do not add all constraints at once but identify those that are violated. Although the number of transfer points is much lower than the number of crowd-shippers, the capacity of the transfer point needs to be considered at every time interval. We only consider transfer points with transfer lockers, as direct time-synchronized transfers do not need to be stored and therefore do not influence the capacity. To identify the violated constraints, we use the following procedure. For every transfer point, we identify the parcel paths that store a parcel at this point. We sort the parcel paths twice: once in ascending order of their arrival time at the transfer point and once in ascending order of their departure time from the transfer point. We start with an empty set of paths V . We then go over those events one by one in chronological order. Every time an arrival is recorded, the parcel path is added to V . Every time a departure is recorded, the parcel path is removed from V . Whenever a parcel arrives that causes the cardinality of V to exceed \bar{Q}_h , we add violation I with $b_{Ik} = 1$ for all $k \in V$ and we store the time t_{Ih} at which the violation occurs, which will later aid the pricing problem. For every transfer point, we only add a single constraint and then re-solve the master problem. This is repeated until no violated constraints are encountered.

The reduced cost can then be computed as follows, where we identify if a parcel path k stores a parcel at transfer point h at time t_{Ih} for any of the incompatibilities $I \in \mathcal{I}^{\text{locker}}$:

$$r_k = \pi_k - v_p - \sum_{c \in C} u_c a_{ck} - \sum_{I \in \mathcal{I}^{\text{crowd}}} b_{Ik} \delta_I - \sum_{I \in \mathcal{I}^{\text{locker}}} b_{Ik} \delta_I. \quad (6.19)$$

If this is the case, $b_{Ik} = 1$ for the new parcel path. The pricing problem can then be extended by exploiting the start and end times of every segment. We recall that every node in the network discussed in Section 6.3.2 corresponds to a segment. So far, a node corresponding to a segment $s \in S_c$ of crowd-shipper $c \in C$ was attributed a cost $u_c + w_{cs}$, and no costs were attributed to arcs. Now, for an arc

between two nodes corresponding to segment s_1 with end time \underline{t} and s_2 with start time \bar{t} where $d_{s_1} = o_{s_2} = h$ a cost of δ_I is added for every $I \in \mathcal{I}^{\text{locker}}$ for which it holds that $\underline{t} \leq t_{Ih} \leq \bar{t}$. We note that if an arc violates multiple constraints, multiple dual variables can be added to the same arc.

6.4 Results

We describe the details of the case study and the parameter settings in Section 6.4.1. In Section 6.4.2 we evaluate the performance of the algorithm in terms of optimality gap and computation time. We compare the results of our approach to a locally optimal dynamic assignment strategy in Section 6.4.3. We evaluate the effect of crowd-shipper capacity and locker capacity in Sections 6.4.4 and 6.4.5, respectively. Finally, we perform a sensitivity analysis on the cost parameters in Section 6.4.6.

6.4.1 Case study

The city of Washington DC is used as a case study. We use data on the spatial distribution of the population (Census Reporter, 2021) and the movement of individuals throughout the city based on bike-sharing users (Capital Bikeshare, 2020). The bike-sharing system of Washington DC has over 500 stations and 4500 bikes, making it one of the largest in the USA. A selection of 240 stations that are in the city center or the closest suburbs is used in our case study. Bike-sharing stations are considered as demand locations. This can either be through parcel lockers or home delivery to an individual living arbitrarily close to a station. Thereby, historical data on the movement of bike-sharing users throughout the city is used to approximate the movement of potential crowd-shippers.

The case study and the construction of the dataset are highly similar to that of Stokkink and Geroliminis, 2023. The main difference is that the size of the network we consider in this work is more than three times as large. Thereby, we consider time-dependent arrival rates of crowd-shippers. For a detailed description of how the case study is constructed, the reader is referred to their work. Figure 6.4 displays a bubble chart of the considered network, where the size of the bubble is determined relative to the population around the corresponding station. Whereas most commuters travel around Union Station, the Mall, and the center of Washington DC, most people live in the suburbs and this is therefore where demand is the highest. We note the large asymmetry in supply and demand for a crowd-shipping system in an urban network, making our case study highly realistic.

The baseline parameters used for the model and the column generation algorithm are given in Table 6.1. These parameters are used in all numerical experiments, except for sensitivity analyses on these parameters. According to an analysis from

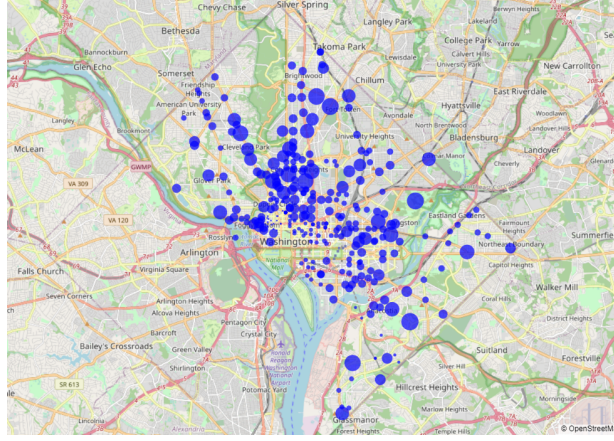


Figure 6.4: Bubble chart of bike-sharing stations, where the size of the bubble is determined by the population in the area.

American survey data in Le and Ukkusuri, 2019a, on average, crowd-shippers expect a compensation of 12\$ per hour. Considering 10 minutes to perform both the pickup and delivery, we set $\alpha^1 = \$2$. Using an average bikers speed of 12km/h, we set $\alpha^3 = \$1/km$, which is similar to the value chosen by Le, Ukkusuri, Xue, and Van Woensel, 2021. Intuitively, $\alpha^2 > \alpha^3$ and therefore we set $\alpha^2 = \$2/km$. This is in line with the findings of Rougès and Montreuil, 2014, who studied 26 crowd-shipping businesses, that found the prices of intra-urban deliveries to start between \$4 and \$10 plus additional charges for inconveniences such as heavy loads and long distances. According to Le and Ukkusuri, 2019a, a traditional carrier charges \$15 per parcel. To accommodate distance aspects, we set the cost per parcel to a base cost of \$10, which can increase up to \$15 with \$2.00 per kilometer between the origin and destination of the parcel. The maximum runtime of the algorithm is set to 1800 seconds. The maximum runtime is checked before every call to the pricing problem and may therefore be slightly exceeded.

The base case we consider has two origin locations and we construct a subgraph for every origin in the pricing problem. This means $\eta = 1/2$. Parcels are stored at a random origin in the morning and not necessarily at the closest origin to the destination. The relative rate of parcels and crowd-shippers ($|C|/|P|$) is fixed. For computational reasons, we reduce the set C by removing crowd-shippers that cannot contribute to any delivery (complete or partial). This yields the reduced set C' . The number of crowd-shippers in $|C'|$ depends on other parameters such as the transfer locations H and the maximum detour τ . Therefore, in the experiments that follow, the reported ratio $|C'|/|P|$ is not necessarily constant.

CPLEX version 12.6.3.0 is used in Java to solve all ILPs and LPs. The LPs during the iterations of the column generation algorithm are solved to optimality

and the IP after the final iteration of the column generation algorithm is solved up to a 0.5% optimality gap.

Table 6.1: Parameter settings

Model parameters	
α^1	\$1.00/parcel
α^2	\$2.00/parcel/km
α^3	\$1.00/parcel/km
τ	100 meters
Δ^{\min}	1 minute
Δ^{\max}	10 hours
ρ	$\min\{\$15.00, \$10 + \$2.00/\text{km}\}/\text{parcel}$
Algorithm parameters	
η	0.5
ζ	0.3
CPU time limit	1800 seconds

6.4.2 Algorithm evaluation

In this section, we evaluate the performance of our column-generation algorithm in terms of objective value and computation time. We evaluate the performance for various model parameters and problem sizes. Thereby, we compare the performance of the algorithm for multiple levels of ζ . The results are displayed in Table 6.2. Clearly, the computation time of the algorithm increases as the size of the problem increases. The most important determinant of the complexity of the algorithm is the number of segments that are used to construct the graph in the pricing problem. Therefore, the computation time increases drastically with $|P|$, $|C|$, and τ . This also explains why using only a random portion of the segments to construct the graph in every iteration leads to a significant reduction in computation time. By using a portion ζ , the computation time is reduced almost by a factor 10. Hence, larger instances can be solved without decomposing the pricing problem over more subgraphs.

The algorithm finds optimal or near-optimal solutions. When ζ is 1 and the algorithm converges before the time limit, we can use the LP solution as an upper bound to the objective value and therefore compute an optimality gap. For $\zeta < 1$, the LP solution is not necessarily an upper bound. Hence, we only compute the optimality gap if the optimal LP solution is found for $\zeta = 1$. The optimality gap is at most 0.5% for all tested instances for which the optimality gap was computable.

Even when $\zeta = 0.3$, the optimality gap is almost negligible. Furthermore, using transfers leads to an improvement between 15% and 50% both in the objective value (i.e., revenue - costs) and the service level (i.e., number of served parcels).

Table 6.2: Algorithm Evaluation

$ P $	$ C' $	τ	$ H $	ζ	CPU time (s)	Opt. gap (%)	Obj. (\$)	Gain (%)	SL	Gain (%)
310	339	250	6	1	4.0	0.0	109472	30.7	111	35.4
310	491	500	6	1	23.2	0.0	147542	24.5	152	28.8
681	798	250	6	1	88.5	0.0	271569	18.0	269	20.1
681	1034	500	6	1	952.3	0.5	375937	13.2	384	16.0
1043	1204	250	6	1	537.4	0.1	398558	14.6	400	17.0
1043	1776	500	6	1	-	-	-	-	-	-
310	442	250	11	1	17.6	0.0	117926	40.8	120	46.3
310	577	500	11	1	284.5	0.0	163944	38.4	172	45.8
681	988	250	11	1	848.7	0.4	281295	22.2	280	25.0
681	1205	500	11	1	*1800.0	-	362884	9.2	377	13.9
1043	1464	250	11	1	*1800.0	-	419287	20.6	423	23.7
1043	1776	500	11	1	-	-	-	-	-	-
310	339	250	6	0.3	2.1	0.0	109472	30.7	111	35.4
310	491	500	6	0.3	5.4	0.0	147565	24.5	152	28.8
681	798	250	6	0.3	13.6	0.0	271571	18.0	269	20.1
681	1034	500	6	0.3	174.7	0.4	376202	13.2	385	16.3
1043	1204	250	6	0.3	67.4	0.0	398895	14.7	400	17.0
1043	1535	500	6	0.3	1418.4	-	547657	12.8	565	16.5
310	442	250	11	0.3	3.4	0.0	117919	40.8	120	46.3
310	577	500	11	0.3	17.1	0.2	163682	38.2	172	45.8
681	988	250	11	0.3	39.2	0.0	282302	22.6	281	25.4
681	1205	500	11	0.3	591.1	-	403945	21.6	421	27.2
1043	1464	250	11	0.3	203.8	-	419365	20.6	422	23.4
1043	1776	500	11	0.3	*1800.0	-	582371	19.9	608	25.4

Note: $|P|$ = number of parcels, $|C'|$ = number of potential crowd-shippers, τ = maximum detour of crowd-shippers, $|H|$ = number of transfer hubs, ζ is the portion of random segments used in the construction of the subgraph for the pricing problem. The optimality gap is the percentage difference between the IP solution and the LP solution for $\zeta = 1$. SL = service level. The gain columns display the improvement that is obtained by using transfers over not using transfers. Scenarios for which the CPU time limit is reached and therefore no optimality gap can be obtained are denoted with an asterisk. The two largest instances for $\zeta = 1$ cannot be solved due to memory issues.

To further evaluate the effect of ζ on computation time and optimality gap, we evaluate the case where $|P|=310$, $|C'|=491$, $\tau=500$, and $|H|=11$ for 6 different values of ζ . In Figure 6.5, the optimality gap is displayed relative to the number of iterations (left) and the computation time in seconds (right). Clearly, the computation time per iteration decreases drastically by decreasing the random portion of segments that are used at every iteration. However, because the subgraphs are not complete, they may lead to not all columns with positive reduced costs being identified in an iteration. Therefore, the algorithm may require more iterations and can lead to suboptimal solutions. The best value of ζ is thus a trade-off between the number of iterations and the computation time per iteration. The best value is also dependent on the size of the problem. In general, for larger problems, smaller values of ζ can be chosen at the cost of limited losses.

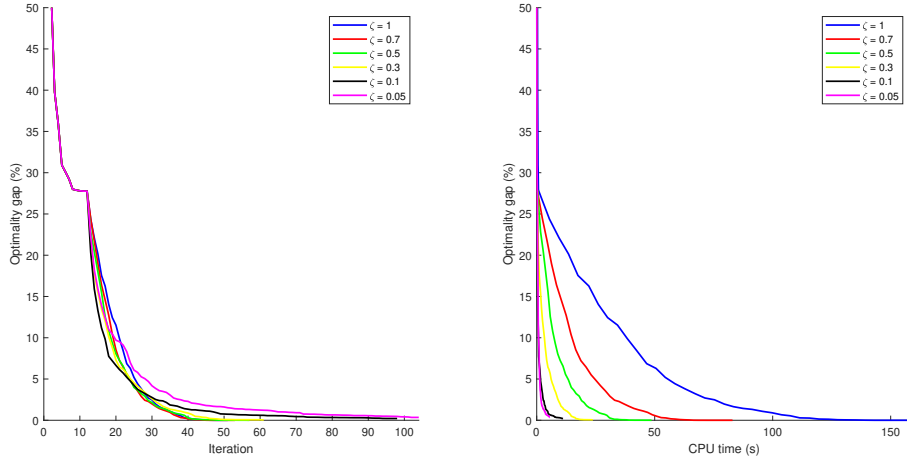


Figure 6.5: Iterative optimality gap for different values of ζ

6.4.3 Performance compared to locally optimized benchmark

In this section, we compare the performance of our optimized assignment procedure to a myopic first-best assignment policy. Such a policy is commonly applied for dynamic settings in practice and in the literature (???).

Crowd-shippers arrive dynamically over time. Every time a crowd-shipper arrives, the first-best parcel is selected which maximizes the revenue. That is, we choose the parcel that is locally optimal, ignoring information about potential future crowd-shippers. For a direct delivery, the profit can be computed exactly as the full trip is known. For an indirect delivery, the costs of the current delivery stage are known. The costs of previous delivery stages have already been incurred and can be considered sunk costs, which are therefore omitted from the optimization. It is assumed that after the current stage, the parcel is directly picked up from the transfer point and taken to the final destination of the parcel. Due to coordination issues in the dynamic arrival of crowd-shippers, we only allow for two-stage deliveries and strictly prefer delivering a parcel to the final destination over delivery to a transfer point. With these cost components and the revenue obtained from delivering the parcel, the expected profit can be computed. Since the strategy is myopic, no information is used on potential future crowd-shippers.

We note that for the myopic first-best assignment policy, parcels may remain at transfer points whereas for the optimized assignment this is not possible. In case a parcel remains at the transfer point, the revenue is not obtained although a part of the costs is already incurred. For the sake of comparison, we compare the optimized assignment to two dynamic benchmarks. One where the costs for uncompleted

deliveries are excluded (B1) and one for which the costs for uncompleted deliveries are included (B2).

The results are displayed in Table 6.3 where the first set of rows denotes the results for $|H|=0$, implying that only direct deliveries are allowed and the second set of rows denotes the results for $|H|=11$ where transfers are allowed. Global optimization allows for the coordination of transfers. As a consequence, global optimization outperforms local optimization by 25% in terms of service level and objective value when transfers are allowed. Without transfers, the effect is only 5%.

Table 6.3: Benchmark comparison

$ P $	$ C' $	τ	$ H $	Global optimization		Local optimization			Effect of global optimization		
				SL	Obj	SL	B1	B2	SL	B1	B2
310	442	250	0	26.5	837.40	25.2	797.75	797.75	-4.9	-4.7	-4.7
310	577	500	0	38.1	1184.80	37.1	1144.06	1144.06	-2.5	-3.4	-3.4
681	976	250	0	32.9	2301.94	32.3	2251.91	2251.91	-1.8	-2.2	-2.2
681	1202	500	0	48.6	3322.31	45.5	3107.11	3107.11	-6.3	-6.5	-6.5
1043	1447	250	0	32.8	3477.75	31.4	3328.68	3328.68	-4.1	-4.3	-4.3
1043	1774	500	0	46.5	4856.33	44.9	4650.70	4650.70	-3.5	-4.2	-4.2
310	442	250	11	38.7	1179.20	29.4	905.14	864.79	-24.2	-23.2	-26.7
310	577	500	11	55.2	1633.30	40.6	1208.52	1098.46	-26.3	-26.0	-32.7
681	976	250	11	41.3	2823.58	34.5	2367.28	2298.17	-16.4	-16.2	-18.6
681	1202	500	11	61.2	4018.04	48.3	3198.28	3044.50	-21.1	-20.4	-24.2
1043	1447	250	11	40.6	4200.88	33.3	3471.00	3379.15	-18.0	-17.4	-19.6
1043	1774	500	11	58.9	5869.61	47.2	4768.41	4522.80	-19.9	-18.8	-22.9

Note: $|P|$ = number of parcels, $|C'|$ = number of potential crowd-shippers, τ = maximum detour of crowd-shippers, $|H|$ = number of transfer hubs, SL = service level given as a percentage, obj = objective value given in dollars, B1 and B2 are the objective values of two local optimization benchmarks given in dollars. The last three columns denote the percentual difference between the local and the global optimization strategies.

6.4.4 Crowd-shipper capacity

In this section, we evaluate the influence of crowd-shipper capacity on the profit and service level. We consider that a part of the potential crowd-shippers can carry multiple parcels at the same time. When generating the instance, every crowd-shipper has an equal probability for each capacity level, such that we obtain an evenly distributed population. The results are obtained for $\zeta = 0.05$, to further reduce the CPU time. Here, we also apply the described row generation procedure to identify violated constraints that we iteratively add to the formulation. As a result of using $\zeta = 0.05$, the obtained LP solution is not necessarily optimal. Therefore, the optimality gap is an approximation.

The results are displayed in Table 6.4. By considering higher capacities, computation times increase drastically. For this reason, a time limit of 3600 seconds (1 hour) is used instead. The reason for the increased computation times is two-fold. First,

we consider duplicate segments, such that the number of considered segments and therefore the computation time of the pricing problem increases. Second, the violated constraints need to be identified and added to the master problem, which makes solving the master problem more computationally demanding. Furthermore, we observe that the optimality gap increases with capacity. However, the highest observed optimality gap is 9%, which is deemed reasonable.

We note that to obtain an optimal solution the column generation framework would have to be integrated into a branch-and-price framework. However, given the relatively small optimality gap, the already substantial computation time of the column generation algorithm, and the fact that we are dealing with an operational problem rather than a strategic one, developing a branch-and-price framework looks unappealing for our specific problem.

The increase in capacity leads to a substantial improvement in the objective (profit) and the service level. Depending on the problem setting, using a capacity of 2 for half of the population improves the objective and service level by 20% to 50%. For higher capacities, the observed increase is even higher, even though the algorithm has reached the time limit before the optimal solution has been found.

Table 6.4: Influence of crowd-shipper capacity

$ P $	$ C' $	q_c	τ (m)	CPU time (s)	Obj. (\$)	SL (%)	Opt. gap (%)	Gain obj. (%)	Gain SL (%)
310	442	{1}	250	2.8	1173.64	38.7	0.0	-	-
310	442	{1,2}	250	139.6	1526.89	51.0	3.5	30.1	31.7
310	442	{1,2,3}	250	272.1	1637.73	54.2	9.0	39.5	40.0
310	577	{1}	500	4.4	1633.81	55.5	0.0	-	-
310	577	{1,2}	500	775.3	1996.02	67.7	3.0	22.2	22.1
310	577	{1,2,3}	500	2192.0	2130.94	72.3	6.2	30.4	30.2
681	976	{1}	250	10.2	2821.64	41.3	0.0	-	-
681	976	{1,2}	250	*3600.0	3997.53	60.1	-	41.7	45.6
681	976	{1,2,3}	250	*3600.0	4314.73	65.3	-	52.9	58.4
681	1202	{1}	500	89.9	4018.09	61.5	0.4	-	-
681	1202	{1,2}	500	*3600.0	4317.14	64.0	-	7.4	4.1
681	1202	{1,2,3}	500	*3600.0	-	-	-	-	-

Note: $|P|$ = number of parcels, $|C'|$ = number of potential crowd-shippers, q_c is the considered crowd-shipper capacity, each with equal probability, τ = maximum detour of crowd-shippers, The optimality gap is the percentage difference between the IP solution and the LP solution. Since we use $\zeta = 0.3$, the optimality gap is not exact but an approximation. SL = service level. The gain columns display the improvement that is obtained by increasing the capacity. Scenarios for which the CPU time limit is reached and therefore no optimality gap can be obtained are denoted with an asterisk. The largest instance cannot be solved due to memory issues.

Figure 6.6 displays a Gantt chart of the movement of parcels. Each colored bar represents the time spent with a crowd-shipper. Identical bars in identical locations signal that a crowd-shipper is carrying multiple parcels at the same time. A deeper investigation reveals that the additional flexibility leads to parcels being

carried collectively on one leg and separately on the other, which is clear from the zoomed figure. We also observe the influence of travel patterns on crowd-shipping activity. We observe a clear morning and evening peak, by the frequency of the activities. The evening peak contains significantly more activities, despite the number of potential crowd-shippers not being significantly different from the morning commute. The reason for this is that most destinations for parcels are in the suburbs. Hence, the evening commute from the center to the suburbs is more useful for reaching these destinations.

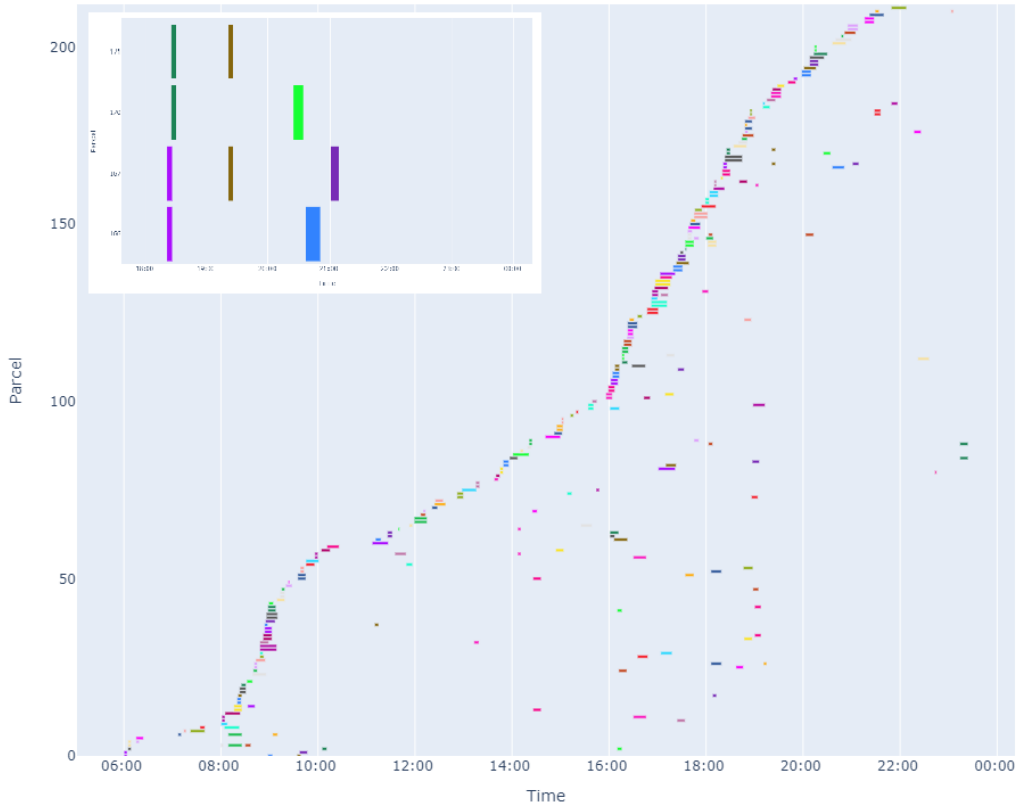


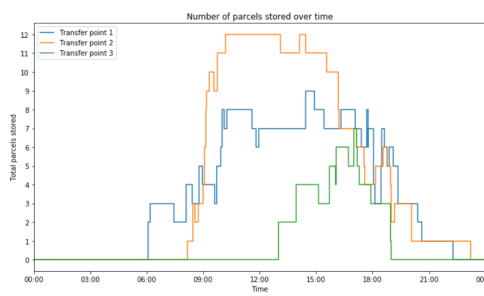
Figure 6.6: Gantt chart of the movement of parcels with time on the x-axis and the parcel index on the y-axis. Parcels are sorted by the start time of the first segment. Each colored bar represents the time spent with a crowd-shipper. In the upper left corner, we zoom on four specific parcels.

6.4.5 Parcel locker capacity

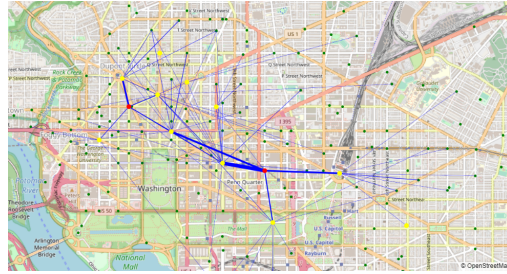
As parcel locker capacity does not seem to be a restrictive parameter for reasonable values of \bar{Q}_h , these constraints are not considered in the obtained results. In this section, we discuss the evolution of locker capacity over time. Out of the 11 transfer hubs, we specifically focus on three locations that have distinct patterns. The

locations are identified in Figure 6.7b, where origins are marked in red, transfer points are marked in yellow, destinations of parcels are marked in green and the flow of parcels that make at least one transfer is marked by blue lines. Here, the size of the line denotes the number of parcels. The total number of parcels stored in the transfer points is displayed in Figure 6.7a.

The three chosen transfer points are the most used among a total of 11. It is clear that a capacity of 10, therefore, suffices for all transfer points. The first two points are in the city center. This is clear because they fill up quickly during the morning commute after which they are emptying out slowly during the evening commute. During the morning commute, potential crowd-shippers travel from the suburbs to the city center, passing by these transfer points. The opposite is observed for the third transfer point, which is at the main train station of Washington DC, parcels gradually accumulate throughout the day before being emptied out rapidly during the evening commute when people are traveling back home (i.e., to the suburbs) from the train station. Clearly, the results in Figure 6.7 align with the results in Figure 6.6.



(a) Number of parcels stored over time



(b) Flow of parcels that make at least one transfer

Figure 6.7: Step and flow charts that indicate the number of parcels stored at transfer points over time and the flow of parcels in the network

6.4.6 Sensitivity for cost parameters

In this section, we evaluate the effect of the cost parameters on the observed performance and the number of transfers per parcel path. We consider similar settings as in the previous experiment, but with a constant $|H|=11$ and $\zeta=0.3$. We consider non-linear cost components for crowd-shipper compensation where we replace the profit in Equation (6.1) with the following function:

$$\pi_k = \rho_p - \left[\sum_{c \in C} a_{ck} \alpha^1 + \sum_{c \in C} \sum_{s \in S_c} b_{csk} \left(\alpha^2 (det_{cs})^{\beta_2} + \alpha^3 (len_s)^{\beta_3} \right) \right]. \quad (6.20)$$

The results are displayed in Table 6.5. We observe that the effect of transfers on service level is relatively constant for different cost parameter combinations. The effect on the objective improvement is more substantial. For higher values of τ , the relative improvement of the objective function compared to the case $|H|=0$ is lower than for lower values of τ . Thereby, if the penalty for distance traveled with a parcel is non-linear, the improvement of the objective decreases by approximately 10%. The value of α_1 has a significant influence on the number of transfers on a path. When the fixed compensation is negligible, transfers become more beneficial and we observe significantly more paths with two or more transfers.

Table 6.5: Sensitivity to cost parameters

$ P $	$ C' $	τ	α_1	α_2	α_3	β_2	β_3	Improvement over $ H =0$ Obj. (%)	SL (%)	Transfers per path (%)		
										0	1	2+
310	442	250	1	2	1	1	1	40.8	46.3	66.7	33.3	0.0
310	577	500	1	2	1	1	1	38.4	45.8	66.3	32.0	1.7
310	442	250	1	1.6	1	1.2	1	40.3	48.1	65.0	34.2	0.8
310	577	500	1	1.6	1	1.2	1	34.4	46.2	64.9	33.3	1.8
310	442	250	1	2	0.8	1	1.2	29.7	46.3	64.2	34.2	1.7
310	577	500	1	2	0.8	1	1.2	23.1	43.2	68.0	30.8	1.2
310	442	250	1	1.6	0.8	1.2	1.2	28.3	46.9	66.4	32.8	0.8
310	577	500	1	1.6	0.8	1.2	1.2	18.6	32.5	70.3	29.0	0.6
310	442	250	0.01	2	1	1	1	46.2	46.3	56.7	38.3	5.0
310	577	500	0.01	2	1	1	1	43.8	45.8	60.5	33.7	5.8
310	442	250	0.01	1.6	1	1.2	1	46.4	48.1	56.7	38.3	5.0
310	577	500	0.01	1.6	1	1.2	1	41.3	47.0	58.7	34.3	7.0
310	442	250	0.01	2	0.8	1	1.2	39.3	46.3	49.2	42.5	8.3
310	577	500	0.01	2	0.8	1	1.2	33.9	44.1	51.8	38.8	9.4
310	442	250	0.01	1.6	0.8	1.2	1.2	38.0	46.9	53.8	42.0	4.2
310	577	500	0.01	1.6	0.8	1.2	1.2	29.0	41.0	53.9	38.8	7.3

Note: $|P|$ = number of parcels, $|C'|$ = number of potential crowd-shippers, τ = maximum detour of crowd-shippers, α_1 = fixed crowd-shipper compensation in dollars, α_2 = variable crowd-shipper compensation per km detour, α_3 = variable crowd-shipper compensation per km traveled with parcel, β_2 = power of detour component, β_3 = power of distance component. SL = service level. α_1 , α_2 , and α_3 are given in dollars (per kilometer).

6.5 Summary

In this chapter, we developed a crowd-shipping model with intermediate transfers. In contrast with the majority of the existing literature, our model allows for high levels of heterogeneity of crowd-shippers, parcels, and transfer points. We consider a detailed individual-specific cost structure for crowd-shipper compensation and allow for different weights to be assigned to different parcels, for example, to differentiate between locations in the network. Thereby, we allow for direct time-synchronized transfers, where a parcel is directly handed from one crowd-shipper to another, as well as transfers with intermediate storage at strategically located parcel lockers. We designed a column generation algorithm to solve large-scale realistic scenarios to optimality within a reasonable amount of time. To improve the performance of

the system, we allow crowd-shippers to carry more than one parcel at the same time. This further complexifies the problem, as additional constraints are required to regulate crowd-shipper capacity and compatibility of parcels. To solve this problem, we extend our column generation algorithm to simultaneous column and row generation. This algorithm identifies violated compatibility constraints and adds these to the master problem after every column generation iteration.

For a large network with 250 regions, 11 transfer points, approximately 500 parcels, and 500 crowd-shippers, our algorithm finds the optimal matching within one minute. For larger models of approximately 1000 parcels and 1000 crowd-shippers, we find solutions that are optimal or near-optimal within 10 to 30 minutes. Our results indicate that the use of parcel lockers for intermediate transfers allows for increasing the total revenue and service level by around 30%, depending on the system configurations. A further increase of 30 to 50% can be obtained by allowing some crowd-shippers to carry two or three parcels at the same time (with an average capacity of two across the population). Due to the complexity of coordination between crowd-shippers in a system with transfers, our optimal approach outperforms a myopic first-best (locally optimized) approach by 25%.

7

Conclusions and future research

In this thesis, we developed optimization models for strategic and operational problems in last-mile logistics and transport systems. The focus is on improving the sustainability and efficiency of these systems, by leveraging existing vehicle flows and stimulating multi-purpose trips for commuters. In Chapter 2 we developed a predictive user-based vehicle relocation strategy for one-way car-sharing systems. In Chapter 3 we identified the effect of dynamic congestion on matching in ride-sharing systems. In Chapter 4 we modeled the multi-modal ride-matching problem with transfers and travel time uncertainty as a stochastic programming problem. In Chapter 5 we focused on the strategic network design problem in an urban crowd-shipping system. We developed an algorithm based on lower-level continuum approximations that efficiently determines depot locations. In Chapter 6 we designed a column-and-row generation algorithm for the crowd-shipping matching problem with transfers. In this final chapter, we summarize the main contributions and findings of this thesis and identify promising areas for future research.

7.1 Main findings

Chapter 2: Predictive user-based relocation through incentives in one-way car-sharing systems

Chapter 2 focuses on the problem of user-based vehicle relocation in one-way car-sharing systems with the aim of finding an effective, low-cost, and sustainable alternative for staff-based vehicle relocation. A predictive incentive scheme is designed that determines the optimal incentive in a dynamic system framework. The main contributions are listed as follows:

- Design an innovative user-based relocation policy to solve the vehicle relocation problem in one-way car-sharing systems.
- Integrate a Markovian prediction method to forecast future demand losses due to vehicle imbalances.
- Propose an adaptive method that determines the optimal incentive based on (predicted) customers' value of time and current and expected future states of the system.
- Illustrate the benefits and performance of user-based relocation compared to staff-based relocation in terms of service level, cost, and sustainability.
- Describe the performance of a hybrid operator-user-based relocation policy in terms of service level, cost, and sustainability.

Specifically, the simulation results indicate that, by using our incentivization approach, we can partially solve the balancing problem of vehicles throughout the network and thereby increase the service level. In addition to this, our methods allow the operator to use fewer staff members while attaining a higher service level and thereby increase the profit. Specifically, by using a hybrid operator-user-based relocation policy, service level, and profit can be maximized. In this case, user-based relocations perform short-distance relocations, while long-distance relocations are executed by staff members. We also observe that by using user-based relocations, the average KM traveled by staff and users per unit of served demand decreases, suggesting our method is environmentally more sustainable than staff-based policies.

Future research can extend our model to include competition between users for incentives. If one user declines the offered incentive, a similar incentive may be offered to the next arriving user who may then choose to accept it. Including future decisions on incentives would result in a computationally expensive recourse problem and is therefore omitted in the current work, but is an interesting direction of future work for mobility systems with higher demand. By incorporating this type of competition for incentives in the optimization problem, offered discounts may be lower without decreasing the performance of the system.

Chapter 3: Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users

Chapter 3 focuses on the problem of matching drivers and passengers in a carpooling system. A bi-level optimization approach is proposed to evaluate the effect of congestion on the matching in such a system. The main contributions are listed as follows:

- Describe the matching of drivers and passengers in a carpooling framework through a MILP and fundamental theoretical properties.
- Design an integrated framework of carpooling matching and dynamic bottleneck congestion.
- Propose an iterative matching algorithm that incorporates dynamic bottleneck congestion.
- Provide insight into the effect of dynamic bottleneck congestion on the optimal carpooling matching solution.

Specifically, the numerical results indicate that carpooling is more attractive during rush hours when there is congestion. Firstly, this is caused by the economies of scale of the matching problem. When the number of drivers and passengers increases, the average cost of a match decrease. Secondly, bottleneck congestion makes drivers and passengers more flexible with respect to their departure time. In equilibrium, commuters may be (almost) indifferent between multiple departure times, which makes matching them to another commuter less costly. We also observe that in such a framework, carpoolers have a preference to depart either earlier or later as schedule delay penalties can be shared between passenger and driver, whereas delay costs are faced by all individuals sharing a ride. Due to the common assumption that lateness is penalized heavier than earliness, most carpoolers arrive early to their destination.

Chapter 4: Multi-modal ride-matching with transfers and travel-time uncertainty

Chapter 4 focuses on the problem of matching riders and drivers in a multi-modal framework where riders can transfer between drivers and modes. We also consider that drivers can carry multiple riders and we investigate the influence of travel time uncertainty. The main contributions are listed as follows:

- We develop a framework for multi-modal transport of riders that considers public transport, solo driving, and ride-sharing. The framework allows for transfers between modes and between drivers.
- We derive optimality conditions for the matching as well as the departure times.
- We formulate the multi-modal ride-matching problems with multiple transfer hubs as a path-based integer programming problem.
- We model the problem with uncertain travel times as a two-stage stochastic programming problem.

Specifically, the numerical results indicate that transfers allow to reduce the cost of riders. Thereby, a modal shift is observed from private and public transport to shared and multi-modal transportation. As a result of this, we observe a reduction in the vehicle hours traveled in private vehicles. A temporal analysis indicates that ride-sharing is mostly beneficial when the number of riders and drivers is higher. In this case, ride-sharing benefits from economies of scale that reduce the matching costs.

Travel time uncertainty may increase the costs of some matches and may even make some matches infeasible because the rider arrives after the driver has already departed. For this reason, we observe that uncertainty in travel times increases the costs of riders. Despite this increase, transfers grant additional flexibility to riders, as they can partially adapt their matching decisions at the transfer point after observing the realized travel times. We observe a large modal shift where a substantial portion of private car users changes to public transport or ride-sharing. As we assume public transport is unaffected by travel time uncertainty, this mode is especially appealing to riders when the variance increases. We observe that up to 25% of the riders change their mode choice after the realized travel time is revealed and up to 60% of the riders change their match after the realized travel time is revealed.

Chapter 5: A continuum approximation approach to the depot location problem in a crowd-shipping system

Chapter 5 focuses on the problem of finding the optimal depot locations in a large-scale urban crowd-shipping system. A depot-based crowd-shipping system is suggested to attract more crowd-shippers and reach more demand destinations. A heuristic algorithm based on continuum approximation is used to determine the optimal hub locations. The main contributions are listed as follows:

- Developing a continuum approximation of the operational problem of the assignment of parcels to crowd-shippers and parcels to depots.
- Developing a heuristic algorithm that uses the continuum approximation to accurately and efficiently approximate lower-level decisions and costs and with that search for the best depot locations.
- Propose predictive decision-making for strategic, tactical, and operational decisions that incorporate the interaction between expected supply and demand.

Specifically, the simulation results indicate the integrating depots in a crowd-shipping system allows to serve more demand. The best depot locations are not necessarily in geographically central locations, but most importantly depend on the supply and demand pattern in an urban area. Our heuristic approach is significantly faster than exact approaches, yet obtains highly similar results. The fast approximation algorithm makes the methodology applicable in large cities. Predictive components in three levels of decision-making significantly improve performance in terms of profit and service level.

Chapter 6: A column generation approach to the crowd-shipping problem with transfers

Chapter 6 focuses on the problem of finding the optimal matching of parcels and crowd-shippers. In the framework we consider, crowd-shippers may carry multiple parcels at the same time and parcels may transfer between multiple crowd-shippers. An exact solution algorithm that uses column and row generation techniques is used to find the optimal solution to the problem. The main contributions are listed as follows:

- Develop a framework for the crowd-shipping problem with transfers that allows for time-synchronized transfers and transfers where the parcel is temporarily stored.
- Consider a highly detailed cost structure for crowd-shipper compensation and heterogeneity among crowd-shippers and parcels.
- Design an exact column and row generation approach to solve the problem at hand to optimality

Specifically, the simulation results of a case study in the city of Washington DC show that transfers can improve the profit and service level of the crowd-shipping system by around 30%. Optimal solutions to instances with around 500 parcels and 500 crowd-shippers in a network with more than 200 nodes can be solved within minutes. Computation times can be reduced significantly by randomly reducing the pricing problem at every iteration. Although the method loses the guarantee of optimality, the obtained solutions are near-optimal in all studied cases.

A comparison to a first-best assignment policy that is locally optimal illustrates that our column generation approach performs approximately 25% better in terms of objective and service level. Without transfers, this is only 5%. This signals the additional complexity that transfers impose on the problem and the benefit of our approach that coordinates between crowd-shippers.

When part of the population of crowd-shippers can carry more than one parcel, this allows for substantial improvements in the performance of the crowd-shipping system. If half of the population can carry up to 2 parcels, the service level and profit can be increased by approximately 40%. A spatiotemporal analysis indicates a clear morning and evening peak, by the frequency of the activities. The evening peak contains significantly more activities, despite the number of potential crowd-shippers not being significantly different from the morning commute. The reason for this is that most destinations for parcels are in the suburbs, where the commuters are going in the evening.

7.2 Future research

Based on the developed methods and findings of the research included in this thesis, various interesting directions of future research arise, related to one or multiple of the main research fields explored in this thesis.

For both ride-sharing and crowd-shipping systems with transfers (Chapters 4 and 6, respectively), we focused on the operational problem of matching drivers and riders, and matching crowd-shippers and parcels. The strategic problem of determining the optimal locations for these transfer points remains an open problem and is an important direction of future research. The difficulty of this problem lies in the connection between incoming and outgoing flows of transfer points. Especially when multiple transfers can be made, in accordance with Chapter 6, there is a strong connection between multiple transfer points.

To solve these problems, a similar algorithm can be developed for transfer points as the one described in Chapter 5 for depot locations. Given that in these types of problems, the main cost component is on the operational level, whereas decisions are made on a strategic or tactical level, the optimization framework needs to encompass the uncertainty between these levels.

Increasing the number of participants in ride-sharing systems such as those described in Chapters 3 and 4 requires the design of subsidy, incentive, or pricing schemes. For such problems, developing an adaptive pricing scheme that minimizes the expected costs or maximizes participation is a promising direction for future research. Such pricing strategies can be implemented directly on the ride-sharing platform, or indirectly through HOV (High-Occupancy Vehicle) and HOT (High-Occupancy Toll) lanes. Through the use of these dedicated lanes, ride-sharing can be promoted as the use of these lanes is generally faster and therefore allows commuters to reduce their generalized costs. Research in this direction can also focus on self-financing policies where a toll can be imposed on solo-driving commuters to facilitate the

subsidy for commuters who choose to share a ride.

Similarly, adaptive reward strategies need to be designed for crowd-shipping systems such as those described in Chapters 5 and 6. For this, inspiration can be taken from the predictive and adaptive pricing in a car-sharing system, as described in Chapter 2. These adaptive reward strategies can incorporate the relationship between supply and demand. When the number of potential crowd-shippers is high but the number of parcels that need to be delivered is low, the crowd-shippers compete for delivering the same parcel and therefore rewards can be lower. On the other hand, when the number of potential crowd-shippers is lower than the number of parcels, crowd-shippers need to be convinced to make the delivery by increasing the rewards.

The methods described in Chapter 4 can be extended to incorporate the evening commute, in addition to the morning commute. By jointly modeling the modal choice in the morning and evening commute, constraints on the availability of cars, or the requirement to go back to the transfer point where the car was parked in the morning can be incorporated. With this, the problem described in Chapter 4 needs to be extended from a two-stage stochastic programming problem to a four-stage stochastic programming problem. Although this problem is computationally more difficult, its importance is emphasized by the reliance of commuters on the ride-sharing platform or public transport during the evening commute if they choose to leave their car at home during the morning commute. In order to convince commuters to leave their car at home, the platform should be able to guarantee a return trip.

The matching approaches in Chapters 4 and 6 have focused on exact approaches for finding the optimal matching. For these problems to be applicable to realistic large-scale networks, efficient heuristic approaches need to be designed that can find high-quality matchings in a relatively short amount of time. In addition to this, we focused on static matching problems. Both of these problems can be extended to a dynamic setting, where information on drivers, riders, crowd-shippers, and parcels is collected gradually throughout the day, rather than all at once.

Finally, field experiments should be performed to evaluate the performance of the developed methods in real-life settings. In these experiments, issues like privacy, security, trust, and safety have to be addressed. In passenger transport and mobility systems, where strangers share the same vehicle during their commute, trust and safety are important concerns that need to be addressed for these systems to become an attractive alternative transport mode. In crowd-shipping systems, similar security

and trust issues need to be addressed. On the other hand, privacy issues with respect to data collection and safety guarantees arise in these online platforms.

Appendices



Proofs for theorems in Chapter 2

For notational convenience we denote the optimization problem as follows:

$$f_i(\Delta_{cost}(i)) = \max_{\Delta_{cost}(i) \geq 0} g_i(\Delta_{cost}(i)) \quad (\text{A.1})$$

In addition to this, we substitute $\Delta_{cost}(i)$ by x and $w \cdot ODL(i)$ by K . This reduces the function to be optimized to the following:

$$g(x) = P(x)(K - x) \quad (\text{A.2})$$

Theorem 1. *If for a given incentive i a profitable discount value $\Delta_{cost}(i)$ exists, there exists a unique most profitable (optimal) discount value $\Delta_{cost}^*(i)$ for which the derivative of the subproblem is equal to 0.*

Proof. We first note that the optimal discount value should lie somewhere on the interval $[0, +\infty)$, as any value below 0 violates the definition of an incentive. Also, we note that as $g(0) \geq 0$ and $g(x) < 0$ for each $x > K$, such that we can reduce the interval to the closed and bounded interval $[0, K]$. Additionally, we know that $g(K) = 0$.

A global optimum for function g on a closed and bounded interval can occur either on the boundary points, a non-differentiable point or a stationary point. As this function is differentiable on the defined interval, only the boundary points and stationary points need to be identified.

Using Fermat's theorem, a first order stationary point requires for the derivative $\frac{dg(x)}{dx} = 0$
 $\frac{dg(x)}{dx} = \frac{dP(x)}{dx}(K - x) - P(x) = \beta P(x)P(-x)(K - x) - P(x)$ where we use the fact that $\frac{dP(x)}{dx} = \beta P(x)P(-x)$

By rearranging the terms, we obtain the following requirement: $\frac{dg(x)}{dx} = P(x)[\beta P(-x)(K -$

$x) - 1] = 0$. As $0 < P(x) < 1$ by definition, we can reduce this to: $\beta P(-x)(K - x) = 1$.

By further rearranging the terms and substituting $y = -x$, it should hold that $P(y)(K + y) = \frac{1}{\beta}$.

As $P(y)(K + y)$ is strictly increasing in y there exists at most one stationary point to which we refer as x^* .

This suggests that, using Weierstrass extreme value theorem, if a profitable incentive value exists, i.e. there exists some $x \geq 0$ for which $g(x) > 0$, there exists an $x^* \geq 0$ which is the unique optimum. \square

Theorem 2. *The optimal discount $\Delta_{cost}^*(i)$ is non-decreasing in the value of $ODL(i)$*

Proof. Using the changed notation, the theorem follows directly from the proof that x^* is increasing in K . We consider two incentives i and j for which $K_i < K_j$ and all other variables are equal. The corresponding optimal discounts are x_i^* and x_j^* respectively. We distinguish between the optimal discount being at a boundary point 0 or at a stationary point. Note that we ignore the boundary point at K as this incentive will not be offered. Therefore, we consider the following three cases:

- (i) x_i^* and x_j^* are both at a stationary point

Given the first order necessary condition derived in Theorem 1, it holds that $P(-x_i^*)K_i - P(-x_i^*)x_i^* = \frac{1}{\beta}$.

Rewriting this equation in terms of K_i yields: $K_i = \frac{\frac{1}{\beta} + P(-x_i^*)x_i^*}{P(-x_i^*)}$ (and similar for K_j).

Given $K_i < K_j$, it follows that $K_j - K_i > 0$ which after some rewriting implies that $\frac{P(x_j^*) - P(x_i^*)}{\beta} + P(-x_i^*)P(-x_j^*)(x_j^* - x_i^*) > 0$ which can only hold if $x_j^* \geq x_i^*$.

- (ii) $x_i^* = 0$ (i.e. i at a boundary point)

By definition, $x_j^* \geq 0$, so $x_j^* \geq x_i^*$

- (iii) $x_j^* = 0$ (i.e. j at a boundary point)

If the discount is optimal at the boundary point, the following relationship must hold: $\max_x P(x)(K_j - x) \leq P(0)K_j$.

Consider specifically $K_i = K_j - \tau$ with $\tau > 0$.

$\max_x P(x)(K_i - x) = \max_x P(x)(K_j - x - \tau) = \max_x P(x)(K_j - x) - \tau P(x) \leq \max_x P(x)(K_j - x) - \tau P(0) \leq P(0)K_j - \tau P(0) = P(0)(K_j - \tau) = P(0)K_i$.

As $\max_x P(x)(K_i - x) \leq P(0)K_i \rightarrow x_i^* = 0 = x_j^*$

We note that in each of these three cases it holds that $x_j^* \geq x_i^*$. As the only variable change is $K_j > K_i$, it follows that x is non-decreasing in K and therefore the optimal discount is non-decreasing in the ODL value. \square

B

Proofs for theorems in Chapter 3

For the sake of completeness, we repeat here the two assumptions that are needed for some of the following theorems.

Assumption A1 (Simple matching): *The (additional) cost of matching driver i to passenger j depends only on the detour driver i makes to pickup passenger j , as defined in Equation (3.1).*

Assumption A2 (Sorting): *Without loss of generality, all passengers and drivers are sorted from left to right based on their location on the Hotelling line.*

Theorem 3. *Let the number of drivers be equal to the number of passengers. Under Assumption A1 and given that at optimality all drivers and passengers match, matching the i^{th} driver and the i^{th} passenger is always among the set of optimal matchings.*

Proof. Using drivers at location x_i and x_j and passengers at location y_k and y_l where without loss of generality $x_i \leq x_j$ and $y_k \leq y_l$. Furthermore we assume $\alpha = 1$ without loss of generality. We show that the following inequality always holds:

$$\mathcal{C}(i, k) + \mathcal{C}(j, l) \leq \mathcal{C}(i, l) + \mathcal{C}(j, k). \quad (\text{B.1})$$

To prove this, we consider six distinct scenarios that together comprise all possible realizations.

$x_i \leq x_j \leq y_k \leq y_l :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 0 + 0 \leq 0 + 0 = \mathcal{C}(i, l) + \mathcal{C}(j, k)$
$x_i \leq y_k \leq x_j \leq y_l :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 0 + 0 \leq 0 + 2(x_j - y_k) = \mathcal{C}(i, l) + \mathcal{C}(j, k)$
$x_i \leq y_k \leq y_l \leq x_j :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 0 + 2(x_j - y_l) \leq 0 + 2(x_j - y_k) = \mathcal{C}(i, l) + \mathcal{C}(j, k)$
$y_k \leq x_i \leq x_j \leq y_l :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 2(x_i - y_k) + 0 \leq 0 + 2(x_j - y_k) = \mathcal{C}(i, l) + \mathcal{C}(j, k)$
$y_k \leq y_l \leq x_i \leq x_j :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 2(x_i - y_k) + 2(x_j - y_l) \leq 0 + 2(x_j - y_k) = \mathcal{C}(i, l) + \mathcal{C}(j, k)$
$y_k \leq x_i \leq y_l \leq x_j :$	$\mathcal{C}(i, k) + \mathcal{C}(j, l) = 2(x_i - y_k) + 2(x_j - y_l) \leq 2(x_i - y_l) + 2(x_j - y_k) = \mathcal{C}(i, l) + \mathcal{C}(j, k)$

This shows that for every pair of matches, it is always better to match them in order. Following this, we can start from any matching, iteratively select a pair of matches that are not matched in order and interchange them. According to (B.1) this will never increase the objective value. This will always lead to the matching where the i^{th} driver is matched to the i^{th} passenger which is therefore at least as good as any other matching. Specifically, the matching where the i^{th} driver is matched to the i^{th} passenger is at least as good as the optimal matching. As we consider that matching all individuals is optimal, we can disregard the option to match to dummies. Therefore, the matching where the i^{th} driver is matched to the i^{th} passenger is always among the set of optimal matchings. \square

Theorem 4. *Let the number of drivers be equal to the number of passengers and let matching cost be defined according to assumption A1. Assume that at optimality k drivers and k passengers are not matched. The matching where the k left-most passengers remain unmatched, the k right-most drivers remain unmatched and the remaining passengers and drivers are matched in sequence according to Theorem 2, is always among the set of optimal matches.*

Proof. Given that k drivers and k passengers remain unmatched in the optimal matching, there exist k matches of drivers to dummy passengers and k matches of passengers to dummy drivers.

Given any match of a driver to a dummy passenger, we can proof that the matching can always be improved by interchanging this driver with another driver to the right of this driver (see proof of Theorem 4.1).

Given any match of a passenger to a dummy driver, we can proof that the matching can always be improved by interchanging this passenger with another passenger to the left of this passenger (see proof of Theorem 5).

Applying this intuition to all matches iteratively, we show that it is optimal that the k right-most drivers are matched to dummies and the k left-most passengers are matched to dummies.

We can then disregard these individuals and on the remaining sets of passengers and drivers, apply Theorem 3. It follows that the remaining passengers and drivers are matched based on their ordered sequence under Assumption A1. \square

Theorem 5. *Consider a set of m drivers and n passengers both ranked from left to right. Under assumption A1, the following match is optimal:*

1. *If $m < n$, the $n - m$ left-most passengers are not matched (i.e. matched to dummy drivers).*
2. *If $m > n$, the $m - n$ right-most drivers are not matched (i.e. matched to dummy passengers).*
3. *The remaining $\min(m, n)$ drivers and passengers are matched according to Theorem 3.*

Proof. (Case I) We consider a driver i_2 and another driver i_1 located before driver i_2 . Thereby j is a passenger and d is a dummy passenger. By definition, $x_{i_1} \leq x_{i_2}$. Furthermore we assume $\alpha = 1$ without loss of generality. We show that the following inequality holds:

$$\mathcal{C}(i_1, d) + \mathcal{C}(i_2, j) \geq \mathcal{C}(i_2, d) + \mathcal{C}(i_1, j) \quad (\text{B.2})$$

To prove this, we consider three distinct scenarios that together comprise all possible realizations.

$$\begin{aligned} x_{i_1} \leq x_{i_2} \leq y_j : \quad & \mathcal{C}(i_1, d) + \mathcal{C}(i_2, j) = b + 0 \geq b + 0 = \mathcal{C}(i_2, d) + \mathcal{C}(i_1, j) \\ x_{i_1} \leq y_j \leq x_{i_2} : \quad & \mathcal{C}(i_1, d) + \mathcal{C}(i_2, j) = b + 2(x_{i_2} - y_j) \geq b + 0 = \mathcal{C}(i_2, d) + \mathcal{C}(i_1, j) \\ y_j \leq x_{i_1} \leq x_{i_2} : \quad & \mathcal{C}(i_1, d) + \mathcal{C}(i_2, j) = b + 2(x_{i_2} - y_j) \geq b + 2(x_{i_1} - y_j) = \mathcal{C}(i_2, d) + \mathcal{C}(i_1, j) \end{aligned}$$

This shows that for any matching, the matching where a later driver is assigned a dummy passenger is always an improvement. This can be applied iteratively until the last $m - n$ drivers in the sequence are assigned dummy passengers. Therefore, it is optimal to assign dummy passengers to the last $m - n$ drivers.

Next, we remove those last $m - n$ drivers from the considered set of drivers. We are therefore left with n drivers and n passengers. Those drivers and passenger are matched optimally as described by Theorem 4. \square

Proof. (Case II) We consider a passenger j_1 and another passenger j_2 located after passenger j_1 . Thereby i is a driver and d is a dummy driver. By definition, $y_{j_1} \leq y_{j_2}$. Furthermore we assume $\alpha = 1$ without loss of generality. We show that the following inequality holds:

$$\mathcal{C}(i, j_1) + \mathcal{C}(d, j_2) \geq \mathcal{C}(i, j_2) + \mathcal{C}(d, j_1) \quad (\text{B.3})$$

To prove this, we consider three distinct scenarios that together comprise all possible realizations.

$$\begin{aligned} x_i \leq y_{j_1} \leq y_{j_2} : \quad & \mathcal{C}(i, j_1) + \mathcal{C}(d, j_2) = 0 + c^A \geq 0 + c^A = \mathcal{C}(i, j_2) + \mathcal{C}(d, j_1) \\ y_{j_1} \leq x_i \leq y_{j_2} : \quad & \mathcal{C}(i, j_1) + \mathcal{C}(d, j_2) = 2(x_i - y_{j_1}) + c^A \geq 0 + c^A = \mathcal{C}(i, j_2) + \mathcal{C}(d, j_1) \\ y_{j_1} \leq y_{j_2} \leq x_i : \quad & \mathcal{C}(i, j_1) + \mathcal{C}(d, j_2) = 2(x_i - y_{j_1}) + c^A \geq 2(x_i - y_{j_2}) + c^A = \mathcal{C}(i, j_2) + \mathcal{C}(d, j_1) \end{aligned}$$

This shows that for any matching, the matching where an earlier passenger is assigned a dummy driver is always an improvement. This can be applied iteratively until the first $n - m$ passengers in the sequence are assigned dummy drivers. Therefore, it is optimal to assign dummy drivers to the first $n - m$ passengers.

Next, we remove those first $n - m$ drivers from the considered set of passengers. We are therefore left with m drivers and m passengers. Those drivers and passengers are matched optimally as described by Theorem 4. \square

Theorem 6. Consider linear scheduling delay with $\gamma \geq \beta$. Driver i is matched to passenger j with desired arrival times τ_i and τ_j . The optimal arrival time t^* that minimizes the total earliness and lateness penalty as given in Equation (3.4) is given as $t^* = \min(\tau_i, \tau_j)$. In this case, the reduced-form cost of matching driver i to passenger j is $\tilde{\mathcal{C}}(i, j) = 2\alpha \max(x_i - y_j, 0) + \beta|\tau_j - \tau_i|$.

Proof. Following Theorem 5.2, $t^* \in [\min(\tau_i, \tau_j), \max(\tau_i, \tau_j)]$. As we consider a closed bounded interval, we can use Weierstrass theorem to prove that $t^* = \min(\tau_i, \tau_j)$. For this, we consider the boundary points, all points that are non-differentiable and all points where the gradient is equal to zero. As the gradient is never equal to zero on the chosen interval, we consider $t = \tau_i$ and $t = \tau_j$.

Without loss of generality we assume $\tau_i \leq \tau_j$. Then,

$$E(t) + L(t) = \begin{cases} \beta(\tau_j - \tau_i) & \text{for } t = \tau_i \\ \gamma(\tau_j - \tau_i) & \text{for } t = \tau_j \end{cases} \quad (\text{B.4})$$

Given that $\gamma \geq \beta$, the minimum is obtained at $t^* = \min(\tau_i, \tau_j)$. Filling this into Equation (3.4) yields $\tilde{\mathcal{C}}(i, j) = 2\alpha \max(x_i - y_j, 0) + \beta|\tau_j - \tau_i|$. \square

Theorem 7. Consider scheduling delay penalties where earliness and lateness are non-increasing and non-decreasing functions of time, respectively. Driver i is matched to passenger j with desired arrival times τ_i and τ_j and $\tau_i \leq \tau_j$, without loss of generality. The optimal arrival time t^* that minimizes the total earliness and lateness is in the closed bounded interval $[\tau_i, \tau_j]$.

Proof. Let earliness and lateness for individual i be defined as $E_i(t)$ and $L_i(t)$ respectively. Let the total earliness and total lateness for arrival time t be denoted by $E(t)$ and $L(t)$ respectively. That is, $E(t) = E_i(t) + E_j(t)$ and $L(t) = L_i(t) + L_j(t)$ for a matched couple i and j . As the non-increasing and non-decreasing properties are additive, $E(t)$ is non-increasing in t and $L(t)$ is non-decreasing in t . Therefore, the following inequalities hold:

$$\begin{aligned} E(t) &\leq E(t') \text{ for } t \geq t', \\ L(t) &\geq L(t') \text{ for } t \geq t'. \end{aligned}$$

Furthermore, given the truncated nature of earliness and lateness (i.e. if both passenger and driver are early (late), lateness (earliness) is strictly 0.), the following equalities hold:

$$\begin{aligned} E(t) &= 0 \text{ for } t \geq \tau_j, \\ L(t) &= 0 \text{ for } t \leq \tau_i. \end{aligned}$$

Consider an arrival time $t < \tau_i$. Combining the statements above, it follows that:

$$E(t) + L(t) \geq E(\tau_i) + L(\tau_i)$$

hence, $t < \tau_i$ is never better than τ_i . Similarly, consider an arrival time $t > \tau_j$, it follows that:

$$E(t) + L(t) \geq E(\tau_j) + L(\tau_j)$$

hence, $t > \tau_j$ is never better than τ_j . It follows that the optimal arrival time t^* lies within $[\tau_i, \tau_j]$. \square

C

Cost formulations and proofs for theorems in Chapter 4

C.1 Cost formulations

In this appendix, we describe the deterministic costs for every type of path. We consider separately all 10 types of paths (3 direct and 7 indirect). For the sake of notation, we define $t_j^*(h)$ as the desired arrival time of driver j at transfer hub h if he travels through that hub. This is simply computed as $t_j^*(h) = t_j^* - tt(h, d_j)$. The three direct paths are described below:

Direct PT

Every rider $i \in I$ has the option to take public transport. Public transport has a fixed cost plus a variable term per unit of time traveled. A rider $i \in I$ that takes public transport incurs a cost:

$$c_k = \alpha^{\text{pt}} tt(o_i, d_i) + \phi^{\text{pt}} \quad (\text{C.1})$$

Direct SD

Every rider that owns a car $i \in I^c$ also has the option to drive from origin to destination directly. In that case, on top of his value of time, drivers pay for fuel consumption and parking at the destination. Departure time choices are made to minimize the costs. In the case of deterministic travel times, this means that they arrive exactly at their desired arrival time, and as such schedule delay penalties are zero. In the case of stochastic travel times, this may not be the case and

departure times are chosen by modeling departures at $t \in T$ as separate paths. The deterministic costs for such a path are defined as:

$$c_k = (\alpha^{\text{car}} + \phi^{\text{fuel}})tt(o_i, d_i) + \phi_{d_i}^{\text{park}} \quad (\text{C.2})$$

Direct RS

For every rider $i \in I$, a direct match can be found with a driver $j \in J$ if $o_i = o_j$ and $d_i = d_j$. As the driver selects the departure time to minimize his/her own cost, the arrival time at the final destination is equal to his/her desired arrival time of the driver, possibly imposing schedule delay costs on the rider. A rider is penalized for earliness by β and for lateness by γ . The notation $(\cdot)^+ = \max(0, \cdot)$, which means that either earliness or lateness is positive, but not both at the same time. Only if $t_i^* = t_j^*$, the rider arrives exactly on time, and therefore both earliness and lateness will be zero. For a match between $i \in I$ and $j \in J$, $e_{ik} = 1$, $a_{jk}^0 = 1$ and all other parameters are equal to 0. The cost of this direct match is as follows:

$$c_k = \alpha^{\text{car}}tt(o_i, d_i) + \beta(t_i^* - t_j^*)^+ + \gamma(t_j^* - t_i^*)^+ \quad (\text{C.3})$$

The seven indirect paths are described below:

Indirect RS \rightarrow RS

We consider a rider $i \in I$ and two drivers $j_1, j_2 \in J$ where j_1 takes i on the first leg and j_2 takes i on the second leg with a transfer at transfer hub h . Similar to before, this is only feasible if $o_i = o_{j_1}$, $d_i = d_{j_2}$. Thereby, $t_{j_1}^*(h) \leq t_{j_2}^*(h)$ to ensure that the rider is dropped off at the transfer hub before the scheduled pickup. The hub h needs to deviate at most τ minutes from the shortest path of both drivers j_1 and j_2 . In this case, $a_{j_1k}^{1h} = 1$ and $a_{j_2k}^{2h} = 1$. The cost for the rider i is then defined as follows:

$$\begin{aligned} c_k = & \alpha^{\text{car}}[tt(o_i, h) + tt(h, d_i)] + \alpha^{\text{wait}}[t_{j_2}^*(h) - t_{j_1}^*(h)] \\ & + \beta[t_i^* - t_{j_2}^*(h) - tt(h, d_i)]^+ + \gamma[t_{j_2}^*(h) + tt(h, d_i) - t_i^*]^+ \end{aligned} \quad (\text{C.4})$$

Indirect RS \rightarrow PT

For a path where only the first leg is a ride-sharing leg, a rider knows in advance when he will be picked up at the transfer hub and can therefore adjust his departure time on the first leg to the departure on the second leg. If rider $i \in I$ and driver $j \in J$ are matched with a transfer at hub $h \in H$, they must share their origin $o_i = o_j$ and driver j may deviate at most τ minutes from his/her shortest path to reach hub h . In this case, $a_{j_1k}^{1h} = 1$. The cost of this match is:

$$c_k = \alpha^{\text{car}}tt(o_i, h) + \alpha^{\text{pt}}tt(h, d_i) + \phi^{\text{pt}} + \beta[t_i^* - t_j^*(h) - tt(h, d_i)]^+ + \gamma[t_j^*(h) + tt(h, d_i) - t_i^*]^+ \quad (\text{C.5})$$

Indirect PT \rightarrow RS

We consider the path where the rider shares a ride on the second leg and takes public transport on the first leg. If rider $i \in I$ and driver $j \in J$ are matched with a transfer at hub $h \in H$, they must share their destination $d_i = d_j$ and driver j may deviate at most τ minutes from his/her shortest path to reach hub h . In this case, $a_{j2k}^{2h} = 1$. The cost of this match is:

$$c_k = \alpha^{\text{pt}} tt(o_i, h) + \phi^{\text{pt}} + \alpha^{\text{car}} tt(h, d_i) + \beta[t_i^* - t_j^*(h) - tt(h, d_i)]^+ + \gamma[t_j^*(h) + tt(h, d_i) - t_i^*]^+ \quad (\text{C.6})$$

Indirect SD \rightarrow RS

We consider an indirect path where a rider drives alone on the first leg and shares a ride on the second leg. We note that the arrival time at the transfer hub should be coordinated, similar to an indirect ride-sharing match. Let rider $i \in I$ be matched to driver $j \in J$ on the second leg and let the departure time of rider i on the first leg be equal to $t \in T$. The arrival time of the rider at the transfer hub $h \in H$ is then equal to $t + tt(o_i, h)$. In a deterministic setting, the rider can optimize their departure time t to arrive exactly on time at the transfer point and therefore incur no waiting time. In a stochastic setting, this is not necessarily the case. Again, driver j may deviate at most τ minutes from his/her shortest path to reach hub h . In this case, $a_{j2k}^{2h} = 1$. The cost is defined as follows:

$$c_k = (\alpha^{\text{car}} + \phi^{\text{fuel}}) tt(o_i, h) + \alpha^{\text{car}} tt(h, d_i) + \alpha^{\text{wait}} [t_j^*(h) - t - tt(o_i, h)] + \beta[t_i^* - t_j^*(h) - tt(h, d_i)]^+ + \gamma[t_j^*(h) + tt(h, d_i) - t_i^*]^+ \quad (\text{C.7})$$

Indirect SD \rightarrow PT

In case a rider $i \in I$ drives their own car on the first leg, departs at time $t \in T$, and transfers to public transport at hub $h \in H$, the cost is defined as follows:

$$c_k = (\alpha^{\text{car}} + \phi^{\text{fuel}}) tt(o_i, h) + \alpha^{\text{pt}} tt(h, d_i) + \beta[t_i^* - t - tt(o_i, h) - tt(h, d_i)]^+ + \gamma[t + tt(o_i, h) + tt(h, d_i) - t_i^*]^+ \quad (\text{C.8})$$

Indirect PT \rightarrow PT and SD \rightarrow SD

For completeness, we introduce two indirect alternatives where the same mode is used on both legs. Clearly, in the deterministic case, a direct option with that same mode would always perform at least as well. However, in the stochastic case, an indirect option (i.e., traveling through the transfer hub without changing driver or mode) may be used in some scenarios to allow for multi-modal paths in other

scenarios. The cost for an indirect public transport path and an indirect solo drive path for individual i with a transfer at hub h are given as follows:

$$c_k = \alpha^{\text{pt}}(tt(o_i, h) + tt(h, d_i)) + 2\phi^{\text{pt}} \quad (\text{C.9})$$

$$c_k = (\alpha^{\text{car}} + \phi^{\text{fuel}})(tt(o_i, h) + tt(h, d_i)) + \phi_{d_i}^{\text{park}} \quad (\text{C.10})$$

C.2 Theorems and proofs

C.2.1 Optimality propositions

Proposition 3 (Stability of optimal matching). *Suppose all riders are aware of their alternative modes of transport but are only aware of their current match and not of other available drivers. Then, the optimal matching is stable (i.e. no rider can improve their costs by changing their mode of transport).*

Proposition 4 (Strictly dominated paths - deterministic). *Consider a rider $i \in I$ and a path $k_1 \in K$. Let path $k_2 \in K$ of rider i be a copy of path k_1 where one or multiple legs are replaced by direct or indirect legs where the rider travels alone (either by car or public transport). If $c_{k_2} < c_{k_1}$, then path k_1 is strictly dominated by path k_2 and can therefore be omitted from K .*

Remark 1 (Strictly dominated paths - stochastic). *In case travel times are stochastic, Proposition 4 does not hold. Consider the following counter-example: Consider a rider $i \in I$ and three paths $k_1, k_2, k_3 \in K$ corresponding to this rider. Consider $|\Omega| = 3$ where every scenario has equal probability. Let path k_1 be a direct public transport path for which $c_{k_1}(\omega_1) = c_{k_1}(\omega_2) = c_{k_1}(\omega_3) = 10$. Let path k_2 be an indirect public transport path through hub h for which $c_{k_2}(\omega_1) = c_{k_2}(\omega_2) = c_{k_2}(\omega_3) = 12$. Let path k_3 be an indirect public transport-carpool path, also through hub h , for which $c_{k_3}(\omega_1) = 6$, $c_{k_3}(\omega_2) = 10$, and $c_{k_3}(\omega_3) = 14$.*

Even though $c_{k_1}(\omega) < c_{k_2}(\omega)$ for every $\omega \in \Omega$, path k_2 cannot be omitted. The reason for this is that k_1 and k_3 have identical first legs and can therefore be combined across scenarios. That is, by choosing a first-leg public transport path in every scenario but choosing to carpool on the second leg in ω_1 and ω_2 and choosing public transport on the second leg in ω_3 , the expected cost can be minimized and this combined path forms an improvement over path k_1 (9.33 versus 10). This disproves Proposition 4 for stochastic travel times.

Proposition 5 (Strictly dominated paths - stochastic). *Let $c_i^{\min}(\omega)$ be the minimal cost for rider $i \in I$ to travel between their origin and destination in scenario $\omega \in \Omega$. If for a path $k_1, k_2 \in K$ it holds that $c_{k_1}(\omega_1) + \sum_{\omega_2 \in \Omega \setminus \omega_1} c_i^{\min}(\omega_2) > \sum_{\omega_2 \in \Omega} c_{k_2}(\omega_2)$ for all $\omega_1 \in \Omega_1$, then path k_1 is strictly dominated by path k_2 and can therefore be omitted from K .*

C.2.2 Departure time choice proofs

Theorem 9 (Optimal departure time for a single leg). *Let all riders and the driver have identical origins and destinations. Let a driver with desired arrival time t_0^* be matched to N riders with desired arrival times $t_1^* \dots t_N^*$. With $\max(\beta_0, \beta_1, \dots, \beta_N) < \min(\gamma_0, \gamma_1, \dots, \gamma_N)$, the jointly optimal departure time is equal to $t^o = \min(t_0^*, \dots, t_N^*)$.*

Proof. Without loss of generality, travel time is set equal to 0. Let t^o be the jointly optimal departure time and let $C(t)$ be the joint total cost for all drivers and passengers. Then, $t^o = \arg \min_t C(t)$. Without loss of generality, we sort the desired arrival times such that $t_0^* \leq t_1^* \leq \dots \leq t_N^*$. The cost $C(t)$ is then defined as follows:

$$C(t) = \sum_{i \in \{0, \dots, N\} | t_i^* < t} \beta_i(t - t_i^*) + \sum_{i \in \{0, \dots, N\} | t_i^* \geq t} \gamma_i(t_i^* - t) \quad (\text{C.11})$$

The function $C(t)$ is piece-wise linear and therefore the optimal departure time t^o has to be at one of the breakpoints $\{t_0^*, \dots, t_N^*\}$. A graphic example is displayed in Figure C.1. Given that $\max(\beta_0, \beta_1, \dots, \beta_N) < \min(\gamma_0, \gamma_1, \dots, \gamma_N)$, it can be shown by contradiction that $t^o = \min(t_0^*, \dots, t_N^*)$. \square

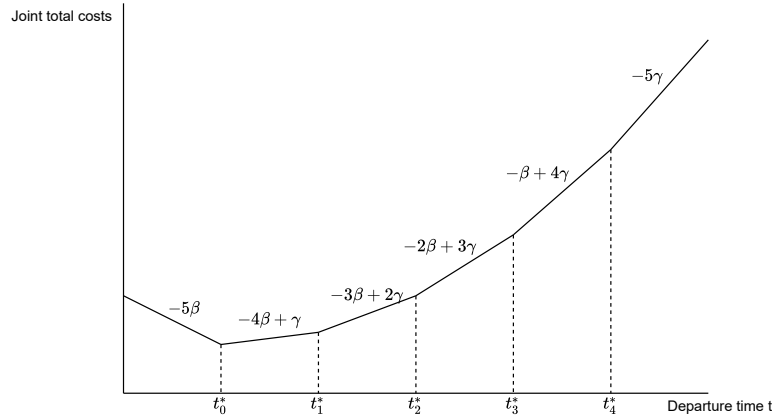


Figure C.1: Theorem 1 example with $N = 4$ riders and homogeneous β and γ

Theorem 10 (Optimal departure time for a second leg trip). *Consider N riders k_1, \dots, k_N from origins o_{k_1}, \dots, o_{k_N} who transfer at hub h to their identical destination d , and a driver i from hub h to the same destination d , with their desired arrival times $t_{k_1}^*, \dots, t_{k_N}^*$ and t_i^* . Let all individuals have identical cost parameters $\alpha, \alpha^{\text{wait}}, \beta, \gamma$, with $\beta < \gamma$ and $\alpha^{\text{wait}} < \gamma$. We let t_1 be the last departure time for the first leg among all riders and the driver. The joint optimal departure time for the second leg t_2^o is a function of the departure time for the first leg t_1 which is defined as follows:*

$$t_2^o(t_1) = \begin{cases} \max(t_1, \min(t_i^*, t_{k_1}^*, \dots, t_{k_N}^*)) & \text{if } \alpha^{\text{wait}} \leq \beta \\ t_1 & \text{if } \alpha^{\text{wait}} > \beta \end{cases} \quad (\text{C.12})$$

Proof. Without loss of generality, travel time is equal to 0. Let $t_2^o(t_1)$ be the jointly optimal departure time on the second leg given the latest departure time t_1 on the first leg and let $C_2(t_1, t_2)$ be the joint total cost on the second leg for all drivers and passengers where t_1 is the latest departure time on the first leg and t_2 is the departure time on the first leg. Then, $t_2^o(t_1) = \arg \min_{t_2} C(t_1, t_2)$. Clearly, leaving before the last passenger has arrived makes the match infeasible and therefore attains a cost of ∞ . This implies that $t_2^o(t_1) \geq t_1$. Therefore, in the remainder of this proof, we disregard the period before t_1 and the costs of waiting during that period.

Without loss of generality, we sort the desired arrival times such that $t_{k_1}^* \leq \dots \leq t_{k_N}^*$. The cost $C_2(t_1, t_2)$ is then defined as follows:

$$C_2(t_1, t_2) = (N+1)\alpha^{\text{wait}}(t_2 - t_1) + \sum_{k \in \{i, k_0, \dots, k_N\} | t_k^* < t} \beta(t - t_k^*) + \sum_{k \in \{i, k_0, \dots, k_N\} | t_k^* \geq t} \gamma_i(t_k^* - t) \quad (\text{C.13})$$

Using the same reasoning as in Theorem 9, earliness is jointly preferred over lateness. In this case, we have an additional trade-off between earliness and waiting time. If waiting time is penalized more than earliness, it is best to leave immediately after everyone has arrived such that $t_2^o(t_1) = t_1$. Otherwise, it might be better to wait. We separately consider the cases where (i) $t_1 > \min(t_i^*, t_{k_1}^*, \dots, t_{k_N}^*)$ and (ii) $t_1 \leq \min(t_i^*, t_{k_1}^*, \dots, t_{k_N}^*)$. This is graphically depicted in Figure C.2. In case (i), at least one matched individual is already late and therefore waiting longer is definitely not desirable as $\alpha^{\text{wait}} < \gamma$. In that case, $t_2^o(t_1) = t_1$. In case (ii), it is better to wait at the transfer hub until $t_2^o(t_1) = \min(t_i^*, t_{k_1}^*, \dots, t_{k_N}^*)$, applying the reasoning from Theorem 9. Combining these individual cases leads to the optimal departure time on the second leg as given in Equation (C.12). \square

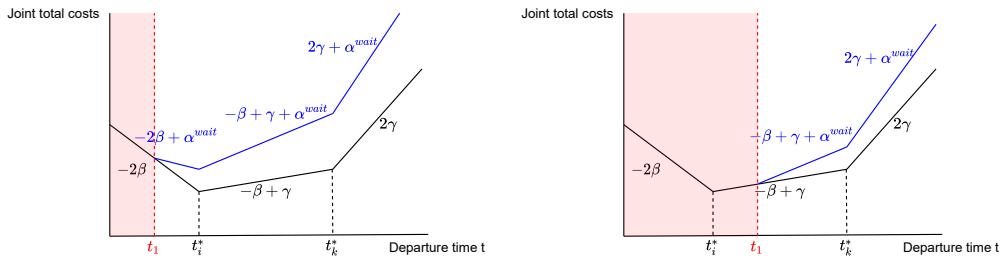


Figure C.2: Schedule delay and waiting time

Theorem 11 (Optimal departure time on first leg). *Consider a driver i from o to d passing through hub h , one rider j from o to hub h and one rider k from hub h to d , with their desired arrival times t_i^*, t_j^*, t_k^* . Let all individuals have identical cost parameters $\alpha, \alpha^{\text{wait}}, \beta, \gamma$, with $\beta < \alpha^{\text{wait}} < \gamma$. The joint optimal departure time for*

the first leg $t_1^o = \min(t_i^*, t_j^*, t_k^*)$. The joint optimal departure time of the second leg can then be determined according to Theorem 10.

Proof. Without loss of generality, travel time is set equal to 0. Let $t_2^o(t_1)$ be the jointly optimal departure time on the second leg given the departure time t_1 on the first leg and let $C(t_1, t_2)$ be the joint cost of the driver and the two riders on both legs. Because $\beta < \alpha^{\text{wait}}$, according to Theorem 10, $t_2^o(t_1) = t_1$ given the relationship between the parameters. We can therefore determine $C(t_1) = C(t_1, t_2^o(t_1))$ which is only composed of schedule-delay costs and not waiting, given the immediate departure from the transfer hub. By applying Theorem 9 we obtain $t_1^o = \min(t_i^*, t_j^*, t_k^*)$. \square

Remark 2 (Optimal departure time on first leg). *If $\alpha^{\text{wait}} > \beta$, Theorem 3 does not necessarily hold and the optimal departure time depends on the order of the desired arrival times as well as the ratio of α^{wait} , β and γ .*

C.2.3 Optimal departure time with stochastic travel times

Lemma 1 (Optimal departure time at breakpoints). *Consider a commuter with desired arrival time t^* and let travel time be subject to uncertain changes of factor Δ_ω (i.e., $tt = \bar{t}t \cdot \Delta_\omega$) where $\omega \in \Omega$ with a discrete distribution $p(\omega)$. The optimal departure time is chosen to minimize total schedule delay costs of the commuter. Let t_ω be the optimal departure time for individual scenario ω . The departure time that minimizes the expected schedule delay costs (t^o) should be at one of the breakpoints of the cost function.*

Proof. Let $C(t, \omega)$ be the generalized cost of departing at time t under scenario $\omega \in \Omega$. The expected cost of departing at time t is defined as $\mathbb{E}_\omega[C(t)]$. Given the discrete distribution, this can be written as $\mathbb{E}_\omega[C(t)] = \sum_{\omega \in \Omega} p(\omega)C(t, \omega)$. Given the axioms of the probability distribution, $p(\omega) \geq 0$ and therefore $\mathbb{E}_\omega[C(t)]$ is a weighted sum of the scenario-specific costs. As $C(t, \omega)$ are piecewise-linear convex functions, the weighted sum is also a piecewise-linear convex function. The breakpoints of this function are the breakpoints of all scenario-specific cost functions and are defined as t_ω , which are the optimal departure times under that scenario. The departure time t^o that minimizes $\mathbb{E}_\omega[C(t)]$ is equal to one of these breakpoints. \square

Theorem 12 (Optimal departure time of drivers with stochastic travel time). *Consider a commuter with desired arrival time t^* and let travel time be subject to uncertain changes of factor Δ_ω (i.e., $tt = \bar{t}t \cdot \Delta_\omega$) where $\omega \in \Omega$ with a discrete distribution $p(\omega)$. Let t_ω be the optimal departure time for individual scenario ω . The departure time that minimizes the expected costs is then equal to $t^o = \max\{t_\omega, \omega \in \Omega | \sum_{\omega' \in \Omega | t_{\omega'} < t_\omega} p(\omega')\beta > \sum_{\omega' \in \Omega | t_{\omega'} > t_\omega} p(\omega')\gamma\}$*

Proof. According to Lemma 1, the optimal departure time t_o should be at one of the breakpoints of the expected generalized cost function $\mathbb{E}_\omega[C(t)]$. The expected generalized cost function is defined as follows:

$$\mathbb{E}_\omega[C(t)] = \sum_{\omega \in \Omega} p(\omega) [\beta(t_\omega - t)^+ + \gamma(t - t_\omega)^+] \quad (\text{C.14})$$

$$\sum_{\omega \in \Omega | t_\omega \geq t} p(\omega) \beta(t_\omega - t) + \sum_{\omega \in \Omega | t_\omega < t} p(\omega) \gamma(t - t_\omega) \quad (\text{C.15})$$

Although the derivative of the function $\mathbb{E}_\omega[C(t)]$ is not defined at the breakpoint, it should hold that the derivative is negative before and positive after the breakpoint. The derivative at non-breakpoints is the weighted sum of the derivative of the scenario-specific cost functions. We identify the last breakpoint for which the slope before that breakpoint is negative, which can be denoted as follows:

$$t^o = \max \left\{ t_\omega, \omega \in \Omega \mid \sum_{\omega' \in \Omega | t_{\omega'} \geq t_\omega} p(\omega') \beta > \sum_{\omega' \in \Omega | t_{\omega'} < t_\omega} p(\omega') \gamma \right\}. \quad (\text{C.16})$$

We observe that we can efficiently find the optimal departure time by computing the individual breakpoints $t_\omega = t^* - \bar{t} \cdot (\Delta_\omega - 1)$ and going over them in increasing order until the inequality in Equation (C.16) is no longer satisfied. This concludes the proof. A graphic example is displayed in Figure C.3. □

Corollary 1 (Special case of Theorem 12 - uniform distribution). *Consider a distribution of Δ_ω where $p(\omega)$ is uniform such that $p(\omega) = p(\omega')$ for all $\omega, \omega' \in \Omega$. Let $t^{(\omega)}$ be the ordered sequence of optimal departure times for the individual scenarios. Then the departure time that minimizes the expected costs is equal to $t^{(n+1)}$ with $n = \left\lfloor \frac{\beta}{\gamma + \beta} |\Omega| \right\rfloor$.*

Proof. According to Theorem 12 and specifically using Equation (C.16) and the fact that $p(\omega) = p(\omega')$ for all $\omega, \omega' \in \Omega$, we are able to eliminate $p(\omega)$ from the inequality by dividing the left and right-hand side by $p(\omega) > 0$. Let n denote number of elements $\omega' \in \Omega$ such that $t_{\omega'} < t^{(n+1)}$ then we can rewrite the function of t^o as follows:

$$t^o = \max \left\{ t_\omega, \omega \in \Omega \mid \sum_{\omega' \in \Omega | t_{\omega'} \geq t_\omega} \beta > \sum_{\omega' \in \Omega | t_{\omega'} < t_\omega} \gamma \right\}. \quad (\text{C.17})$$

$$t^o = \max \left\{ t_\omega, \omega \in \Omega \mid (|\Omega| - n) \beta > n \gamma \right\}. \quad (\text{C.18})$$

$$t^o = \max \left\{ t_\omega, \omega \in \Omega \mid |\Omega| \frac{\beta}{\gamma + \beta} > n \right\}. \quad (\text{C.19})$$

$$t^o = t^{(n+1)} \text{ where } n = \left\lfloor \frac{\beta}{\gamma + \beta} |\Omega| \right\rfloor. \quad (\text{C.20})$$

This concludes the proof. A graphic example is displayed in Figure C.3. □

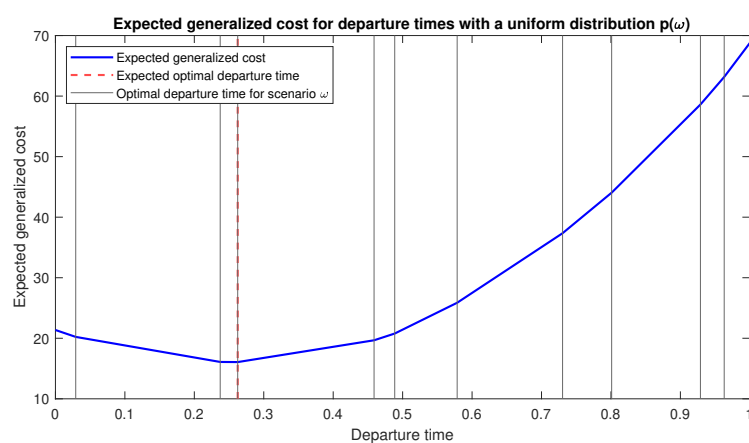


Figure C.3: Example of Corollary 1

References

- Agatz, N., Bouman, P., and Schmidt, M. (2018). “Optimization approaches for the traveling salesman problem with drone”. In: *Transportation Science* 52.4, pp. 965–981.
- Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). “Optimization for dynamic ride-sharing: A review”. In: *European Journal of Operational Research* 223.2, pp. 295–303.
- Akeb, H., Moncef, B., and Durand, B. (2018). “Building a collaborative solution in dense urban city settings to enhance parcel delivery: An effective crowd model in Paris”. In: *Transportation Research Part E: Logistics and Transportation Review* 119, pp. 223–233.
- Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment”. In: *Proceedings of the National Academy of Sciences* 114.3, pp. 462–467.
- Angelopoulos, A., Gavalas, D., Konstantopoulos, C., Kypriadis, D., and Pantziou, G. (2016). “Incentivization schemes for vehicle allocation in one-way vehicle sharing systems”. In: *2016 IEEE International Smart Cities Conference (ISC2)*. IEEE, pp. 1–7.
- Ansari, S., Başdere, M., Li, X., Ouyang, Y., and Smilowitz, K. (2018). “Advancements in continuous approximation models for logistics and transportation systems: 1996–2016”. In: *Transportation Research Part B: Methodological* 107, pp. 229–252.
- Archetti, C., Savelsbergh, M., and Speranza, M. G. (2016). “The vehicle routing problem with occasional drivers”. In: *European Journal of Operational Research* 254.2, pp. 472–480.
- Arnott, R., Palma, A. de, and Lindsey, R. (1990). “Economics of a bottleneck”. In: *Journal of urban economics* 27.1, pp. 111–130.
- Arnott, R., Palma, A. de, and Lindsey, R. (1993). “A structural model of peak-period congestion: A traffic bottleneck with elastic demand”. In: *The American Economic Review*, pp. 161–179.
- Arslan, A. M., Agatz, N., Kroon, L., and Zuidwijk, R. (2019). “Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers”. In: *Transportation Science* 53.1, pp. 222–235.
- Bahat, O. and Bekhor, S. (2016). “Incorporating ridesharing in the static traffic assignment model”. In: *Networks and Spatial Economics* 16.4, pp. 1125–1149.
- Bai, R., Li, J., Atkin, J. A., and Kendall, G. (2014). “A novel approach to independent taxi scheduling problem based on stable matching”. In: *Journal of the Operational Research Society* 65.10, pp. 1501–1510.

- Baptista, P., Melo, S., and Rolim, C. (2014). “Energy, environmental and mobility impacts of car-sharing systems. Empirical results from Lisbon, Portugal”. In: *Procedia-Social and Behavioral Sciences* 111, pp. 28–37.
- Bento, A. M., Hughes, J. E., and Kaffine, D. (2013). “Carpooling and driver responses to fuel price changes: Evidence from traffic flows in Los Angeles”. In: *Journal of Urban Economics* 77, pp. 41–56.
- Beojone, C. V. and Geroliminis, N. (2021). “On the inefficiency of ride-sourcing services towards urban congestion”. In: *Transportation research part C: emerging technologies* 124, p. 102890.
- BlaBlaCar (2020). *www.blablacar.com*. <https://blog.blablacar.com/about-us>.
- Blad, K., Almeida Correia, G. H. de, Nes, R. van, and Annema, J. A. (2022). “A methodology to determine suitable locations for regional shared mobility hubs”. In: *Case Studies on Transport Policy*.
- Boyacı, B., Zografos, K. G., and Geroliminis, N. (2015). “An optimization framework for the development of efficient one-way car-sharing systems”. In: *European Journal of Operational Research* 240.3, pp. 718–733.
- Brandstätter, G., Kahr, M., and Leitner, M. (2017). “Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand”. In: *Transportation Research Part B: Methodological* 104, pp. 17–35.
- Brendel, A. B., Brauer, B., and Hildebrandt, B. (2016). “Toward User-Based Relocation Information Systems in Station-Based One-Way Car Sharing”. In: *Americas Conference on Information Systems (AMCIS)*. Association For Information Systems.
- Browning, M., Chiappori, P.-A., and Weiss, Y. (2014). *Economics of the Family*. Cambridge University Press.
- Burkard, R. E. and Cela, E. (1999). “Linear assignment problems and extensions”. In: *Handbook of combinatorial optimization*. Springer, pp. 75–149.
- Caggiani, L. and Ottomanelli, M. (2013). “A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems”. In: *Procedia-Social and Behavioral Sciences* 87, pp. 203–210.
- Capital Bikeshare (2020). *Capital Bikeshare*, <https://www.capitalbikeshare.com/system-data>.
- Caulfield, B. (2009). “Estimating the environmental benefits of ride-sharing: A case study of Dublin”. In: *Transportation Research Part D: Transport and Environment* 14.7, pp. 527–531.
- Cenedese, C., Stokkink, P., Geroliminis, N., and Lygeros, J. (2022). “Incentive-based electric vehicle charging for managing bottleneck congestion”. In: *European Journal of Control* 68, p. 100697.
- Census Reporter (2021). *Census Reporter*, <https://censusreporter.org/>. Accessed: 10-03-2021.

- Chemla, D., Meunier, F., and Calvo, R. W. (2013). “Bike sharing systems: Solving the static rebalancing problem”. In: *Discrete Optimization* 10.2, pp. 120–146.
- Chen, W., Mes, M., and Schutten, M. (2018). “Multi-hop driver-parcel matching problem with time windows”. In: *Flexible Services and Manufacturing Journal* 30.3, pp. 517–553.
- Chen, Y. et al. (2019). “PPtaxi: Non-stop package delivery via multi-hop ridesharing”. In: *IEEE Transactions on Mobile Computing* 19.11, pp. 2684–2698.
- Chiappori, P.-A., Oreffice, S., and Quintana-Domeque, C. (2012). “Fatter attraction: anthropometric and socioeconomic matching on the marriage market”. In: *Journal of Political Economy* 120.4, pp. 659–695.
- Clemente, M., Fanti, M. P., Iacobellis, G., Nolich, M., and Ukovich, W. (2017). “A decision support system for user-based vehicle relocation in car sharing systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.8, pp. 1283–1296.
- Cohn, A., Root, S., Wang, A., and Mohr, D. (2007). “Integration of the load-matching and routing problem with equipment balancing for small package carriers”. In: *Transportation Science* 41.2, pp. 238–252.
- Cornuéjols, G., Nemhauser, G., and Wolsey, L. (1983). *The uncapacitated facility location problem*. Tech. rep. Cornell University Operations Research and Industrial Engineering.
- Correia, G. H. D. A., Jorge, D. R., and Antunes, D. M. (2014). “The added value of accounting for users’ flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal”. In: *Journal of Intelligent Transportation Systems* 18.3, pp. 299–308.
- Cui, T., Ouyang, Y., and Shen, Z.-J. M. (2010). “Reliable facility location design under the risk of disruptions”. In: *Operations Research* 58.4-part-1, pp. 998–1011.
- Daganzo, C. F. (2005). *Logistics systems analysis*. Springer Science & Business Media.
- Dahle, L., Andersson, H., and Christiansen, M. (2017). “The vehicle routing problem with dynamic occasional drivers”. In: *International Conference on Computational Logistics*. Springer, pp. 49–63.
- Di Febbraro, A., Sacco, N., and Saeednia, M. (2018). “One-Way Car-Sharing Profit Maximization by Means of User-Based Vehicle Relocation”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.2, pp. 628–641.
- DiPugliaPugliese, L., Guerriero, F., Macrina, G., and Scalzo, E. (2021). “Crowd-Shipping and Occasional Depots in the Last Mile Delivery”. In: *Optimization and Decision Science*. Springer, pp. 213–225.
- Drews, F. and Luxen, D. (2013). “Multi-hop ride sharing”. In: *International Symposium on Combinatorial Search*. Vol. 4. 1.
- Dumas, Y., Desrosiers, J., and Soumis, F. (1991). “The pickup and delivery problem with time windows”. In: *European Journal of Operational Research* 54.1, pp. 7–22.
- Duranton, G. and Puga, D. (2004). “Micro-foundations of urban agglomeration economies”. In: *Handbook of regional and urban economics*. Vol. 4. Elsevier, pp. 2063–2117.

- Elmachtoub, A. N. and Grigas, P. (2017). "Smart" predict, then optimize". In: *arXiv preprint arXiv:1710.08005*.
- Enthoven, D. L., Jargalsaikhan, B., Roodbergen, K. J., Broek, M. A. Uit het, and Schrotenboer, A. H. (2020). "The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers". In: *Computers & Operations Research* 118, p. 104919.
- Ermagun, A. and Stathopoulos, A. (2018). "To bid or not to bid: An empirical study of the supply determinants of crowd-shipping". In: *Transportation Research Part A: Policy and Practice* 116, pp. 468–483.
- Fabri, A. and Recht, P. (2006). "On dynamic pickup and delivery vehicle routing with several time windows and waiting times". In: *Transportation Research Part B: Methodological* 40.4, pp. 335–350.
- Fernandes, D. R. et al. (2014). "A simple and effective genetic algorithm for the two-stage capacitated facility location problem". In: *Computers & Industrial Engineering* 75, pp. 200–208.
- Firnkorn, J. and Müller, M. (2011). "What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm". In: *Ecological Economics* 70.8, pp. 1519–1528.
- Frost & Sullivan (2016). "Future of Carsharing Market to 2025". In: *Technology Advancements, Market Consolidation and Government Initiatives to Influence Market Growth Over the Next Decade*.
- Galichon, A. (2018). *Optimal transport methods in economics*. Princeton University Press.
- Gendron, B., Khuong, P.-V., and Semet, F. (2016). "A Lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints". In: *Transportation Science* 50.4, pp. 1286–1299.
- George, D. K. and Xia, C. H. (2011). "Fleet-sizing and service availability for a vehicle rental system via closed queueing networks". In: *European Journal of Operational Research* 211.1, pp. 198–207.
- Google (2019). "Google Maps". In: Accessed: 27-11-2019.
- Google Maps (2020). *Google Maps*. Accessed: 13-03-2020.
- Gurumurthy, K. M., Kockelman, K. M., and Simoni, M. D. (2019). "Benefits and costs of ride-sharing in shared automated vehicles across Austin, Texas: opportunities for congestion pricing". In: *Transportation Research Record* 2673.6, pp. 548–556.
- Heller, I, Tompkins, C., Kuhn, H. W., and Tucker, A. W. (1957). "14. An Extension of a Theorem of Dantzig's". In: *Linear Inequalities and Related Systems.(AM-38), Volume 38*. Princeton University Press, pp. 247–254.
- Herbawi, W. and Weber, M. (2011). "Evolutionary multiobjective route planning in dynamic multi-hop ridesharing". In: *European conference on evolutionary computation in combinatorial optimization*. Springer, pp. 84–95.

- Herbawi, W. and Weber, M. (2012). “The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm”. In: *2012 IEEE Congress on Evolutionary Computation*. IEEE, pp. 1–8.
- Herrmann, S., Schulte, F., and Voß, S. (2014). “Increasing acceptance of free-floating car sharing systems using smart relocation strategies: a survey based study of car2go Hamburg”. In: *International Conference on Computational Logistics*. Springer, pp. 151–162.
- Huang, H., Bucher, D., Kissling, J., Weibel, R., and Raubal, M. (2018). “Multimodal route planning with public transport and carpooling”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.9, pp. 3513–3525.
- Illgen, S. and Höck, M. (2019). “Literature review of the vehicle relocation problem in one-way car sharing networks”. In: *Transportation Research Part B: Methodological* 120, pp. 193–204.
- Iryo, T. and Yoshii, T. (2007). “Equivalent optimization problem for finding equilibrium in the bottleneck model with departure time choices”. In: *4th IMA International Conference on Mathematics in Transport Institute of Mathematics and its Applications*.
- Iwan, S., Kijewska, K., and Lemke, J. (2016). “Analysis of parcel lockers’ efficiency as the last mile delivery solution—the results of the research in Poland”. In: *Transportation Research Procedia* 12, pp. 644–655.
- Jorge, D., Molnar, G., and Almeida Correia, G. H. de (2015). “Trip pricing of one-way station-based carsharing networks with zone and time of day price variations”. In: *Transportation Research Part B: Methodological* 81, pp. 461–482.
- Kafle, N., Zou, B., and Lin, J. (2017). “Design and modeling of a crowdsourcing-enabled system for urban parcel relay and delivery”. In: *Transportation research part B: methodological* 99, pp. 62–82.
- Karak, A. and Abdelghany, K. (2019). “The hybrid vehicle-drone routing problem for pick-up and delivery services”. In: *Transportation Research Part C: Emerging Technologies* 102, pp. 427–449.
- Kek, A. G., Cheu, R. L., Meng, Q., and Fung, C. H. (2009). “A decision support system for vehicle relocation operations in carsharing systems”. In: *Transportation Research Part E: Logistics and Transportation Review* 45.1, pp. 149–158.
- Kızıl, K. U. and Yıldız, B. (2022). “Public Transport-Based Crowd-Shipping with Backup Transfers”. In: *Transportation Science*.
- Kong, H., Zhang, X., and Zhao, J. (2020). “How does ridesourcing substitute for public transit? A geospatial perspective in Chengdu, China”. In: *Journal of Transport Geography* 86, p. 102769.
- Kumar, P. and Khani, A. (2021). “An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access”. In: *Transportation Research Part C: Emerging Technologies* 122, p. 102891.
- Kumar, P. and Bierlaire, M. (2012). “Optimizing locations for a vehicle sharing system”. In: *Swiss Transport Research Conference*.

- Lan, Y.-L., Liu, F., Ng, W. W., Gui, M., and Lai, C. (2022). “Multi-Objective Two-Echelon City Dispatching Problem With Mobile Satellites and Crowd-Shipping”. In: *IEEE Transactions on Intelligent Transportation Systems*.
- Laporte, G. and Nobert, Y. (1988). “A vehicle flow model for the optimal design of a two-echelon distribution system”. In: *Advances in Optimization and Control*. Springer, pp. 158–173.
- Le, T. V., Stathopoulos, A., Van Woensel, T., and Ukkusuri, S. V. (2019). “Supply, demand, operations, and management of crowd-shipping services: a review and empirical evidence”. In: *Transportation Research Part C: Emerging Technologies* 103, pp. 83–103.
- Le, T. V. and Ukkusuri, S. V. (2019a). “Crowd-shipping services for last mile delivery: Analysis from American survey data”. In: *Transportation Research Interdisciplinary Perspectives* 1, p. 100008.
- Le, T. V. and Ukkusuri, S. V. (2019b). “Influencing factors that determine the usage of the crowd-shipping services”. In: *Transportation Research Record* 2673.7, pp. 550–566.
- Le, T. V., Ukkusuri, S. V., Xue, J., and Van Woensel, T. (2021). “Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems”. In: *Transportation Research Part E: Logistics and Transportation Review* 149, p. 102209.
- Lei, C. and Ouyang, Y. (2018). “Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems”. In: *Transportation Research Part B: Methodological* 109, pp. 90–109.
- Li, B., Krushinsky, D., Reijers, H. A., and Van Woensel, T. (2014). “The share-a-ride problem: People and parcels sharing taxis”. In: *European Journal of Operational Research* 238.1, pp. 31–40.
- Li, J. et al. (2007). “Who chooses to carpool and why? Examination of Texas carpoolers”. In: *Transportation Research Record* 2021.1, pp. 110–117.
- Li, Q., Liao, F., Timmermans, H. J., Huang, H., and Zhou, J. (2018). “Incorporating free-floating car-sharing into an activity-based dynamic user equilibrium model: A demand-side model”. In: *Transportation Research Part B: Methodological* 107, pp. 102–123.
- Li, W.-H., Huang, H.-J., and Shang, H.-Y. (2020). “Dynamic equilibrium commuting in a multilane system with ridesharing”. In: *Physica A: Statistical Mechanics and its Applications* 557, p. 124860.
- Li, X., Gao, J., Wang, C., Huang, X., and Nie, Y. (2022). “Ride-Sharing Matching Under Travel Time Uncertainty Through Data-Driven Robust Optimization”. In: *IEEE Access* 10, pp. 116931–116941.
- Li, X., Ma, J., Cui, J., Ghiasi, A., and Zhou, F. (2016). “Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model”. In: *Transportation Research Part B: Methodological* 88, pp. 21–45.
- Li, X. and Ouyang, Y. (2010). “A continuum approximation approach to reliable facility location design under correlated probabilistic disruptions”. In: *Transportation Research Part B: Methodological* 44.4, pp. 535–548.

- Li, Y., Li, X., and Zhang, S. (2021). “Optimal pricing of customized bus services and ride-sharing based on a competitive game model”. In: *Omega* 103, p. 102413.
- Li, Y. and Chung, S. H. (2020). “Ride-sharing under travel time uncertainty: Robust optimization and clustering approaches”. In: *Computers & Industrial Engineering* 149, p. 106601.
- Li, Z.-C., Huang, H.-J., and Yang, H. (2020). “Fifty years of the bottleneck model: A bibliometric review and future research directions”. In: *Transportation Research Part B: Methodological* 139, pp. 311–342.
- Li, Z., Hong, Y., and Zhang, Z. (2016). “Do ride-sharing services affect traffic congestion? An empirical study of uber entry”. In: *SSRN Electronic Journal* 2002, pp. 1–29.
- Lindsey, R., Palma, A. de, and Silva, H. E. (2019). “Equilibrium in a dynamic model of congestion with large and small users”. In: *Transportation Research Part B: Methodological* 124, pp. 82–107.
- Liu, Y. and Li, Y. (2017). “Pricing scheme design of ridesharing program in morning commute problem”. In: *Transportation Research Part C: Emerging Technologies* 79, pp. 156–177.
- Long, J., Tan, W., Szeto, W., and Li, Y. (2018). “Ride-sharing with travel time uncertainty”. In: *Transportation Research Part B: Methodological* 118, pp. 143–171.
- Lu, W., Liu, L., Wang, F., Zhou, X., and Hu, G. (2020). “Two-phase optimization model for ride-sharing with transfers in short-notice evacuations”. In: *Transportation Research Part C: Emerging Technologies* 114, pp. 272–296.
- Ma, R., Ban, X. J., and Szeto, W. (2017). “Emission modeling and pricing on single-destination dynamic traffic networks”. In: *Transportation Research Part B: Methodological* 100, pp. 255–283.
- Ma, R. and Zhang, H. (2017). “The morning commute problem with ridesharing and dynamic parking charges”. In: *Transportation Research Part B: Methodological* 106, pp. 345–374.
- Ma, T.-Y., Rasulkhani, S., Chow, J. Y., and Klein, S. (2019). “A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers”. In: *Transportation Research Part E: Logistics and Transportation Review* 128, pp. 417–442.
- Macrina, G., Pugliese, L. D. P., Guerriero, F., and Laporte, G. (2020). “Crowd-shipping with time windows and transshipment nodes”. In: *Computers & Operations Research* 113, p. 104806.
- Martin, E. and Shaheen, S. (2011). “The impact of carsharing on public transit and non-motorized travel: an exploration of North American carsharing survey data”. In: *Energies* 4.11, pp. 2094–2114.
- Masoud, N. and Jayakrishnan, R. (2017). “A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem”. In: *Transportation Research Part B: Methodological* 99, pp. 1–29.

- Masoud, N., Nam, D., Yu, J., and Jayakrishnan, R. (2017). “Promoting peer-to-peer ridesharing services as transit system feeders”. In: *Transportation Research Record* 2650.1, pp. 74–83.
- Mitrović-Minić, S. and Laporte, G. (2006). “The pickup and delivery problem with time windows and transshipment”. In: *INFOR: Information Systems and Operational Research* 44.3, pp. 217–227.
- Mousavi, K., Bodur, M., and Roorda, M. J. (2022). “Stochastic last-mile delivery with crowd-shipping and mobile depots”. In: *Transportation Science* 56.3, pp. 612–630.
- Nair, R. and Miller-Hooks, E. (2011). “Fleet management for vehicle sharing operations”. In: *Transportation Science* 45.4, pp. 524–540.
- Nieto-Isaza, S., Fontaine, P., and Minner, S. (2022). “The value of stochastic crowd resources and strategic location of mini-depots for last-mile delivery: A Benders decomposition approach”. In: *Transportation Research Part B: Methodological* 157, pp. 62–79.
- Ouyang, Y. (2007). “Design of vehicle routing zones for large-scale distribution systems”. In: *Transportation Research Part B: Methodological* 41.10, pp. 1079–1093.
- Özkan, E. and Ward, A. R. (2020). “Dynamic Matching for Real-Time Ride Sharing”. In: *Stochastic Systems* 10.1, pp. 29–70.
- Palma, A. de, Javaudin, L., Stokkink, P., and Tarpin-Pitre, L. (2022). “Ride-sharing with inflexible drivers in the Paris metropolitan area”. In: *Transportation*, pp. 1–24.
- Palma, A. de, Lefèvre, C., and Ben-Akiva, M. (1987). “A dynamic model of peak period traffic flows and delays in a corridor”. In: *Computers & mathematics with applications* 14.3, pp. 201–223.
- Palma, A. de, Stokkink, P., and Geroliminis, N. (2022). “Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users”. In: *Transportation Research Part B: Methodological* 155, pp. 479–498.
- Pfrommer, J., Warrington, J., Schildbach, G., and Morari, M. (2014). “Dynamic vehicle redistribution and online price incentives in shared mobility systems”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.4, pp. 1567–1578.
- Pisinger, D. and Ropke, S. (2010). “Large neighborhood search”. In: *Handbook of metaheuristics*. Springer, pp. 399–419.
- Pourrahmani, E. and Jaller, M. (2021). “Crowdshipping in Last Mile Deliveries: Operational Challenges and Research Opportunities”. In: *Socio-Economic Planning Sciences*, p. 101063.
- Punel, A., Ermagun, A., and Stathopoulos, A. (2019). “Push and pull factors in adopting a crowdsourced delivery system”. In: *Transportation Research Record* 2673.7, pp. 529–540.
- Qian, Z. S. and Zhang, H. M. (2011). “Modeling multi-modal morning commute in a one-to-one corridor network”. In: *Transportation Research Part C: Emerging Technologies* 19.2, pp. 254–269.

- Rais, A., Alvelos, F., and Carvalho, M. S. (2014). “New mixed integer-programming model for the pickup-and-delivery problem with transshipment”. In: *European Journal of Operational Research* 235.3, pp. 530–539.
- Raviv, T. and Kolka, O. (2013). “Optimal inventory management of a bike-sharing station”. In: *IIE Transactions* 45.10, pp. 1077–1093.
- Raviv, T. and Tenzer, E. Z. (2018). “Crowd-shipping of small parcels in a physical internet”. In: *URL: [http://www.eng.tau.ac.il/~talraviv/Publications/Crowd-shipping% 2of% 20small% 20parcels% 20in% 20a% 20physical% 20internet. pdf](http://www.eng.tau.ac.il/~talraviv/Publications/Crowd-shipping%20of%20small%20parcels%20in%20a%20physical%20internet.pdf)*. preprint.
- Repoux, M, Boyaci, B, and Geroliminis, N. (2015). “Simulation and optimization of one-way car-sharing systems with variant relocation policies”. In: *94th Annual Meeting of the Transportation Research Board USA, Washington DC*.
- Repoux, M., Kaspi, M., Boyacı, B., and Geroliminis, N. (2019). “Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations”. In: *Transportation Research Part B: Methodological* 130, pp. 82–104.
- Ropke, S. and Cordeau, J.-F. (2009). “Branch and cut and price for the pickup and delivery problem with time windows”. In: *Transportation Science* 43.3, pp. 267–286.
- Rougès, J.-F. and Montreuil, B. (2014). “Crowdsourcing delivery: New interconnected business models to reinvent delivery”. In: *1st international physical internet conference*. Vol. 1, pp. 1–19.
- Sampaio, A., Savelsbergh, M., Veelenturf, L. P., and Van Woensel, T. (2020). “Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers”. In: *Networks* 76.2, pp. 232–255.
- Santi, P. et al. (2014). “Quantifying the benefits of vehicle pooling with shareability networks”. In: *Proceedings of the National Academy of Sciences* 111.37, pp. 13290–13294.
- Santos, A. G. dos, Viana, A., and Pedroso, J. P. (2022). “2-echelon lastmile delivery with lockers and occasional couriers”. In: *Transportation Research Part E: Logistics and Transportation Review* 162, p. 102714.
- Savelsbergh, M. W. and Sol, M. (1995). “The general pickup and delivery problem”. In: *Transportation Science* 29.1, pp. 17–29.
- Schaller, B. (2021). “Can sharing a ride make for less traffic? Evidence from Uber and Lyft and implications for cities”. In: *Transport Policy* 102, pp. 1–10.
- Schulte, F. and Voß, S. (2015). “Decision support for environmental-friendly vehicle relocations in free-floating car sharing systems: The case of car2go”. In: *Procedia CIRP* 30, pp. 275–280.
- Shaheen, S. and Cohen, A. (2019). “Shared ride services in North America: definitions, impacts, and the future of pooling”. In: *Transport Reviews* 39.4, pp. 427–442.
- Singla, A. et al. (2015). “Incentivizing users for balancing bike sharing systems”. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

- Small, K. A. (1982). “The scheduling of consumer activities: work trips”. In: *The American Economic Review* 72.3, pp. 467–479.
- Small, K. A., Verhoef, E. T., and Lindsey, R. (2007). *The economics of urban transportation*. Routledge.
- Smilowitz, K. R. and Daganzo, C. F. (2007). “Continuum approximation techniques for the design of integrated package distribution systems”. In: *Networks: An International Journal* 50.3, pp. 183–196.
- Smith, M. (1993). “A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks”. In: *Transportation Research Part B: Methodological* 27.1, pp. 49–63.
- Soto Setzke, D. et al. (2017). “Matching drivers and transportation requests in crowd-sourced delivery systems”. In: *Twenty-third Americas Conference on Information Systems, Boston*.
- Spiess, H. and Florian, M. (1989). “Optimal strategies: a new assignment model for transit networks”. In: *Transportation Research Part B: Methodological* 23.2, pp. 83–102.
- Stokkink, P. and Geroliminis, N. (2021). “Predictive user-based relocation through incentives in one-way car-sharing systems”. In: *Transportation Research Part B: Methodological* 149, pp. 230–249.
- Stokkink, P. and Geroliminis, N. (2023). “A continuum approximation approach to the depot location problem in a crowd-shipping system”. In: *Transportation Research Part E: Logistics and Transportation Review* 176, p. 103207.
- Teal, R. F. (1987). “Carpooling: Who, how and why”. In: *Transportation Research Part A: General* 21.3, pp. 203–214.
- Toth, P. and Vigo, D. (2002). *The vehicle routing problem*. SIAM.
- U.S. Department of Energy (DOE) (2022). *Oak Ridge National Lab (2022) Transportation Energy Data Book Edition 40*.
- Vickrey, W. S. (1963). “Pricing in urban and suburban transport”. In: *The American Economic Review* 53.2, pp. 452–465.
- Vickrey, W. S. (1969). “Congestion theory and transport investment”. In: *The American Economic Review* 59.2, pp. 251–260.
- Vincent, F. Y., Jodiawan, P., and Redi, A. P. (2022). “Crowd-shipping problem with time windows, transshipment nodes, and delivery options”. In: *Transportation Research Part E: Logistics and Transportation Review* 157, p. 102545.
- Voigt, S. and Kuhn, H. (2022). “Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers”. In: *Networks* 79.3, pp. 403–426.
- Wandelt, S., Dai, W., Zhang, J., and Sun, X. (2022). “Toward a reference experimental benchmark for solving hub location problems”. In: *Transportation Science* 56.2, pp. 543–564.

- Wang, X. and Ouyang, Y. (2013). “A continuum approximation approach to competitive facility location design under facility disruption risks”. In: *Transportation Research Part B: Methodological* 50, pp. 90–103.
- Wang, X., Agatz, N., and Erera, A. (2018). “Stable matching for dynamic ride-sharing systems”. In: *Transportation Science* 52.4, pp. 850–867.
- Wang, Y., Zhang, D., Liu, Q., Shen, F., and Lee, L. H. (2016). “Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions”. In: *Transportation Research Part E: Logistics and Transportation Review* 93, pp. 279–293.
- Weikl, S. and Bogenberger, K. (2013). “Relocation strategies and algorithms for free-floating car sharing systems”. In: *IEEE Intelligent Transportation Systems Magazine* 5.4, pp. 100–111.
- Xiong, C. et al. (2019). “An integrated and personalized traveler information and incentive scheme for energy efficient mobility systems”. In: *Transportation Research Part C: Emerging Technologies*.
- Yang, H., Shao, C., Wang, H., and Ye, J. (2020). “Integrated reward scheme and surge pricing in a ridesourcing market”. In: *Transportation Research Part B: Methodological* 134, pp. 126–142.
- Yatskiv, I. and Budilovich, E. (2017). “A comprehensive analysis of the planned multimodal public transportation HUB”. In: *Transportation research procedia* 24, pp. 50–57.
- Yıldız, B. (2021a). “Express package routing problem with occasional couriers”. In: *Transportation Research Part C: Emerging Technologies* 123, p. 102994.
- Yıldız, B. (2021b). “Package routing problem with registered couriers and stochastic demand”. In: *Transportation Research Part E: Logistics and Transportation Review* 147, p. 102248.
- Yildiz, B. and Savelsbergh, M. (2019). “Service and capacity planning in crowd-sourced delivery”. In: *Transportation Research Part C: Emerging Technologies* 100, pp. 177–199.
- Yu, X., Berg, V. A. van den, and Verhoef, E. T. (2019). “Carpooling with heterogeneous users in the bottleneck model”. In: *Transportation Research Part B: Methodological* 127, pp. 178–200.
- Zenou, Y. (2009). *Urban labor economics*. Cambridge University Press.
- Zha, L., Yin, Y., and Du, Y. (2018). “Surge pricing and labor supply in the ride-sourcing market”. In: *Transportation Research Part B: Methodological* 117, pp. 708–722.
- Zha, L., Yin, Y., and Xu, Z. (2018). “Geometric matching and spatial pricing in ride-sourcing markets”. In: *Transportation Research Part C: Emerging Technologies* 92, pp. 58–75.

Curriculum vitae



Patrick Stokkink

05-05-1997

www.linkedin.com/in/patrickstokkink

patrick.stokkink@epfl.ch

patrick.stokkink@epfl.ch

+31 6 27386307

Address

Allée des Cèdres 1
1022 VD,
Chavannes-Près-Renens
Switzerland

Patrick Stokkink

Education

**Sept 2018 - Aug 2019, Master Econometrics and Management Science,
Specialization: Operational Research and Quantitative Logistics**

Erasmus University Rotterdam, The Netherlands

GPA 9.1 / 10 (summa cum laude)

Master thesis on a robust optimization approach to the capacitated vehicle routing problem using representatives. This thesis was rewarded with grade 9.0 / 10.

Sept 2015 - Aug 2018, Bachelor Econometrics and Operations Research

Erasmus University Rotterdam, The Netherlands

GPA 8.7 / 10 (cum laude)

Bachelor thesis on a tabu-search heuristic for the vehicle routing problem with self-imposed time windows. This thesis was rewarded with grade 8.5 / 10.

Academic Experience

Sept 2019 - now, PhD candidate, École Polytechnique Fédérale de Lausanne

I am a PhD student in the Urban Transport Systems Laboratory (LUTS). My primary research fields and main research interests involve on-demand mobility and transportation systems. I use a combination of optimization, simulation and control methods to solve operational and strategic challenges in these systems. My current work is mainly focused on large-scale ride-sharing and crowd-shipping systems. I am an instructor and course coordinator for the Transportation Economics course for the Civil Engineering master program. [EPFL coursebook](#)

April 2022 - Aug 2022, Visiting PhD, École des Hautes Études Commerciales (HEC) in Montreal

During the summer of 2022 I visited the CIRRELT research centre as a visiting PhD. CIRRELT is a cross-university research center focussing on logistics and transportation. During my visit, I worked closely together with Prof. Jean-François Cordeau.

May 2017 - July 2019, Research and Teaching Assistant, Erasmus University Rotterdam

During my bachelor and master at Erasmus University Rotterdam, I was employed at the department of Econometrics and Operational Research as a research and teaching assistant. As a teaching assistant, I facilitated exercise lectures, and was involved in the construction and grading of exams for, a.o., the courses Statistics, Markov Processes and Non-Linear Optimization

Publications

Published:

- J1. A continuum approximation approach to the depot location problem in a crowd-shipping system
Patrick Stokkink and Nikolas Geroliminis
Transportation Research Part E: Logistics and Transportation Review, 2023
<https://arxiv.org/pdf/2202.05016.pdf>
- J2. Incentive-based electric vehicle charging for managing bottleneck congestion
Carlo Cenedese, Patrick Stokkink, Nikolas Geroliminis, and John Lygeros
European Journal of Control, 2022
<https://www.sciencedirect.com/science/article/pii/S0947358022000905>
- J3. Ride-sharing with inflexible drivers in the Paris metropolitan area
André de Palma, Lucas Javaudin, Patrick Stokkink, and Léandre Tarpin-Pitre
Transportation, 2022
<https://link.springer.com/article/10.1007/s11116-022-10361-1>
- J4. Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users
André de Palma, Patrick Stokkink, and Nikolas Geroliminis
Transportation Research Part B: Methodological, 2022
<https://www.sciencedirect.com/science/article/pii/S0191261521002113>

- J5. Predictive user-based relocation through incentives in one-way car-sharing systems
Patrick Stokkink and Nikolas Geroliminis
Transportation Research Part B: Methodological, 2021
<https://www.sciencedirect.com/science/article/pii/S0191261521000874>

Under review:

- R1. Multi-modal ride-matching with transfers and travel-time uncertainty
Patrick Stokkink, André de Palma, and Nikolas Geroliminis
on-request

Working papers:

- W1. A column generation approach to the crowd-shipping problem with transfers
Patrick Stokkink, Jean-François Cordeau, and Nikolas Geroliminis
on-request
- W2. Adaptive reward strategies in a crowd-shipping system for urban last-mile delivery
Patrick Stokkink, Jinwen Ye, Giovanni Pantuso, David Pisinger, and Nikolas Geroliminis
on-request

Presentations

- Multi-modal Ride-Matching with Transfers, **Swiss Transport Research Conference (STRC)**, Switzerland, May 10, 2023
- A Column Generation Approach to the Crowd-Shipping Problem with Transfers, **Transportation Research Board (TRB) 102nd Annual Meeting**, Washington, USA, January 10, 2023
- Predictive user-based relocation through incentives in one-way car-sharing systems, **The 24th International Symposium on Transportation and Traffic Theory (ISTTT24)**, Beijing, China, July 25, 2022
- A Continuum Approximation Approach to the Hub Location Problem in a Crowd-Shipping System, **The 11th Triennial Symposium on Transportation Analysis (TRISTANXI)**, Mauritius, June 20, 2022
- Managing Bottleneck Congestion through Incentives on Electric Vehicle Charging and Lane Segmentation, **10th Symposium of the European Association for Research in Transportation (hEART)**, Leuven, Belgium, June 1, 2022
- A Continuum Approximation Approach to the Hub Location Problem in a Crowd-Shipping System, **Journées de l'Optimisation 2022 (JOPT)**, Montréal, Canada, May 18, 2022
- A Continuum Approximation Approach to the Hub Location Problem in a Crowd-Shipping System, **Transportation Research Board (TRB) 101st Annual Meeting**, Washington, USA, January 11, 2022
- Influence of dynamic congestion with scheduling preferences on carpooling matching with heterogeneous users, **Transportation Research Board (TRB) 101st Annual Meeting**, Washington, USA, January 12, 2022
- A Continuum Approximation Approach to the Hub Location Problem in a Crowd-Shipping System, **Swiss Transport Research Conference (STRC)**, Switzerland, September 12, 2021
- Influence of dynamic congestion with scheduling preferences on carpooling matching, **International Transportation Economics Association (ITEA)**, Rome, Italy, June 26, 2021
- Predictive user-based relocation through incentives in one-way car-sharing systems, **Forum on Integrated and Sustainable Transportation System (ISTS)**, Delft, the Netherlands, November 4, 2020
- Influence of Dynamic Congestion on Carpooling Matching, **Swiss Transport Research Conference (STRC)**, Switzerland, May 13, 2020

Reviewed Journal Articles for:

European Journal of Operational Research, IEEE ITS Transactions, Journal of the Operational Research Society, Transportation Research Part A, Part B, Part C, Part E, Transportmetrica A: Transport Science

Master Thesis Supervision

- Alfio Simone Mosset, Pricing in large-scale crowd-shipping systems, *nominated for a best-thesis award in transportation*
- Léandre Tarpin-Pitre, Ride-sharing with inflexible drivers in the Paris metropolitan area, *published in an academic journal*
- Minru Wang, The impact of ride-splitting revenue optimization on service level and traffic operation, *continued her research on this topic during her PhD*
- Yassine El Ouazzani, Optimized path planning of a swarm of drones for massive data collection

Languages

Dutch (native), English (fluent), German (intermediate), French (intermediate)

Programming Languages

Java, Matlab, C#, Python

Personal Hobbies

Travelling, Photography, Running, Hiking