Innovative business-to-business last-mile solutions

INNOVATIVE BUSINESS-TO-CONSUMER LAST-MILE SOLUTIONS

MODELS AND ALGORITHMS

Printed by Proefschriftmaken.nl || Uitgeverij BOXPress
Cover design by Ostapenko Olena

This thesis is part of the Ph.D. thesis series of the Beta Research School for Operations Management and Logistics in which the following Dutch universities cooperate: Eindhoven University of Technology, Maastricht University, University of Twente, VU Amsterdam, Wageningen University and Research, and KU Leuven.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN 978-90-386-5192-7

Innovative Business-to-Consumer Last-Mile Solutions

Models and Algorithms

Proefschrift

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven
op gezag van de Rector Magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie
aangewezen door het College voor Promoties, in het openbaar te
verdedigen op woensdag 27 januari 2021 om 13:30 uur

door

Afonso Henrique Sampaio Oliveira

geboren te Belo Horizonte, Minas Gerais, Brazilië.

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter	prof.dr.ir G.J.J.A.N. van Houtum
1e promotor:	prof.dr. T. Van Woensel
copromotor(en):	dr. L.P. Veelenturf dr. J. Kinable
leden:	prof.dr. M. Savelsbergh (Georgia Institute of Technology) prof.dr. T. Vanelslander (Universiteit Antwerpen) prof.dr. D. Vigo (Università di Bologna) prof.dr. F.C.R. Spieksma prof.dr. A.G. de Kok
reserve lid:	dr.ir. N.P. Dellaert

Prof. dr. M. Savelsbergh heeft in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven in uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Beta Research School for Operations Management and Logistics

NWO Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Muita coisa é inata, mas muito é feito pelo treinamento. Por isso, ninguém será bem-sucedido se se poupar, se não mergulhar fundo nos temas maiores e se não estiver em condições de, às vezes, se empenhar até o extremo por causas insignificantes.[1]

Walter Benjamim, Rua de Mão Única - Obras Escolhidas II.

[1] Many things are innate, but a lot is done by training. Therefore, no one will be successful if (s)he spares (her)himself, if (s)he does not delve deeply into the bigger issues and if (s)he is not willing to strive to the extreme even for the insignificant causes.

Acknowledgements

Feeling gratitude, and not expressing it,
is like wrapping a present and not giving it.

William Arthur Ward

I would like to express my deepest gratitude to all who contributed to the completion of this thesis, one way or another, for their help and support throughout. Even if not properly wrapped, I'm giving this present as it is, and that should be the most important.

Thanks to Tom van Woensel, for having given me the opportunity of embarking on the PhD journey, for your support along the way, providing directions and valuable insights on all parts of the thesis. Thanks for believing in my work, always sharing your expertise and access to your network. Moreover, thanks for your patience in dealing with my continuous tendency on delaying deadlines. I like to believe I have improved on my organizational skills, and you had a good influence on that.

I would also like to thank Luuk Veelenturf for his daily supervision and friendship! You were always available for discussing ideas, providing crucial guidance to make the work progress, but also to discuss and help with other relevant (and also the not so relevant!) matters. You always shown enthusiasm in the work we did together, making sure that our research addressed the complexities of solving hard, significant problems but without losing sight of the practical relevance of the underlying problems. I would not be in a position of defending my thesis were not for you help. Thanks, Luuk!

Also, thanks to Joris Kinable, for also being part of my daily supervision. Although we mostly worked together during the last year of the PhD, it was not short of learning, insightful ideas and discussions. Your enthusiasm and knowledge for computational problems were a tremendous help for completing Chapters 4 and 5. Not to mention that you also preferred using an environment more suitable to development, a.k.a a Linux distro, on your laptop. Although there were also moments of disagreement, thank you for the invaluable guidance.

During my PhD, I had the great privilege to work with Martin Savelsbergh. I would like to thank you for hosting me during my visit to Geogia Tech. Despite your extensive knowledge and experience, you always listened and tried to understand what I had to say. It never ceased to impress me how, no matter how busy, you made the time to have insightful discussions, to read and give deep feedback on the work we did. Thank you for the cooperation that resulted in Chapters 2 and 3 (and the papers that resulted from them) and for being part of the committee.

Many thanks to Thierry Vanelslander, Daniele Vigo, Frits Spieksma, Ton de Kok and Nico Dellaert for also being part of the committee, for the time you spent reading and reviewing the thesis. I really appreciated our discussion during the review meeting. Thanks for the feedback, corrections and suggestions that helped improving the final version of the text.

I would like to thank Jolanda, José, Claudine, Christel and Geertje for the support they provided throughout the years by arranging all meetings, synchronizing schedules and making sure we all had what we needed to do our work. Special thanks to Jolanda for your help in putting me in contact with your network, that eventually landed me my current job!

I also have to mention the nice environment at TU/e, in general, and at OPAC, in particular. I would like to thank all great people I had the chance to meet there, specially to my friends in the transportation research, with whom I shared office and insightful ideas: Peng, Fardyn, Vincent, Natasja, Albina, Yanfei and Alexandre.

Looking back since the end of 2015, when I started the PhD at 30, I did not expect to have made friendships as strong as the ones I made in those last years. *The time you enjoy wasting is not wasted time*! Then I would like to specially thank some folks for the wasted time together: thanks Sami, Virginie, Simon and Claudia for the many Thursday nights drinks and for the nice summer trip! Sonja, for sharing nice conversations and cooking skills! Shaunak, for sharing a beer at a real Dutch bar, and to my running mates Prince and Volkan! Joost and Erwin for the many game board sessions we played together. Thanks to Reza, Mehdi and all my other Iranian friends for organizing football matches and invite me for the futsal tournaments. It goes without saying how much your companionship meant to me and that you all had a special part in this journey.

Moving to the Netherlands to pursue a PhD meant not only I had to give up on my dream to be a football player and have a Ferrari, but also that I would no longer be in close reach with my dearest Brazilian friends. Fortunately, technology is advanced enough and those friendship so strong that distance was not an issue. For the philosophical discussions which helped endure in the face of failure but rejoice in the eventual success, for sharing some few bad but lots of great moments, for the companionship, my sincerely big thanks to Harlley, Henrique, Phillipe, Felipe, Nayara and Sara *amigo — é que a gente seja, mas sem precisar de saber o por quê é que é!* (Actually, Harlley could not stand being so far, and eventually also moved to the Netherlands, so we could be closer and share our suffering while cheering for Atlético Mineiro here).

Of course, someone with whom I shared many wonderful moments: thanks Anne for your support and patience. Without you this journey would most likely lack many new things to me. At least, I would not have had experienced the Dutch/Brabant culture the same! Many thanks also to your family, specially to Gidi and Margreeth, for the many nice barbeques and special beers. I hope they will, eventually, appreciate chicken hearts on the skewer.

Last, but certainly not least, thanks, praise and love to my mother, Clarinda, and my to my sister, Tainara. Because of these two, I don't remember any moment of worry or fear during this work - I always could count on their support and it's hard to express how hugely I am grateful to them. For all his previous support and great incentive, from the

very beginning of my life to the end of his, I would not be in the position of accomplish this work if was not by my father, Afonso — *only some time after a man's death we regard his absence as incomprehensible.*

Eindhoven, December 2020

CONTENTS

List of Figures

List of Tables

1

INTRODUCTION

You had my curiosity... but now you have my attention.

Quentin Tarantino, Django Unchained

OR so I hope! This thesis focuses on novel approaches for dealing with last-mile operations faced by logistics service providers in urban contexts. The last-mile refers to the last link in the transport chain followed by a parcel to fulfill consumers' requests for goods, from the shelf of the last distribution center to the hands of the buyer. We investigate two recent innovations and the potential cost-benefits of introducing such models into transportation logistics for last-mile operations. More specifically, we first consider a *crowd-sourced* solution – where drivers are not employed by a carrier but occasionally offer their services through on-line platforms and are contracted as required by the carrier – for the fulfillment of transportation requests and evaluate the benefits of introducing transfers to support driver activities. We frame the problem as an extension of a pickup and delivery problem with transfers and propose a heuristic optimization method to solve it. The second novel model we consider is what has been defined as roaming delivery systems, in which the service provider has access to private cars' storage compartments, and can service customers using the trunk of their cars. Supported by automotive and communication technologies, the model has the potential to make e-commerce operations more convenient, mitigating failed deliveries at home. We introduce a stochastic version of the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) and propose a two-stage stochastic model using the possibility of servicing customers at different locations as a recourse action. Finally, we introduce a dynamic variant of the VRPRDL in which customers announce in real-time the locations where their cars are or will be parked and the service provider decides whether visiting customers at home or at their roaming locations.

1

1.1. THE RISE OF E-COMMERCE AND LAST-MILE CHALLENGES

IT took me a while to recall when I had my first on-line purchase experience and, if memory serves me right, buying that digital camera was a mixture of excitement and skepticism: I only had to create a customer account, select the camera's model, use the numbers of a credit card that was not even mine and the purchase would be sealed. Not having someone from the shop present to make the transaction a more formal occasion was a concern but the deal too good to pass. After a few days, worries that we had been scammed and would have to visit the store, after all, the delivery person was at the door. Nowadays, online shopping is so pervasive that it is hard to remember when it was only a doubtful possibility. The time from purchase to delivery has decreased significantly, orders are now placed on mobile devices and delivered not only at home.

By facilitating the search of products, the comparison of the many available options and prices, more and more consumers are using the internet for ordering from tonight's dinner to a larger TV to the latest smart phone. While total retail sales have been steadily increasing worldwide, jumping from U$22.97 trillion in 2017 to U$25.04 trillion in 2019, the share of e-commerce[1] in the global retail sales has increased from 10% to 14.1% during the same period. By 2023, it is projected that e-commerce will represent 22% of retail sales worldwide, reaching U$6.54 trillion (Lipsman, 2019).

This was the scenario until the beginning of 2020, when the COVID-19 crisis has brought unprecedented changes to many human activities. As tight and tighter measures were imposed to halt the spread of the virus and diminish the human toll, the economic damage plunged most countries into recession in 2020. On the one hand, social-distancing policies meant that most physical interactions were significantly limited and, combined with strict confinement measures, have put on hold almost all of traditional brick-and-mortar retail. On the other hand, whereas total retail sales have decreased, the crisis has amplified the aforementioned shift from traditional means of purchasing to e-commerce – for example, in the EU, while total retail sales in April 2020 decreased by 17.9% compared to April 2019, sales via mail or via the Internet increased by 20% during the same period (OECD, 2020). The crisis is accelerating the expansion of e-commerce, which proved to be a crucial element during these times as a means of keeping customers with access to a variety of products from the convenience and safety of their, and by allowing many business to continue operating, even under social-distancing measures. The landscape of e-commerce has been changing significantly in response to the COVID-19 pandemic and will likely continue to be even more dynamic as the crisis unfold. E-commerce growth is expected to increase at an even faster pace than predicted before 2020.

Despite all benefits, with the explosion of e-commerce and on-line sales new issues emerged, challenging logistic service providers to handle the sheer amount of direct-to-consumer orders while still providing efficient service. In 2018, the six largest carriers in The Netherlands, a country with a population of 17.18 million inhabitants, delivered

[1] including products or services ordered via the internet, excluding travel and event tickets, payments, food services or gambling sales

504 million parcels, with a turnover of €2.15 billion (Authority for Consumers & Markets, 2019). For around 10% of those parcels, delivering to the correspondent address or to a neighbor was not possible during the first visit and, consequently, the company had to arrange an alternative option e.g., a second visit, which can increase the total transportation cost significantly, or ask the customer to pickup the parcel at a collection point, which decreases customer's experience. The fulfillment of so many individual orders at consumers' homes cannot be done at the same economies of scales when replenishing retail stores, increasing the number of freight movements (Savelsbergh and Van Woensel, 2016). Inevitably, the growing number of delivery vans in residential areas and in the city centers contributes negatively to living conditions, reducing safety (Bandler et al., 2019) and adding to already increasing levels of congestion and growing emissions of pollutants in the environment (DePillis, 2019). Getting a parcel delivered to its purchaser at the right address and at the right time incurs not only in high operational costs but also in high environmental impacts.

In the transport chain followed by a product from the warehouse shelf to the customer's hands, the so called last-mile represents the very last step of that sequence. But despite comprising only one of the many steps in the total process, last-mile logistics account for around 50% of the total costs of shipping (McCrea, 2016) – it is a highly work intensive process (Wall, 2019), hindered by slow travel speeds within cities, and performed by low capacity vehicles with low fill rates. Complicating matters further, customers are more and more the driving factor when it comes to how the last-mile supply chains of e-tailers are organized. While traditional brick-and-mortar stores offer instant gratification to consumers, in online retail getting the product to the customer is an important, if not crucial, part of the experience, as it represents the single concrete touchpoint for e-commerce: customers tend to not differentiate between the retailer and the delivery service provider. In the era of social media, a faulty delivery experience is quickly shared by annoyed customers, reaching many potential clients before the delivery person rings the door-bell.

In an attempt to attract customers for shopping online, e-tailers have started to offer increased service levels, promising next-day or even same-day, 2-hour delivery services, sometimes not even charging extra fees. Fast and (almost) free delivery services have become commoditized, not in the least an extra enhancement for customers' experience while shopping online but a competitive edge. As such, and considering all challenges involved, last-mile logistics remain a cornerstone for business-to-consumer (B2C) e-commerce. Due to the high operational costs and environmental impacts, logistic service providers are considering new and innovative concepts to provide more efficient and sustainable models to overcome the challenges in last-mile operations.

1.1.1. INNOVATIVE MODELS IN B2C LAST-MILE SYSTEMS

In the last few decades, advances in computing and communications systems have drastically transformed modern societies. Recent innovations and the increased use of information technology are having far-reaching impacts not only on how we shop, but also on how we work, travel, learn and interact with others. Not surprisingly, technology progresses are also being introduced in the last-mile sector. Together with advances in the automotive industry, information technology is being used not only to find more cost efficient delivery plans, but also to support innovative last-mile models focusing toward automation, safety and emission reduction.

Perhaps one of the most significant changes due the use of information technology is in the way we access and use services like ordering food, a ride, a place to stay or even hiring a handyman to mount a TV. The last decade has seen the rise of the *sharing economy*. Also termed *gig economy*, it is characterized by a business model that enables and facilitates the sharing of goods and services (Hu, 2019). It represents a shift from traditional schemes of ownership and workforce employment to models in which physical assets are managed as services that can be shared and reused at a lower price and more flexibly.

In the last-mile sector, solutions within the sharing economy have been proposed connecting demand for (last-mile) logistic services to a *crowd-based* supply offering excess capacity in terms of time and/or space to perform such services, using their own means. Generally, these services are offered by individuals without certified logistic skills and working without any formal professional contract. As such, this concept is commonly referred as "Uber for logistics" (Rai, 2019).

1.2. RESEARCH PROBLEMS

In this thesis, we investigate two recent last-mile novel systems built around the idea of using excess capacity (time/space) of non-professional individuals, possibly customers, to improve last-mile operations within urban contexts. In the first one, we focus on a crowd-sourced system where drivers express their availability to perform delivery tasks for a given period of time and a platform communicates a schedule with requests to serve. In the second, we consider a last-mile model in which direct-to-consumer deliveries can be made using the trunk of the customer's car, while the vehicle is parked at a location along the customer's itinerary.

1.2.1. LAST-MILE IN THE ERA OF SHARING ECONOMY: CROWD-SHIPPING

Solutions built around the emerging concept of *sharing economy* are one of the alternatives that some companies are now exploring to supplement traditional freight transportation systems (e.g., last mile). *Crowd-shipping* refers to contexts where transportation capacity is provided by individuals willing to provide their time and vehicles for a limited (short) duration. In one realization of *crowd-shipping*, the transportation service is performed by individuals during trips they would drive anyway e.g., from home to work. An on-line platform matches trips to suitable requests based on a number of aspects, such as the required detour to visit the delivery address, offered compensation. This realization of crowd-shipping solutions are still in a very incipient stage (Dablanc et al., 2017). In another realization of the concept, already being implemented by large e-tailers and platforms (e.g, Amazon Flex, Uber-Freight), drivers willing to perform trans-

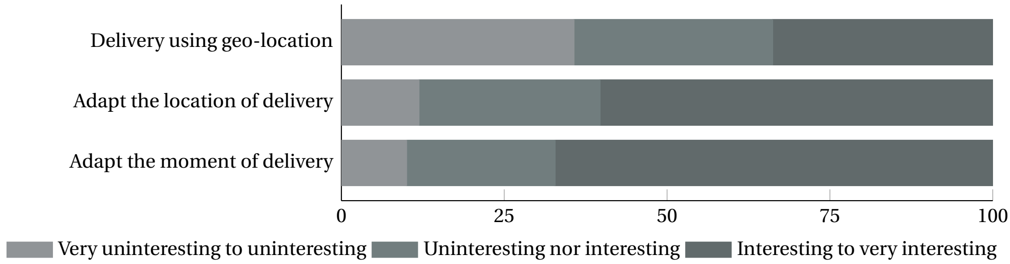


Figure 1.1: Consumers' perception of last mile innovations regarding delivery flexibility. Adapted from Rai (2019).

portation services are matched to demand for such services in real-time by the on-line platform. The drivers are independent, work for a given period of time and are paid on a hourly basis. Therefore, the platform is able to manage available capacity in a more flexible way (e.g., deciding on the requests and the sequence of visits). In the thesis, we focus on the latter realization of *crowd-shipping*.

1.2.2. ENTER THE TRUNK: THE CAR AS A B2C LAST-MILE SOLUTION

In a survey conducted with 1000 consumers in Belgium, Rai (2019) aimed at capturing consumers' attitude towards crowd logistics and identifying which crowd logistics services are considered of interest. In particular, one of the points surveyed was related to consumers interest for more flexible last-mile delivery options. The survey showed that 72,2% of consumers would like to be able to adapt the location of delivery, even if the order is already placed and in transit. Moreover, 33.8% of the consumers showed interest in an option in which parcels could be delivered on their geo-location i.e., the current customer location at the time of delivery. Figure 1.1 reports the findings of the survey for that particular question.

Trunk delivery is a novel approach to last-mile delivery being tested by even the largest e-tailers companies (e.g., Amazon In-Car Delivery) as a means to provide customers more flexible delivery options. In this approach, the service provider has access to the trunk of the customer's vehicle where delivery couriers can leave packages. Some researches show that the average car is parked away from home for a significant period of time, turning the vehicle, virtually, into a pickup/delivery station on wheels. Such models have the potential of alleviate failed deliveries at home, one of the main logistics service provider's challenges.

1.2.3. RESEARCH QUESTIONS

With the scenario presented so far and the particular last-mile systems we consider, the research objective of the thesis can be summarized as: investigate and quantify possible cost benefits and trade-offs in using spare capacity, either in the form of available time or available space, offered by private individuals as well as customers, to perform last-mile services within urban contexts.

In each chapter of the thesis, a particular sub-question will be considered within a given last-mile application and aiming toward the main objective, namely:

- What are the characterizing features of *crowd-logistics*, specially *crowd-shipping*?
- How available crowd-sourced capacity (drivers) can be managed and used in order to effectively meet transportation demand?
- Can trunk delivery be an option to mitigate failed deliveries due to uncertainty in servicing customers at planned locations in a priori designed delivery routes?
- Is trunk delivery still effective when information regarding customers' itineraries are unknown beforehand?

1.3. CONTRIBUTIONS

Rapid urban growth has posed both challenges and opportunities for city planners, not in the least when it comes to the design of transportation and logistics systems for freight. But urbanization also fosters innovation and sharing, which have led to new models for organizing movement of goods within the city. Chapter 2 highlights one of these new models: *Crowd-Logistics*. We define the characterizing features of crowd logistics, review applications of crowd-based services within urban environments, and discuss research opportunities in the area of crowd logistics.

In Chapter 3, we examine the potential benefits of introducing transfers to support pickup and delivery operations in urban areas employing crowd-sourced drivers. Transfer locations are service points available in the network and serve as intermediate locations where one driver can exchange freight loads with other drivers. We believe our research helps in advancing the implementation of decision-making tools used to operate crowd-sourced delivery systems and understanding how such systems can be better explored.

In the second part of the thesis, we consider last-mile systems in which the service provider has access to the trunk of the customer's car. Chapter 4 considers a problem recently proposed to model such systems, the Vehicle Routing Problem with Roaming Deliveries Locations (VRPRDL). Our contribution to this routing literature stems from considering stochastic travel times while solving the problem. We believe such contexts are important given the increasing level of urbanization and its consequences on the ability of retail companies to fulfill promised service levels (in particular, in e-commerce) for last-mile delivery within urban environments e.g., due to increased levels of congestion.

In Chapter 5, we drop the assumption that customers' itineraries are known to the service provider when deciding on the routing plans to serve customers. Alternatively, we consider that a customer announces the location where his/her car is currently or will be parked dynamically, as time goes by, defining a novel dynamic variant of the VRPRDL. We investigate the trade-offs a delivery company might consider when offering

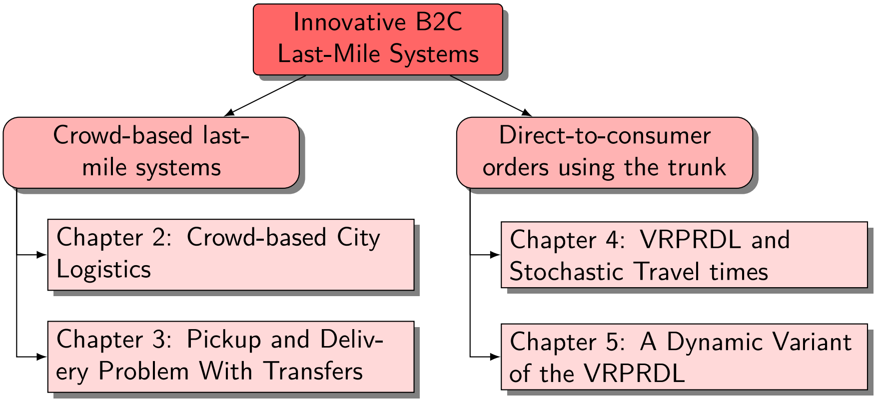


Figure 1.2: Illustrative outline of the thesis

such service compared to a traditional, only-home delivery option. Moreover, we evaluate the potential benefits of including dynamic roaming locations as a mean to integrate the fulfillment of both delivery and return (pickup) flows.

1.4. OVERVIEW

For the sake of clarity, Figure 1.2 presents a diagrammatic overview of the thesis, positioning each chapter within the two main topics considered in the work. Each chapter in this thesis is self-contained and can be read individually. Consequently, back-to-back readers might notice some overlap, especially during introduction sections, where the problems are motivated.

Finally, when solving vehicle routing problems, the common assumption is that the values of all input parameters e.g., travel times, customer demands, are known before the design phase of the routing plans and will not change during the execution phase of the plans. Input data can be classified accordingly to two main dimensions: evolution and quality of information (Bektas et al., 2014a). The former concerns how data changes or become available as time passes by, whereas the latter specifies the level of certainty regarding the value of an input parameter. In a *static and deterministic* problem, all input parameters are known in advance with a high degree of certainty, and they do not change after the problem is solved. In *static and stochastic* problems, input parameters are revealed in stages but the degree of uncertainty is captured by e.g., random or stochastic variables. In *dynamic* problems, part or all input data is not known beforehand, and only revealed concomitantly to the execution of the routing plan. Here, the system must react quickly as new information is revealed (*dynamic and deterministic*), or anticipate future information to better handle incoming events, if stochastic information is available (*dynamic and stochastic*). Figure 1.3 illustrates a matrix in which the two aforementioned dimensions for classifying input data are represented. With the exception of Chapter 2,

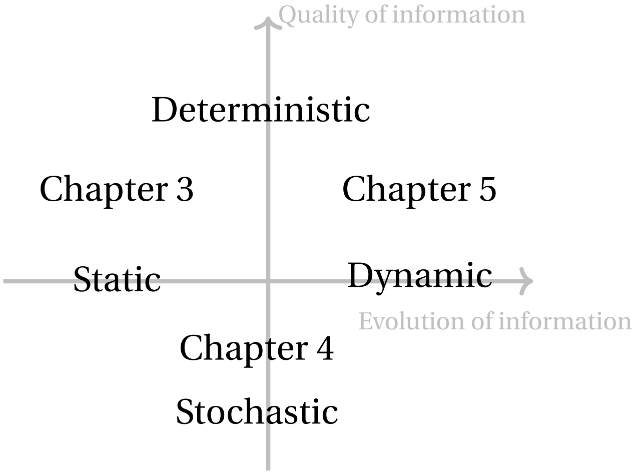


Figure 1.3: Classification of the problems treated on each chapter.

each chapter of the thesis is positioned on a quadrant of that matrix accordingly to the evolution and quality of information assumed on the problem being considered.

2

Crowd-Based City Logistics

We tend to overestimate the impact of a new technology in the short run,
but we underestimate it in the long run.

Roy Amara

Cities are drivers of economic development, providing infrastructure to support countless activities and services. Today, the world's 750 biggest cities account for more than 57% of the global GDP and this number is expected to increase to 61% by 2030. More than half of the world's population lives in cities, or urban areas, and this share will continue to grow. Rapid urban growth has posed both challenges and opportunities for city planners, not in the least when it comes to the design of transportation and logistics systems for freight. But urbanization also fosters innovation and sharing, which have led to new models for organizing movement of goods within the city. In this chapter, we highlight one of these new models: Crowd Logistics. We define the characterizing features of crowd logistics, review applications of crowd-based services within urban environments, and discuss research opportunities in the area of crowd logistics.

The work in this chapter has been published in Sampaio et al. (2018).

2

2.1. INTRODUCTION

City Logistics advocates a holistic view of the transport and logistic activities within a city, considering the negative (e.g., congestion and pollution) as well as the positive (e.g., economic, mobility, safety) impacts on the city's population. It seeks cost-efficient, but sustainable, solutions that minimize required flows of people and goods (Savelsbergh and Van Woensel, 2016; Crainic and Montreuil, 2016). In this chapter, we focus on how the *crowd* may be part of such cost-efficient, but sustainable solutions, especially related to the flows of goods.

As the urbanization trend will continue in the coming decades, the number of people living in urban areas is expected to grow from today's 54% of the world population to 66% by 2050 (United Nations, 2015). Moreover, the largest 750 cities in the world are responsible for more than 57% of the global GDP and this share is expected to increase to 61% by 2030 (Oxford Economics, 2017). At the same time, in today's increasingly global and interconnected world, the share of e-commerce of total global retail sales is also expected to continue to increase, from 7.4% in 2015 to 15.5% in 2021 (eMarketer Editors, 2017). Furthermore, consumers have higher service expectations than ever before. In a survey with more than 2,000 customers in the US, 64% of those interviewed indicated they are willing to pay a premium for faster delivery, and 39% would pay more for same day delivery (Accenture Interactive, 2015). E-tailers are stimulating and exploiting these service expectations by offering fast delivery options as a means to compete with brick-and-mortar retailers and, in many cases, are not even charging consumers for the increased level of service (Savelsbergh and Van Woensel, 2016). The population growth and urbanization, the explosion of e-commerce, and the proliferation of fast delivery options, require innovative solutions and business models to ensure cost-effective, but also environmentally and socially friendly, transportation of goods.

Logistic practices in which infrastructure sharing and service integration are core concepts represent a new trend in transportation systems aimed at providing a more economically and environmental gainful alternative to current practices. Facilitated by advances in information and communication technologies and the ubiquity of personal, mobile smart devices, a shift to new ways of collaborative consumption is seen. Often termed *sharing economy*, this new phenomenon is characterized by managing physical (underused) assets as services and recognizing the possible benefits of the temporary use of a third-party service over the long-term possession of an (expensive) asset (DHL Trend Research, 2017). Whereas the concept of outsourcing in and of itself is not necessarily innovative, what the sharing economy adds is a technology platform in which unorganized individuals (the *crowd*) can offer their services, i.e. outsourcing to the crowd or, as coined by Howe (2006), *crowdsourcing*. As a matter of interest, the aforementioned survey conducted by Accenture Interactive (2015) also reported that a significant share of those surveyed who enable geolocation on their mobile devices do so for travel and transportation applications, and that 73% of those surveyed would be willing to receive deliveries from third-parties rather than directly from a retailer.

As with most crowd-sourced activities, reaching critical mass is key to a successful implementation of the concept (Agatz et al., 2012; Klumpp, 2017). Combined with the fact that business-to-consumer (B2C) e-commerce ends at the customer's preferred location (e.g. house, office, trunk of the car), densely populated cities are, therefore, prominent environments where *crowd logistics* may flourish. Crowd logistics should thus be seen as part of a broader web/mobility logistics systems (Goetting and Handover, 2016).

In line with the essential aim of city logistics, i.e., reducing the number of vehicle movements to fulfill freight demands (contributing to reducing greenhouse emissions and enhancing people's life), logistics with the crowd begets manifold opportunities but also challenges. Recently, Buldeo Rai et al. (2017) provided a literature review on crowd logistics initiatives and conducted interviews with practitioners willing to adopt the concept to leverage their business activities. The authors list 18 characteristics defining the broad variety of concepts found, and assess the impact of each on economic, social and environmental sustainability in order to identify the factors that determine the overall sustainability potential of crowd logistics. Those characteristics are classified accordingly to the stakeholders they relate to, namely, receiver and commissioner (either business or consumer), logistic service provider, platform provider and the crowd, and include, among others, involvement of dedicated logistic providers, crowd motivation, and modal choice to perform the services. Most of the literature on crowd logistics limits itself to urban distribution and last mile activities, i.e., so far crowd logistics has been considered intrinsic to city logistics.

Despite growing interest in applications of crowd logistics, few studies exist investigating the many challenges that need to be resolved before a full realization of the concept can be achieved. The objective of this chapter is to provide an overview of applications of crowd-sourced logistics services, and also point out and discuss some of the relevant issues pertaining to the deployment of such innovative systems and their impact and relevance for city logistics.

A related and complementary initiative to city logistics is the *Physical Internet* (Montreuil, 2011), which takes the concepts of the Digital Internet to propose an efficient, sustainable, and resilient logistic infrastructure to move physical objects. In this conceptual vision, freight and people move in the transportation network similarly to how data travels through the Internet. Physical objects are encapsulated in modular packets (π–containers) having unique identifiers that help in the routing, monitoring and traceability, allowing them to follow distinct routes, even if having the same origin and destination, to get to their final delivery. Applied to an urban environment, the Physical Internet underpins what Crainic and Montreuil (2016) define as *Hyperconnected City Logistics*, for which crowdsourcing the transportation activities exemplifies the possible synergy between people mobility and freight logistics.

The remainder of this chapter is organized as follows. City logistics covers a variety of activities, and some of these activities lend themselves well to crowdsourcing. In Section 2.2, we present and discuss not only the obvious ones, e.g., the transport of packages to consumers, but also the less obvious ones, e.g., the receiving of packages on behalf of consumers, in neighborhoods, apartment and office complexes, and the return transport of packages to retailers. In Section 2.3, we highlight and elaborate on aspects that may be of special interest to the transportation science and logistics community, e.g., compensation schemes, supply management, and demand smoothing, and the implications on sustainability. Our perspective on the future of crowd logistics and its role in city logistics is given in Section 2.4. Finally, in Section 2.5, we present concluding remarks.

2.2. CROWDSOURCING LOGISTICS SERVICES

Crowdsourcing initiatives have been considered for a variety of applications, ranging from real-time image search to journalism, from health and medical research to voting. Whereas the term entails the delegation of an activity or process to an independent mass of people (the crowd), there is no commonly agreed upon definition of the concept. A first attempt to integrate many of the existing definitions was made by Estellés-Arolas and González-Ladrón-de Guevara (2012), wherein the authors propose to identify crowdsourcing activities based on aspects such as who forms the crowd, the tasks it has to do, and the incentives it receives for doing them. *Crowd Logistics* concerns crowdsourcing of logistics activities (Mladenow et al., 2015), for example, the delivery of goods to consumers using non-professional drivers who are already on the road and willing to detour to the location of these consumers (*crowdsourced delivery*) or the offering of short-term storage space by non-dedicated third-parties for missed deliveries and later collection (*crowdsourced receiving*).

As with crowdsourcing, there is no agreed upon definition of crowd logistics. Recently, Buldeo Rai et al. (2017) defined the term as "*an information connectivity enabled marketplace concept that matches supply and demand for logistics services with an undefined and external crowd that has free capacity with regards to time and/or space, participates on a voluntary basis, and is compensated accordingly*", which, in our view, indeed captures the essential characteristics of crowd logistics. The authors also argue that crowd logistics fits in the *4 A's of sustainable city distribution* framework proposed by Macharis and Kin (2017) wherein innovative concepts are classified accordingly to Awareness, Avoidance, Act and shift, and Anticipation of new technologies. As such, environmental benefits are envisioned as one of the main benefits of crowd logistics. Crowdsourced delivery, for example, allows for a better utilization of transportation capacity, by fostering consolidation and coordination of existent vehicle flows, offered by the crowd, potentially reducing congestion and greenhouse gas emissions, as it can reduce the number of vehicles dedicated to goods movements.

We note, however, that the "crowd" in crowd logistics refers to a (large number of) independent individuals (participating on a voluntary basis), but that the specific realization of this crowd can have a significant impact on whether or not crowdsourced logistic services contribute to improving city logistics. For example, when existent flows (e.g., existent vehicle movements) are exploited for service fulfillment, this will likely contribute to more sustainable city logistics (Paloheimo et al., 2016; Chen et al., 2017b; Punel and Stathopoulos, 2017). However, in many of the popular platforms for on-demand trans-

Figure 2.1: A schematic view of crowd logistics activities within urban environments.

portation services, either for people mobility (e.g., Uber, Lyft) or for freight delivery (e.g., UberRUSH, Lalamove), in which those seeking a service are matched/connected to independent agents providing that service, fulfillment is realized by creating new service flows rather than exploiting existing ones.

2.2.1. EXAMPLES OF CROWD LOGISTICS

In the following, we discuss a few examples of crowdsourced logistics services. Figure 15.1 illustrates how city logistics might be (re-)organized, and how many activities can be performed by a combination of professional and crowdsourced services. A distribution center (1), on the outskirts of the city, utilizes a professional delivery service employing company vans as well as spare capacity on public transportation, e.g, intercity buses (2) to convey goods to the city. Individuals on their way to work (3) — the crowd — can also participate and handle part of the flow of goods. Within the city, individual packages can also reach final customers by different means. A professional delivery service might exploit crowd-delivery opportunities: it might engage a commuter (4) — an individual — to take a package to an address nearby the commuter's destination. Individuals choosing to live in the city might make deliveries in return for discounts on their bike rentals (5). To streamline the receipt of goods, neighbors may be willing to accept, store, and ultimately deliver a packages (6) — crowd-storage — or large residential buildings may install locker boxes (7), thus reducing the occurrence of missed deliveries.

CROWDSOURCED DELIVERY

Also termed *crowdshipping*, crowdsourced delivery within urban environments is considered as a promising opportunity to accommodate the higher consumer service levels, e.g., same-day or even 2-hour delivery. In this setting, a customer or business (crowdsourcee) uses an ICT platform (crowdsourcer) to place a request for a delivery service to be fulfilled by one of many independent drivers (crowd) registered in the platform. A matching system assigns the request to one driver based on the characteristics of the service (time, destination, capacity) and availability (proximity, en route drivers, detour).

Within crowdsourced delivery, we distinguish two main types of offered services: *door-to-door* and *store-to-door*. Hitch and Roadie are examples of platform providers facilitating door-to-door delivery services wherein travelers (drivers, bikers) pick up and deliver packages for shippers (senders). Hitch allows shippers to post requests for items they want picked up and delivered, and travelers to announce journeys they plan to undertake. Roadie takes the concept a step further and does not require travelers to announce journeys they plan to take, but continuously monitors the movements of its "roadies" and uses machine learning algorithms to recognize travel patterns and automatically identify travelers that can serve requests posted by shippers.

Crowdsourced store-to-door delivery services focus on the B2C market. As an example, e-tailer Zalando relies on Trunkrs to offer same-day-delivery for its customers in certain cities in Europe. Trunkrs uses crowdsourced delivery, but also established courier services. This allows them to provide the reliability demanded by its customers (the e-tailers). Walmart is considering another form of crowdsourced store-to-door delivery services by having in-store customers (the crowd) fulfilling the delivery of items purchased by its on-line clients (Barr and Wohl, 2013).

Other popular store-to-door delivery initiatives are found in the grocery and food service industry, where the platform provider not only arranges the delivery service, but also acts as store front and allows its customers to select the retailer/restaurant from which they want to purchase. Instacart, for example, offers same-day grocery delivery for products bought at grocery stores selected by the customer. This is also typical for meal delivery services, like GrubHub, UberEats and Foodora wherein couriers (drivers or bikers) pick up a meal at the restaurant selected by the customer and deliver it to the customer's home.

Despite the fast growing number of companies offering crowdsourced delivery, literature addressing aspects and issues related to these services is still limited. Paloheimo et al. (2016) conducted a case study in Finland applying crowdsourcing to library deliveries, e.g., books and other media. The study highlights the potential carbon footprint reduction, on average, an equivalent of 1.6 kilometers in spite of the fact that 80% of the deliveries involved trips of less than 5km, and the benefits on leveraging social cohesion that can be achieved with crowdsourced deliveries. Based on the crowdshiping concept envisioned by Walmart, Archetti et al. (2016) introduce the Vehicle Routing Problem with Occasional Drivers (VRPOD). The occasional drivers are the in-store customers willing to deliver an online order for a small compensation. The authors stress the challenges associated with designing an appropriate compensation scheme and the need to continue to employ company drivers to be able to ensure a certain service level. A more in-depth study of this form of crowdsourced delivery is provided in Dayarian and Savelsbergh (2017). We discuss pricing and supply management related issues in more detail in Section 2.3.

Kafle et al. (2017) proposes a two-tiered delivery system, in which the second tier is crowdsourced. In that second tier, cyclists and pedestrians (the crowd) relay parcels from trucks and fulfill the last mile of the delivery. In the system, the carrier/courier posts pickup and delivery requests on a platform and individuals bid to carry out a subset of those requests. Relay points are locations where parcels are transferred between a truck and (one or more) individuals. The company decides on the winning bids and plans the truck routes that visit the relay points and delivery addresses of requests for which no bids were received (or for which the received bids were too expensive). Compared to a pure truck based solution, the system can provide cost reductions (including a reduction in penalty costs associated with late deliveries). The use of transfers points in crowdsourced delivery systems is also considered by Chen et al. (2017b). The authors introduce the Multi-Driver Multi-Parcel Matching Problem (MDMPMP), in which parcels may be transported by a single or by multiple drivers, being transferred between drivers en route to the parcel's end destination in this case. Relaying parcels between drivers allows for a more flexible matching of drivers and parcels, since drivers do not need to fulfill the complete parcel's journey and use transfer opportunities to bring the parcel closer to its end destination. Moreover, since trip duration is mostly important to the driver, as long as the parcel reaches the customer in time, assigning longer paths to the parcels may facilitate the system-wide matching. Similar to Paloheimo et al. (2016), the authors also highlight that without the condition of using pre-existent vehicle flows, crowdsourced delivery operations may induce extra vehicle movements, reducing potential environmental benefits and positive impacts on city logistics.

More recently, some companies are also experimenting with a new method to provide last-mile delivery that would not even require (or require very limited) human intervention. Google and Amazon, for example, are running trials to investigate the use of drones to support package delivery. Usually, these machines can only carry one package at a time, with a maximum weight ($\approx$ 2Kg) and for a limited range ($\approx$ 20Km). Thus, in order to make more efficient use of the capacity and range of a drone, they are deployed from a mobile delivery vehicle. An example can be found in Agatz et al. (2018), where the authors consider the case of one single truck and single drone. With deliveries being conducted simultaneously by both the truck and the drone, not only the total distance travelled by the truck can be reduced but also the total time to service all customers. Crowdsourcing the drone activities is considered by Behroozi and Ma (2020), where the authors consider a combined system of regular delivery trucks and crowdsourced drones. Package delivery is performed by a big truck carrying a large number of packages to a neighborhood or a town in a metropolitan area. The packages are then assigned to crowdsourced delivery agents who operates drones to deliver the packages

2

to their final destinations.

Cargo-hitching

Integration of freight and passenger transport may also play a role in efficient and reliable delivery services, since people and goods may be able to share the same infrastructure for a part of their journey, especially within a city (Trentini et al., 2012; Fatnassi et al., 2015). Cargo-hitching is a realization of this idea and extends the crowdsourced delivery concept by (also) exploiting spare capacity available in public transportation, including tram, metro, buses and taxi service systems, in urban areas for the movement of freight. Van Duin et al. (2019) provides an overview of projects considering the integration of passenger and freight transport, including examples combining bus, bike, train and trams services. The authors also provide insights on understanding how to organize viability for cargo hitching projects, as a concept providing environmental and social benefits while at the same time providing a sustainable business model.

Long-haul implementations of cargo-hitching have existed for many years in the airline and railroad industry. Short-haul implementations, however, are less common. PostBus Courier (DHL, 2015) is a DHL service integrating parcel transport and passenger service on its long-distance intercity bus network. Initially, in 2015, the service was offered between Berlin and Hamburg, particularly for same-day, urgent, shippings for both B2C and B2B customers. In a fully integrated system, however, different stakeholders may be involved (Jesus Gonzalez-Feliu and Routhier, 2014; Arvidsson et al., 2016), e.g., a logistics service provider who leases (spare) capacity on buses from a city bus operator. Coordination and synchronization are challenging in such environments and only a limited number of research efforts exploring these issues have been reported in the literature.

Masson et al. (2017) propose a Mixed Urban Transportation Problem consisting of two tiers for the distribution of goods within cities. In the first tier, city buses are used to transport goods from distribution centers to a set of bus stops and then, in the second tier, goods are transferred to city freight distributors to be delivered to the end customer. Ghilas et al. (2016) consider the feasibility and opportunity of incorporating scheduled public transportation in the distribution of goods. Pickup and Delivery (PD) vehicles are used to bring (collect) goods to (from) a bus station, and spare capacity on the scheduled bus services, which can be high, especially in off-peak hours, is used to move goods for part of their journey to their end destination.

Whereas buses and other public transport modes operate on predetermined routes and schedules, taxis are more flexible as passengers determine pickup and delivery locations as well as times. Thus, taxis may be used, at times, to move freight within the city on an individual on-demand basis. Li et al. (2014) introduce and explore the Share-a-Ride Problem, which is an extension of the Dial-a-Ride-Problem (Cordeau and Laporte, 2007), but taking into account the different requirements to transport people and freight using a taxi network (e.g., maximum ride-time, detours, number of stops, etc.). Taxis are allowed to deliver parcels as long as the service level for the passenger does not deteriorate significantly. A Freight Insertion Problem (FIP) is proposed to insert parcel collections in a given routing plan for passengers aimed at minimal passenger disruptions. Chen and Pan (2016) specifically refers to a "crowd of taxis" to propose, in the same vein as the solution for reverse flows in Chen et al. (2017a), using a taxi fleet in the city in tandem with a network of 24/7 shops to satisfy last-mile delivery requests.

2.2.2. BEYOND TRADITIONAL CROWD LOGISTICS

Crowd logistics, up to now, has been seen mostly as an opportunity to reduce the cost and the speed of delivery in the urban distribution of goods, in particular for the fulfillment of home deliveries. In the future, crowd logistics will likely cover a wider range of city logistics functions. We discuss some in this section.

RECEIVING PACKAGES

For home delivery, an important aspect of the fulfillment process is the actual receiving of the package. Failing to deliver because a recipient is not at home to accept (and, possibly, sign for) the package will not only disappoint consumers, but will also result in extra costs, because courier companies usually retry delivery (a few times). To prevent missed deliveries, alternative delivery options have been introduced, e.g., customers are offered (convenient) locations to collect parcels, ranging from strategically located locker boxes (e.g., at subway or train stations) to pickup points at gas stations and convenience stores. Locker box solutions for apartment blocks and other housing complexes are offered, for example, by Amazon (The Hub). Such solutions not only help in reducing missed deliveries, but also in increasing service efficiency. Due to the large number of residents in apartment complexes, door-to-door delivery of every package can be quite time consuming. Moreover, the number of packages delivered in these residential buildings has been increasing as fast as e-commerce (17%) for the last three years (Rodrigue, 2017).

At such pace, even those alternative collection services, e.g., locker boxes, will soon reach the limit of their usefulness. Moreover, installing locker boxes is expensive and, even though unattended delivery can be mitigated, the problem is not entirely solved, since parcels may be kept in the locker for several days. The use of crowdsourced solutions might provide a better means to alleviate the situation.

Wang et al. (2016) propose a last-mile fulfillment system in which the delivery of a parcel, from a locker/pickup station to the end consumer, is crowdsourced to a pool of citizen workers. Compared to a self-collection model, having the crowd collect and deliver parcels from locker boxes can reduced costs, since potentially fewer locker boxes are required and the turn-over rate can be increased. Experiments using datasets from Singapore and Beijing show that the approach can be used in large-scale settings (with a huge number of workers and parcels to be collected and delivered).

Another crowdsourced solution is neighbor or neighborhood delivery, where an individual uses his/her available time and space (capacity) at home to receive and temporarily store packages. This is a possibly attractive proposition for elderly residents in a neighborhood, both from a social and economic perspective. In Europe, such service is offered by DHL and PostNL as long as the sender does not require strict delivery to the customer (signed delivery). Even though companies do not offer any compensation for the person receiving the package, such approaches not only mitigate the negative effects of unsuccessful deliveries but also can help in building a sense of community around the neighborhood.

RETURNING PACKAGES

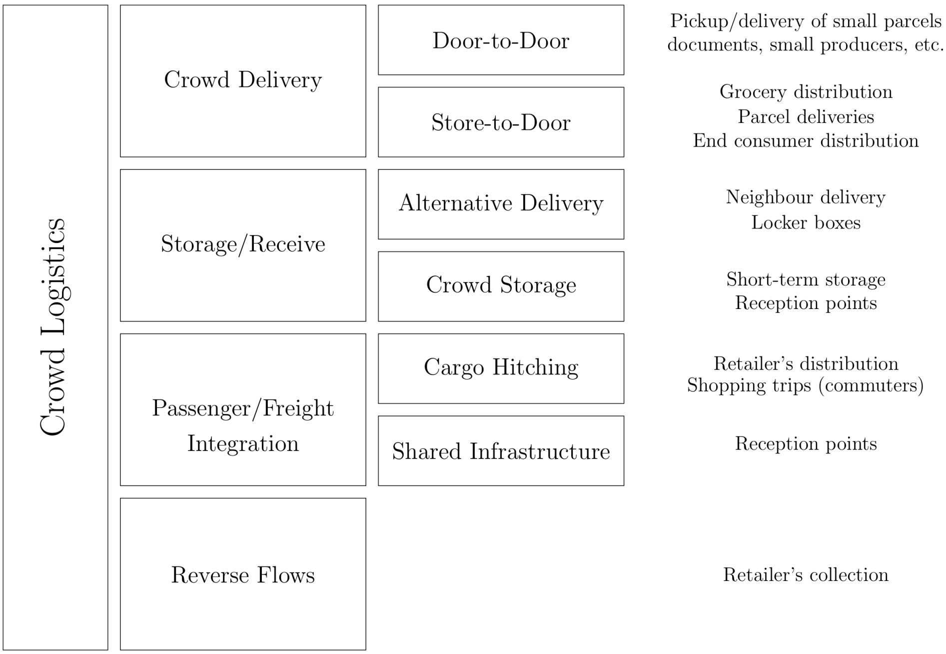
The continuous growth of on-line and e-commerce sales has also led to an increase in returns. To increase on-line sales, especially in the apparel industry, offering free returns has been not only essential, but has also become the norm. In Europe, customers have the right to return items within 14 days, for any reason, and get a full refund. Free returns, of course, are not free for the retailer and the need for more cost-effective ways to manage reverse flows has become obvious (de Brito and Dekker, 2004). Figures from 2015 showed that 30% of goods purchased on-line are returned (Reagan, 2016). On specific markets, as apparel and shoes, that rate can be even higher. Moreover, customers are more willing to buy when offered free returns (United Parcel Service of America, 2015) and “buy anywhere return anywhere" policies have contributed to omni-channel strategies that shortens the supply chain towards the customer but the impact of such strategies on city logistic is not yet clear (Savelsbergh and Van Woensel, 2016).

Thus, not only increased direct-to-consumer deliveries pose new challenges for city logistics, but also the increased rate of returns. However, integrating the fulfillment of both flows is not straightforward as an item might be returned to the same distribution center where it came from, to another distribution center, or to a store (omni-channel solutions), for example, and might require different handling. Nonetheless, and despite being more time flexible than the last-mile delivery, an efficient return process is beneficial for both customers and companies. For the former, a fast return may result in an earlier refund deposit or a new product delivered. For the latter, it impacts the possible reselling of the returned item.

Crowdsourcing reverse flow activities is, therefore, a promising option for companies and could also contribute to mitigate the negative effects of extra vehicle movements within a city to fulfill returns. Yet, real cases of such solutions are not known to the authors. Research on feasibility issues, though, has already begun, but is limited. Chen et al. (2017a) propose to use shops (since they provide flexible delivery and pickup times, and are more convenient to customer and drivers) to build a collection network for returned goods in which en-route taxi services are used to collect packages at shops, before picking up the passenger, or to deliver packages at the shops, after dropping off the passenger. Different collection strategies are used to dispatch the taxis to transport goods from shops to the distribution centers, exploiting the extra capacity for small parcel on taxis, thus diminishing the carbon footprint to fulfill the service compared to using a truck for the same purpose.

Whereas crowdsourced receiving solutions (e.g., neighborhood delivery) are mostly applied to support last-mile deliveries, returned goods might also be stored by the crowd, facilitating collection operations for companies.

Figure 2.2: Example of crowdsourced activities within city logistics.



An overview of the city logistic applications discussed in the previous sections, with examples of crowdsourcing services, is given in Figure 2.2.

2.3. RESEARCH OPPORTUNITIES IN CROWD LOGISTICS

In this section, we discuss a few topics that are not only relevant to crowd logistics, but also provide, in our view, exciting research opportunities.

The rise of the sharing economy makes it possible to monetize goods and services not deemed as assets before (Geron, 2013). As a consequence, new models have emerged based on the (temporary) access to rather than the ownership of (expensive) assets. In the context of logistics, this represents a paradigm shift from traditional models with a focus on optimizing asset ownership for a given activity (DHL Trend Research, 2017). The adoption of crowd logistics follows a transition away from the traditional schemes, in which a company owns assets and employs workers to perform its logistic activities, or outsources its logistic activities to third-party providers. More and more, mixed schemes are developed in which a company reduces its owned assets (e.g. vehicles and workers) to perform its logistic activities, and relies, instead, on crowd logistics for some of these activities (and, potentially, even all of its logistics activities).

This paradigm shifts necessitates more research on a number of relevant topics. Clearly, for these time being, crowd logistic raises more questions than answers. Research addressing these questions is still in its infancy.

Below, we discuss the following research topics: **Consolidation using existing flows**, which avoids the need for additional resources to be put on the heavily congested infrastructure, thus leading to more sustainable logistics services. **Willingness to participate**, which is critical to the success of a crowd logistics market, and involves both the supply of capacity and the demand for capacity in the market. **Scale and dynamics**, which, for crowd logistics, are significantly different from more traditional logistics services, as the number of participants (both on the supply and the demand side) tends to be much larger and their entry and exit faster and less predictable.

2.3.1. CONSOLIDATION USING OF EXISTING FLOWS

Consolidation, coordination and cooperation are fundamental to city logistics, and central for achieving an integrated system in which freight movements are performed as efficiently as possible e.g., by minimizing the fleet size, reducing empty traveling. Most of the literature considers the availability (capacity and time) of *preexistent* flows as a central aspect to crowd logistics.

Clearly, independent agents, not necessarily already performing another duty (e.g., a commuting driver), can be considered as part of a crowd-logistics solution. However, Paloheimo et al. (2016) also point out that rebound effects, where drivers travel longer distances, particularly motivated by monetary compensation, can reduce the potential environmental effects. Additionally, Chen et al. (2017b) mention that crowdsourcing activities not using pre-existing flows generate new movements, reducing the overall impact on sustainability.

One activity that best exemplifies the aforementioned issues, in the context of people transportation, is ride-sharing (Kamar and Horvitz, 2009; Agatz et al., 2012; Furuhata et al., 2013), in which drivers offer empty seats to other travelers with similar itineraries and time schedules. Those arrangements benefit not only the driver (sharing the costs of owning and maintaining a car), but also the passengers, since sharing a ride can be less costly and more convenient than using traditional forms of transportation. Moreover, ride-sharing also has an impact on the efficiency of urban transportation: potentially less vehicles are used to provide the required mobility, traffic congestion can be reduced, as well as fuel consumption and greenhouse gas emissions.

BlaBlaCar is a platform providing ride-sharing support, connecting drivers with empty seats to interested passengers. Recently, Uber started to offer a new service, UberPOOL, wherein drivers can announce their journeys and are matched with riders heading in the same direction. Perhaps due to the collaborative aspects of these sharing platforms, non-economic benefits such as improving the environment and social welfare may be regarded as primary objectives of such services.

2.3.2. Willingness to participate

In a study investigating the motivations why people participate in sharing economy activities, Hamari et al. (2016) highlight that while sustainability aspects influence how collaborative consumption is perceived, participants are mostly motivated by economic benefits. Nevertheless, other factors such as enjoyment in performing the activity and social awareness are also important. This is in accordance with the results reported in Paloheimo et al. (2016) for the case study of a crowd-delivery pilot for a library, where monetary compensation, while important, was not the main driver for participation.

An important characteristic of crowd logistics services is that they are offered by independent providers on a voluntary basis i.e., there is no employee-employer agreement between the company (platform owner) and the crowd. Relying for all or part of your logistics activities on the crowd is thus a non-trivial strategical/tactical decision that has major implications. In the context of crowd delivery, professional drivers are more expensive, but are available when required and do what you need, thus providing certainty and reliability. Independent drivers are less expensive, but are only available when it is convenient for them and perform tasks that they deem beneficial, leading to uncertainty and, potentially, a significant loss of service quality. Additionally, a characteristic of on-demand services is that customers are sensitive to price and waiting time (Tang et al., 2016; Taylor, 2017). The availability of independent drivers may be a concern as well as the willingness of independent drivers to perform certain transportation requests. To ensure reliability and quality of service towards its customers, a company may still have to rely on (some) professional drivers (either company employees or third-party drivers).

A key mechanism to manage this capacity uncertainty is the compensation scheme utilized for the independent providers (crowd). Regardless of whether the crowd is driven by altruistic, non-monetary motivations or by the possible economic gains that can be achieved, an efficient compensation scheme is crucial for attracting participants. These can include both monetary and non-monetary incentives. On the platform side, relying on independent providers to fulfill real-time requests is challenging since the providers decide whether and when to work and this decision is driven by the offered compensation. Few providers implies that customers will have to wait more to be serviced which, in turn, will decrease customer satisfaction and demand. The platform has to choose an appropriate compensation level for providers, in some cases dynamically, given the available number of providers and customers. So far, this problem has only been modeled using concepts from queueing theory (Tang et al., 2016) but could also be framed using concepts of cooperative and non-cooperative game-theory. More research is required to better understand how to make these trade-offs, which will have to involve modeling the behavior of independent providers.

Economy of scales dictate costs and pricing strategies for professional logistic services providers, e.g., consolidating large amounts of low-valued small activities. Due to the nature of the services offered by the crowd (e.g., spare capacity on free time), crowd services tend to be more personalized and, thus, more fragmented. As an example, crowd-delivery services operate on a parcel level (e.g., the driver/agent only performs one delivery) and do not take advantage of consolidation. Setting appropriate prices should take into account a large number of small and low-valued single activities performed by different agents (Klumpp, 2017). Other pricing mechanisms, such as the bidding system proposed in Kafle et al. (2017), could be leveraged to address such issues. Operational planning for horizontal cooperations between road transportation carriers often is performed through auction-based mechanisms in which requests are exchanged among carriers (Song and Regan, 2003; Verdonck et al., 2013). In a crowd-delivery context, bidding mechanisms could allow for a better assessment for which compensation is considered appropriate for a crowd agent and, also, help the crowd logistic platform in deciding which activities to crowdsource. Furthermore, bidding mechanisms stimulate the use of existing flows or space instead of generating new traffic and/or capacity. For example an independent driver already heading to a certain delivery location has lower marginal cost and effort and thus can ask for a smaller compensation than someone who especially has to drive there.

2.3.3. SCALE AND DYNAMICS

As a highly interconnected and interdependent environment, information regarding different aspects of the city changes constantly. Congestion, for example, might have a significant impact on travel times. For delivery services, new requests might arrive after the route planning for the vehicle has been decided. For a survey on the inherent issues and challenges posed by vehicle routing optimization in city logistics contexts, the reader is referred to Cattaruzza et al. (2015). One of these challenges, in particular, is especially relevant for the crowd-sourcing of transportation activities, namely, how to handle dynamic incoming information to (re-)optimize decisions already made taking into account the new information. In the context of crowd logistics, since participation is voluntary, real-time information regarding crowd availability, for example, will be crucial for successful implementations.

To date, only few works have considered crowd logistic approaches taking into account dynamic information. Li et al. (2014) extended the SARP to consider dynamic scenarios in which passenger requests are accepted or denied in real-time (at the time of the call), but parcel demands are known (pickup and delivery locations and time windows) beforehand. Routes for the accepted passengers are generated and feasible parcels are inserted by solving an associated FIP. Arslan et al. (2019) introduces a variant of the dynamic pickup and delivery problem in which ad-hoc drivers (occasional drivers) are willing to make a small detour in exchange for a small compensation to improve on-demand delivery. Both ad-hoc drivers and delivery requests arrive in real-time and the crowdsourcing platform has to assign delivery tasks to drivers. Real-time information is handled in a rolling horizon framework that re-optimizes the system whenever a new request or driver is available. A simulation study is conducted to evaluate the feasibility of the concept, based on data collected from the city of Rotterdam, the Netherlands. Dayarian and Savelsbergh (2017) consider the dynamics of employing in-store customers to deliver online orders (as envisioned by Walmart).

2.4. BEYOND CITY LOGISTICS

The focus of the chapter is on crowdsourcing opportunities in city logistics. In this section, we briefly discuss crowdsourcing opportunities in inter-city logistics, which, of course, may impact city logistics as well.

Whereas the rise of car-sharing initiatives may in the future provide a valid, and possibly attractive, alternative to renting cars, P2P car-sharing, at the moment, still represents only a small share of the market. Regardless, a critical issue when supporting one-way renting (or one-way sharing) is repositioning. Rental car companies incur significant costs in the repositioning of cars in order to re-balance fleet availability among their rental counters. Such a re-balancing operation is either performed by trucks or by dedicated drivers. To reduce such repositioning costs, Transfercar, among others, initiated a service that connects rental car companies having cars to be repositioned, and drivers willing to perform the repositioning task: *crowdsourced repositioning*. However, a seemingly missed opportunity is to use the capacity created by the repositioning of vehicles for the movement of freight.

Inter-city flows, (e.g., the repositioning of a rental car from one city to another) while not having a direct impact on city logistics, still can influence the goods distribution inside the city. We provide two such examples in airline transportation. Airmule is a platform wherein air travelers flying with spare luggage allowance can register themselves as on-board couriers, offering a transport service from their departure to their arrival airports. Under the promise of improved access to products not available locally, Grabr is a door-to-door crowd-sourced delivery service but, differently from Airmule, the crowd also takes care of the purchase. The requester posts a solicitation for a particular item to be purchased and travelers post offers for servicing such request (if they can buy the product while traveling and if they are willing to deliver the purchase upon their return). After the requester accepts an offer, the platform handles associated financial transactions, i.e., the traveler gets paid after the delivery has been confirmed.

Moreover, cities infrastructure in the future will more and more make use of technological advances such as sensors and other types of electronic data collection. Their use will not be limited to transportation systems, but will include the numerous other systems in urban areas (e.g., waste management, law enforcement), all of them integrated on a network to optimize the overall city performance. This will call for new strategies to extract information from the infrastructure and advanced data analytics methods to make sense of the data and use it to support (optimized) decision making. Finally, in these new scenarios, citizenship participation will be a central aspect.

2.5. FINAL REMARKS

Crowd logistics, i.e., involving the crowd in freight related activities, is one of the strategies that may help achieve the goals of city logistic. Despite its significant potential, it is far from obvious how to best use the crowd for logistic services, from an economic, societal and environmental point of view. There is no commonly accepted definition of crowd logistics and of who constitutes the crowd, and yet crowd logistics, one way or another, will play an important role in city logistics.

To be able to achieve the objectives of city logistics i.e., reducing the number and improving the efficiency of freight movements within the city, it is necessary for everyone using the already stressed urban infrastructure and for everyone impacted by urban freight transport to come together (Jesus Gonzalez-Feliu and Routhier, 2014; Bektaș et al., 2015). The challenge is that these objectives can no longer be achieved by investing in extra capacity. There is too little space and the costs are prohibitive (Savelsbergh and Van Woensel, 2016). New strategies to organize and control freight movements within cities are required.

The goal of this chapter was to provide an overview of crowdsourcing solutions in transportation and logistics, from natural applications in (home) delivery to less obvious applications in receiving and returning of goods, and to highlight opportunities for interesting and high impact research. We hope that our perspective will stimulate and encourage others to seek creative and innovative solutions to the challenges of city logistics as more and more of us will be living in larger and larger cities all around the world.

3

Delivery Systems with Crowd-Sourced Drivers: A Pickup and Delivery Problem with Transfers

Fall in love with the problem, not the solution.

Uri Levine, startup co-founder

O problema são problemas demais
se não correr atrás da maneira certa de solucionar[1]

Chico Science, Samba de Lado

Rapid urban growth, the increasing importance of e-commerce and high consumer service expectations have given rise to new and innovative models for freight delivery within urban environments. *Crowdsourced* solutions – where drivers are not employed by a carrier but occasionally offer their services through on-line platforms and are contracted as required by carriers – are receiving growing attention from industry. We consider a crowdsourced system where drivers express their availability to perform delivery tasks for a given period of time and the platform communicates a schedule with requests to serve. We investigate the potential benefits of introducing transfers to support driver activities. At transfer locations, drivers can drop off packages for pick up by

[1] The problem becomes too many problems if you do not look after the right way to solve it.
The work in this chapter is published in Sampaio et al. (2020)

other drivers at a later time. We frame the problem as a Multi-Depot Pickup and Delivery Problem with Time Windows and Transfers, and propose an Adaptive Large Neighborhood Search algorithm that effectively identifies beneficial transfer opportunities and synchronizes driver operations. Computational experiments indicate that introducing transfer options can significantly reduce system-wide travel distance as well as the number of drivers required to serve a given set of requests, especially when drivers have short availability and requests have high service requirements.

3.1. INTRODUCTION

Delivering goods in urban areas is one of the most challenging logistics activities. And given the continuing urbanization and the increasing share of e-commerce in retail sales, it will only become more challenging in the future. Therefore, creative ideas and innovative concepts are needed and have to be explored. Among the most promising, so far, is *crowdshipping*, where transportation capacity is provided by individuals willing to provide their time and their vehicle for a (short) period of time. The use of crowdshipping to improve B2C delivery and to support high consumer service levels is being explored by various companies. As an example, e-tailer Zalando relies on Trunkrs to offer same-day delivery for its customers in certain cities in Europe. Trunkrs uses crowdsourced delivery, but also established courier services to provide the reliability demanded by its customers (the e-tailers). For an overview of crowded-based applications in transportation and logistics, see Buldeo Rai et al. (2017); Sampaio et al. (2018).

The primary challenge associated with the use of crowd-sourced transportation capacity is that the capacity is provided by individuals who are not under contract and incentive schemes have to be used to secure capacity when and where it is needed. One option is to dynamically adjust compensation based on the required capacity, i.e., increase compensation at times when and at places where more capacity is needed. Another option, the one we assume is used in our research, is to ask individuals to commit to work for a specific period of time, a block, in return for a minimum pay assurance. Such a scheme is employed, for example, by Amazon Flex and Grubhub (a meal delivery platform).

Because blocks tend to be relatively short, as this provides flexibility both for the company and for the individuals, we consider a system in which goods to be delivered can be transferred, i.e., be taken from their origin to their destination by more than one driver. This allows a company to make use of available crowd-sourced capacity to handle short-distance as well as long-distance transportation tasks.

To model the proposed crowd-sourced delivery system, we consider a Pickup and Delivery Problem with Time Windows and Transfers (PDPTW-T). At transfer locations, drivers can drop off packages for pick up by other drivers at a later time. The number and location of transfer points are strategic decisions and not considered in our study. A transfer location can be situated at accessible locations – facilitating the operations between drivers – such as gas stations, stores and supermarkets. It can also be an automated facility, such as a locker station, situated in one of those locations. Characterizing delivery settings that might benefit from the introduction of transfers has yet to be studied (to the best of our knowledge, this is only investigated by Mitrović and Laporte (2006)).

When the transfer of requests is not an option, the problem is a Pickup and Delivery Problem with Time Windows (PDPTW) and defined as follows. A fleet of vehicles is available to satisfy a set of transportation requests, each defined by a load to be transported (goods or people) by a single vehicle from a pickup location (origin) to a delivery location (destination) where the pickup has to occur after a given time and delivery has to take place before a given time. Multiple requests can be served by the same vehicle, as long as the vehicle capacity is never exceeded. When all requests are known in advance, the problem is referred as the static PDPTW; when requests arrive during the execution of planned routes and routes have to be updated to accommodate new requests, the problem is referred to as a dynamic PDPTW. The objective is to determine a set of vehicle routes serving all requests, where the function that drives the costs is chosen according to the application at hand (e.g., distance, duration, number of vehicles, or pollution Savelsbergh and Sol (1995)). The reader is referred to Berbeglia et al. (2010) for a survey on dynamic pickup and delivery problems.

As mentioned above, transfer points are locations in the network where requests can be transferred between vehicles and temporarily stored. Hence, more than one vehicle can be used to serve a single request, e.g., a request may be picked up at its origin by one vehicle, then dropped at a transfer point where another vehicle (potentially with other characteristics) picks it up and delivers it at its destination. Note that direct transportation from the pick up location to the delivery location by the same vehicle is still possible. Transfer points allow, among others, service by a mixed fleet of vehicles ((electric) truck, van, or bike), but also allow integration of freight and passenger transportation. As any pickup and delivery problem can be seen as a special case of a pickup and delivery problem with transfers, pickup and delivery problems with transfers are NP-Hard. A critical challenge when solving pickup and delivery problems with transfers is synchronization. Whereas allowing goods to be temporarily stored (or people to wait) at an intermediate location provides more flexibility and, thus, may allow more effective use of resources, these resources need to be carefully synchronized in order to obtain a feasible solution.

Our contributions can be summarized as follows. We analyze the potential benefits of transfers in pickup and delivery operations in urban areas, focusing specifically on settings in which drivers operate short shifts (as is likely to happen in crowdshipping settings models). In such settings, the flexibility provided by transfers may allow serving long-distance requests that would otherwise be impossible. We propose a heuristic for the solution of instances of the multiple-depot PDPTW-T. The heuristic produces high-quality solutions in a reasonable amount of time. We compare the performance of the heuristic to that of a state-of-the art heuristic for the (multiple-depot) PDPTW. The main limitation of our approach, at this point in time, is that it assumes that all information is available at the start of the planning horizon. In the future, we plan to extend our technology to a dynamic setting, where information arrives over time during the planning horizon. Nevertheless, we believe that our current analysis already provides relevant and informative insights into the potential benefits of transfers for transportation systems relying, in part, on crowd-sourced capacity.

The remainder of the paper is organized as follows. The relevant literature and related problems are discussed in Section 3.2. The notation used throughout the text is introduced in Section 3.3, as well as a formal problem definition of the problem and a mathematical programming formulation. The heuristic, an Adaptive Large Neighborhood Search heuristic, is introduced in Section 3.4, and the results of an extensive computational study are presented in Section 3.5. Finally, we present conclusions and directions for further research in Section 3.6.

3.2. RELATED LITERATURE

The literature considering the use of transfer points in pickup and delivery problems can be divided in two groups: (1) pickup and delivery problems with transshipment (PDP-T) and (2) pickup and delivery with cross-docking (PDP-CD). Whereas both, transfers and cross-docks, act similarly in providing a consolidation mechanism in which short term storage for a limited amount of requests allows vehicles to potentially improve their loading plans, a few differences are worth mentioning. First, in a cross-docking system transportation requests are executed by a set of separate pickup and delivery routes. Inbound vehicles collect the items, bring them to the CD where they are consolidated, and then a set of outbound vehicles delivers them to their final destinations (Maknoon and Laporte (2017)). Transfer locations, on the other hand, can be seen as a possible transfer opportunity i.e., a request can be carried out either through transfers and, in this case, one vehicle collects the load and another one delivers it to its final destination, or the request can be served directly by the same vehicle. Second, the CD is the start and end location for each route in the plan, whereas a transfer can be any location where vehicles can exchange loads throughout their routes. Third, in some CD applications, consolidation activities can only be executed when all vehicles are in the CD. In the PDP-T, this is not required as long as synchronization requirements are met.

The static (i.e., all transportation requests are known in advance of the optimization) PDP variants have given rise to a substantial amount of research Berbeglia et al. (2007). Only recently the possibility of allowing transfers of goods (or people) is addressed in the literature. Mitrović and Laporte (2006) proposed a two-phase heuristic to solve the PDP-T consisting of a construction phase, in which a start solution is obtained, and an improvement phase defined by iteratively removing and inserting requests in a candidate solution. Nakao and Nagamochi (2008) consider the PDP-T, without time windows on the requests, and analyze lower bounds for the case where one transfer location is available and vehicles are allowed to visit the transfer at most once. The authors show that $z(PDP) < (6\lceil\sqrt{k}\rceil+1)\cdot z(PDPT)$, where k is either the number of routes in an optimal solution of the PDP-T or the number of requests, $z(PDP)$ and $z(PDPT)$ are the optimal travel costs for the PDP and the PDP-T, respectively. A first mathematical formulation and a branch-and-cut approach were proposed in Cortés et al. (2010), but only small instances could be solved. Motivated by an air cargo carrier application, Qu and Bard (2012) introduced an insertion heuristic to identify profitable circumstances to exploit the transshipment option. The authors developed a GRASP algorithm and proposed a set of randomly generated instances. More recently, Rais et al. (2014) proposed a new model for the problem, distinguishing between vehicle (routes) and request flows and using multi-commodity flows to match these two. Masson et al. (2013) proposed an adaptive large neighborhood (ALNS) algorithm for the problem, and tackled the Dial-a-Ride Problem (DARP) with transfers in Masson et al. (2014). A similar problem was proposed by Ghilas et al. (2016), where requests are allowed to be transferred to/from scheduled lines such as bus, train and metro, operating between two terminals. In a recent survey from Guastaroba et al. (2016), pickup and delivery problems with transfers are mentioned as one of the extensions of the PDP with cross-docking.

An increasing number of new start-ups within the shared-economy context, exploring different ways of monetizing underused assets e.g., cars, rooms, is giving rise to new strategic and operational issues that have to be addressed in order to support such systems. It is only recently that crowdsourced delivery systems have received academic attention. Based on the *crowdshipping* concept envisioned by Walmart, Archetti et al. (2016) introduce the Vehicle Routing Problem with Occasional Drivers (VRPOD). Occasional drivers are in-store customers willing to fulfill the delivery for an online customer on their journey (performing a small detour, if necessary) after leaving the store. The authors highlight the challenges associated with designing appropriate compensation mechanisms and the importance of employing company drivers in order to ensure a certain service level. Dayarian and Savelsbergh (2017) explore using in-store customers to supplement company drivers to deliver dynamic incoming, on-line customer orders. Dahle et al. (2019) build on the VRPOD and introduce the Pickup and Delivery Problem with Time Windows and Occasional Drivers (PDPTW-OD). The authors propose compensation schemes for the occasional drivers and show that cost savings of 10–15% can be achieved even when the company utilizes a sub-optimal scheme.

Behrend and Meisel (2018) investigate a form of *crowdshipping*, in which private drivers offer to execute delivery jobs for other people on trips they would make anyway, within item-sharing contexts, where assets e.g. tools, leisure equipment, can be temporarily rented. Whereas the prevalent practice on item-sharing platforms is that the transportation of an asset is delegated to the consumer (either the consumer actually performs the task or hire a courier company), the authors show that crowdshipping can increase profits and service levels for an item-sharing platform. Arslan et al. (2019) consider a service platform that dynamically creates matches between parcel delivery tasks and private drivers. Since the platform is only able to control the available number of drivers, a regular, dedicated vehicle fleet is also operated by the platform. The authors introduce a new variant of the dynamic pickup and delivery problem, and propose a rolling horizon framework for its solution. Results show that, compared to traditional delivery systems, crowdshipping systems can provide savings of up to 37%.

In another realization of the *crowdshipping* concept, similar to the approach considered in this work, drivers willing to perform transportation services use an online platform (Uber-Freight, Amazon Flex) and are matched to demand for such services in real-time. The drivers are independent, work for a given period of time and are paid in a hourly basis. To the best of our knowledge, there is no work dealing specifically with such scenarios. Nevertheless, in a similar context, taxi drivers might be willing, at times, to move freight within the city. Li et al. (2014) introduce and explore the Share-a-Ride Problem taking into account different requirements to transport people and freight using a taxi network. Taxis are allowed to deliver parcels as long as the service level for the passenger does not deteriorate significantly. A Freight Insertion Problem (FIP) is proposed to insert parcel collections in a given routing plan for passengers aimed at minimal passenger disruptions (e.g., maximum ride-time for passengers, maximum detours, number of stops). Chen and Pan (2016) refers to a “crowd of taxis” to propose, in the same vein as the solution for reverse flows in Chen et al. (2017a), using the taxi fleet and a network of 24/7 shops to satisfy last-mile delivery requests.

The use of intermediate locations in crowdsourced delivery systems, as an interface between company drivers and the crowd, is invstigated by Kafle et al. (2017). The authors propose a two-tiered delivery system, in which the second tier is crowdsourced to cyclists and pedestrians (the crowd). In the system, a truck carrier posts pickup and delivery requests on a platform and individuals in the crowd bid to carry out a subset of those requests. Parcels can be transferred between a truck and (one or more) individuals at relay locations. The company decides on the winning bids and plans the truck routes that visit the relay points and delivery addresses of requests for which no bids were received or were too expensive. Chen et al. (2017b) introduce the the Multi-Driver Multi-Parcel Matching Problem (MDMPMP), in which parcels may be transported by a single or by multiple drivers, using existing planned routes of the drivers. Parcels can be transferred between drivers, allowing for a more flexible matching of drivers and parcels, since drivers do not need to fulfill the complete parcel’s journey and use transfer opportunities to bring the parcel closer to its final destination.

3.3. PROBLEM DESCRIPTION

The Pickup and Delivery Problem with Time Windows and Transfers (PDPTW-T) is defined as follows. We are given a set of transportation requests $R = \{(i^+, i^-) | \ i^+ \in P, i^- \in D\}$, where P and D are the set of pickup and delivery locations, respectively, and where a request $r_i = (i^+, i^-)$ concerns the pickup of a load at i^+ and its delivery at i^- within time window $[E_{i^+}, L_{i^-}]$, specifying the earliest time, E_{i^+}, when the load is available at the origin and the latest time, L_{i^-}, when the load must be delivered at the destination. A fleet of vehicles, V, is available to serve the requests. Each vehicle $v \in V$ starts and ends its shift at a depot located at $m_v \in M$, with M the set of all depot locations. Each vehicle $v \in V$ has a duty period, or shift, $[E_v, L_v]$; the vehicle can depart its depot at E_v and has to return to its depot by L_v. There is a set Γ of transfer locations. A request $r_i \in R$ can be served by one vehicle visiting both i^+ and i^-, or by two vehicles with one vehicle visiting i^+ and the other visiting i^-. In the latter case, the request is transferred at a location $t \in \Gamma$. Note that we allow only a single transfer. Two vehicles serving a request are not required to be at the transfer location at the same time, i.e., transfer locations offer short-term storage (a vehicle can drop off a request and leave, while another vehicle can come and pick up the request at a later time). Multiple vehicles can visit a transfer location at the same time and vehicles can wait at a transfer location. We focus on settings where the number of drivers is large and shift durations are short (relative to the planning horizon) – which is representative of crowdshipping settings. We also assume, for simplicity of exposition, that the size of loads is small compared to the vehicle capacity, and that the vehicle capacity is never restricting.

3.3.1. MATHEMATICAL FORMULATION

The PDPTW-T can be modeled on a directed graph $G(W, A)$ with node set $W = P \cup D \cup M \cup \Gamma$ and $A = \{(i, j) \mid i, j \in W, i \neq j\}$. It takes a vehicle $\tau_{ij} > 0$ units of time to travel from i to j, $(i, j) \in A$, and incurs a cost $c_{ij} > 0$. We assume travel times satisfy the triangle inequality. The objective is to find a minimum cost set of vehicle routes serving all requests (either by one vehicle or by two vehicles). Rais et al. (2014) propose a flow-based mixed integer programming formulation for the PDPTW-T, in which request flows are linked with vehicle flows. The vehicle flow is modeled by binary variables x_{ij}^v, indicating the traversing of arc $(i, j) \in A$ by vehicle v. Request flows are modeled by binary variables y_{ij}^{vr} indicating whether vehicle $v \in V$ carries request $r \in R$ over arc $(i, j) \in A$. Synchronization of transfer operations is achieved through variables a_i^v and d_i^v, for arrival and departure times of vehicle v at location i, respectively, and binary variables s_{tr}^{vw} indicating whether request r is transferred between vehicles v and w at transfer location t. The formulation is given below

$$\min \sum_{v \in V, (i,j) \in A} c_{ij} x_{ij}^v \tag{3.1}$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij}^v \leq 1 \qquad \forall v \in V, i = m_v \tag{3.2}$$

$$\sum_{(i,j) \in A} x_{ij}^v = \sum_{(j,i) \in A} x_{ji}^v \qquad \forall v \in V, i = m_v \tag{3.3}$$

$$\sum_{(i,j) \in A} x_{ij}^v - \sum_{(j,i) \in A} x_{ji}^v = 0 \qquad \forall v \in V, \forall i \in W \setminus M \tag{3.4}$$

$$\sum_{v \in V} \sum_{(i^+,j) \in A} y_{i^+j}^{vr} = 1 \qquad \forall r \in R, r = (i^+, i^-) \tag{3.5}$$

$$\sum_{v \in V} \sum_{(j,i^-) \in A} y_{ji^-}^{vr} = 1 \qquad \forall r \in R, r = (i^+, i^-) \tag{3.6}$$

$$\sum_{v \in V} \sum_{(t,j) \in A} y_{tj}^{vr} - \sum_{v \in V} \sum_{(j,t) \in A} y_{jt}^{vr} = 0 \qquad \forall r \in R, t \in \Gamma \tag{3.7}$$

$$\sum_{(i,j) \in A} y_{ij}^{vr} - \sum_{(j,i) \in A} y_{ji}^{vr} = 0 \qquad \forall v \in V, r = (i^+, i^-) \in R, i \in W \setminus \{\Gamma \cup \{i^+, i^-\}\} \tag{3.8}$$

$$y_{ij}^{vr} \leq x_{ij}^v \qquad \forall (i, j) \in A, v \in V, r \in R \tag{3.9}$$

$$d_i^v + \tau_{ij} - a_j^v \leq M(1 - x_{ij}^v) \qquad \forall (i, j) \in A, v \in V \tag{3.10}$$

$$d_{i^+}^v \geq E_{i^+}, a_{i^-}^v \leq L_{i^-} \qquad \forall r = (i^+, i^-) \in R, v \in V \tag{3.11}$$

$$\sum_{(j,t) \in A} y_{jt}^{vr} + \sum_{(t,j) \in A} y_{tj}^{wr} \leq s_{tr}^{vw} + 1 \qquad \forall r \in R,\ t \in \Gamma,\ v, w \in V \tag{3.12}$$

$$a_t^v - d_t^w \leq M(1 - s_{tr}^{vw}) \qquad \forall r \in R,\ t \in \Gamma,\ v, w \in V \tag{3.13}$$

$$x_{ij}^v \in \{0, 1\} \qquad \forall (i, j) \in A,\ v \in V \tag{3.14}$$

$$y_{ij}^{vr} \in \{0,1\} \qquad \forall (i,j) \in A,\ v \in V,\ r \in R \tag{3.15}$$

$$s_{tr}^{vw} \in \{0,1\} \qquad \forall t \in \Gamma,\ r \in R,\ v, w \in V \tag{3.16}$$

$$a_i^v, d_i^v \geq 0 \qquad \forall i \in W,\ v \in V \tag{3.17}$$

The objective (1) is to find a minimum cost set of routes satisfying all requests. Constraints (2) enforce that each vehicle executes at most one route and constraints (3) enforce that a vehicle starts and ends a route at the same depot. Vehicle flow conservation is expressed by constraints (4). Constraints (5) and (6) guarantee that each request is served, i.e., it is picked up and delivered, respectively. The flow conservation for requests, at transfer locations, is enforced by constraints (7), and at non-transfer locations by constraints (8). Note that for transfer locations, a request entering the transfer is allowed to leave in a different vehicle, whereas for non-transfer locations a vehicle entering the location with a given request must leave the location with that request. Request and vehicle flows are linked by constraints (9), which enforce that a request has to travel in a vehicle. Consistency of arrival and departure times is enforced by big-M constraints (10) and time windows are enforced by constraints (11). Transfer operations are synchronized by constraints (12) and (13): if request $r \in R$ is transferred between vehicles v and w, $v, w \in V$, at location $t \in \Gamma$, then vehicle w can depart from transfer location t only after the arrival of vehicle v at transfer location t.

As in the above formulation, we will assume for the remainder that a request is only transferred once, that a vehicle visits a given transfer location only once, and that two vehicles meet and exchange requests only once. A vehicle can visit multiple transfer locations on its route.

3.3.2. BENEFITS OF TRANSFERS

To illustrate and assess the potential benefits of using transfers, we analyze a set of stylized instances and compare solutions with and without transfers.

Consider an instance in which the pickup and delivery locations are vertices of a regular n-sided polygon with side length S, inscribed in a circle of radius Υ. A single depot ($m \in M$) is located at the center of the circle, and also acts as a transfer location ($t \in \Gamma$). Let the common shift for vehicles be $[0, 4\Upsilon]$ and let the common time window for requests be $[0, 4\Upsilon]$ as well, with $S > \Upsilon$.

Next, we locate the pickup and delivery locations of the requests in such a way that the ratio k_1/k_2 can get very large, where k_1 is the minimum fleet size required to serve all requests when transfers are not allowed and k_2 is the minimum number of vehicles required when transfers are allowed, can get very large. Figures 3.1a, 3.1b, 3.1c, and 3.1d illustrate instances with 3, 4, 5, and 6 requests, respectively.

In Figure 3.1a, for example, six requests, $(1^+, 1^-), \ldots, (6^+, 6^-)$ are shown. If fewer than six vehicles are available to serve all requests when transfers are not allowed, then two requests must be picked up and delivered by the same vehicle. However, this is impossible, because two requests with a common pickup location do not have a common delivery location, which implies that the vehicle needs to visit three locations and the travel time

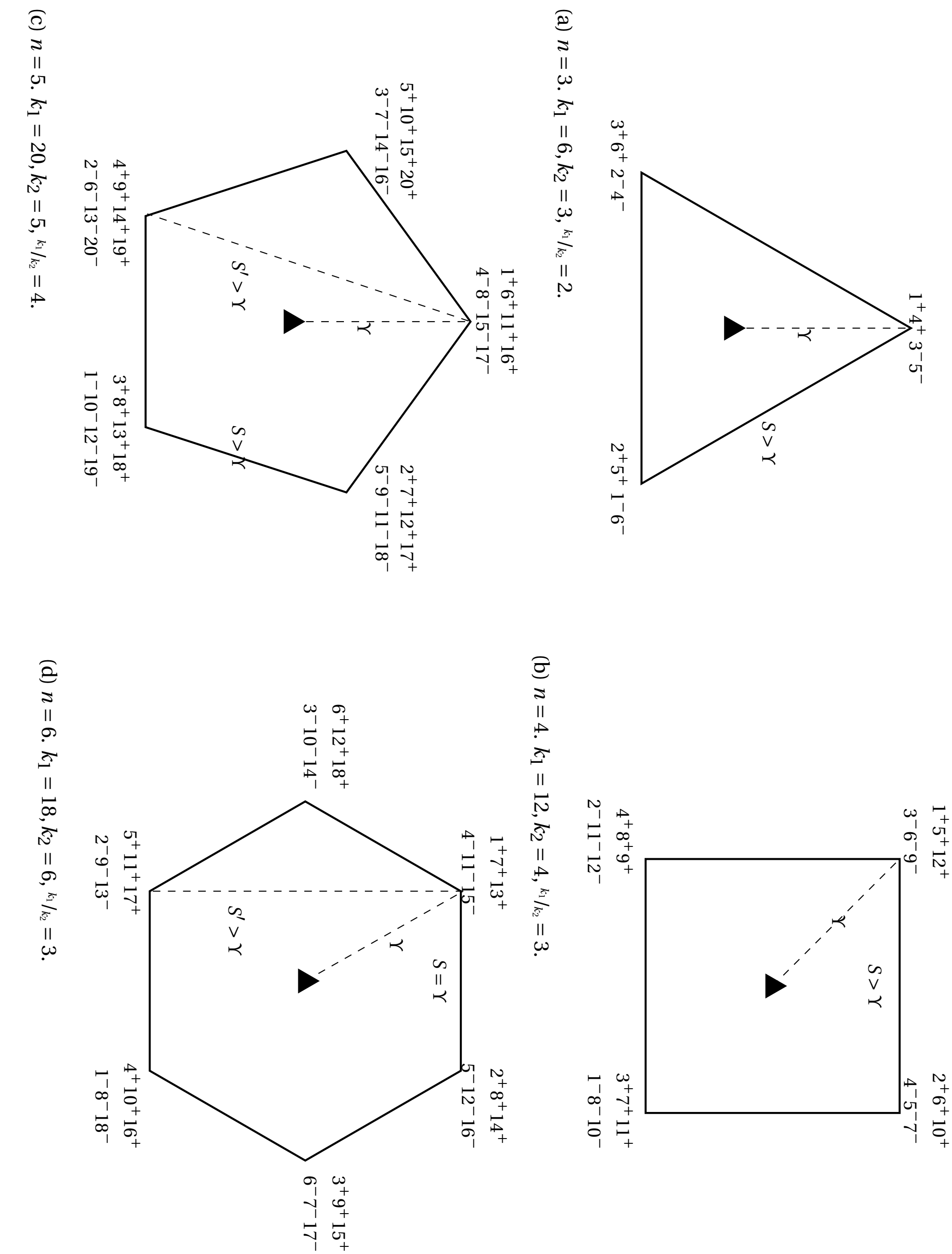


Figure 3.1: PDPTW-T instances where transfers result in significant benefits.

is at least $\Upsilon + S + S + \Upsilon$ which exceeds 4Υ (since $S > \Upsilon$). Observe that, if more requests are added, a fleet of six vehicles would still be enough to serve all requests, no matter where the pickup and delivery locations for the added requests are located. Thus, in the best solution without transfers, $k_1 = 6$ vehicles (routes) are required, each departing from the depot, visiting two vertices of the polygon (one with the pickup and another with the delivery), and returning to the depot.

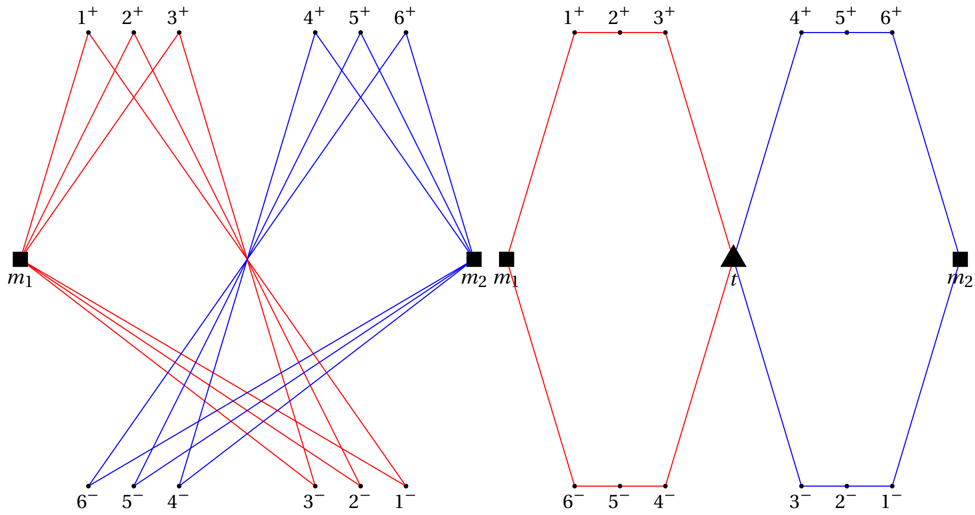
When transfers are allowed, the best solution requires $k_2 = 3$ vehicles, each performing a route departing from the depot, visiting one vertex of the polygon, then the transfer location, followed by a vertex and returning to the depot. All loads are picked up and taken to the transfer location where they are consolidated and then taken to their final destination. It is not possible to use fewer than three vehicles. If two vehicles are used, then one of them has to visit two locations to pickup loads and take them to the transfer location, which would not leave enough time for delivery. Thus, for this setting, we have $^{k_1}/_{k_2} = 2$.

Figure 3.2 illustrates the potential benefits of transfers in a setting with two depots and a single transfer location. In the example, the six requests $(1^+, 1^-), \ldots, (6^+, 6^-)$ can be serviced by vehicles from any depot. Deliveries $1^-, 2^-$ and 3^- are located in a line and spaced by S time units. The time windows for those locations are such that if a vehicle picks all three requests, the deliveries should be visited in the order $3^-, 2^-, 1^-$. In particular, latest delivery times are set as $L_{1^-} = L_{3^-} - S$ and $L_{2^-} = L_{1^-} - S$ and L_{3^-} small enough such that a vehicle visiting 1^- or 2^- before visiting 3^- is not able to reach 3^- before L_{3^-}. Moreover, the shift length is such that a vehicle departing from 1^- after visiting 3^- and/or 2^- is not able to return to m_1 in time. As a consequence, when transfers are not allowed, requests $(1^+, 1^-), (2^+, 2^-), (3^+, 3^-)$ have to be served by a different vehicle (Figure 3.2a). Observe that for requests $(4^+, 4^-), (5^+, 5^-), (6^+, 6^-)$ we have a similar situation, as the locations are positioned on symmetrical positions, with depot m_2 replacing m_1. When transfers are allowed, two vehicles, one from depot m_1 and one from depot m_2, can serve all requests (Figure 3.2b). Note that, after visiting transfer t and swapping the requests, the deliveries still have to be visited in the order $3^-, 2^-, 1^-$ (and $4^-, 5^-, 6^-$.).

3.4. ADAPTIVE LARGE NEIGHBORHOOD SEARCH

Solving instances of the mathematical model presented in Section 3.3.1 in a reasonable amount of computation time is only possible for small instances; Rais et al. (2014) presents results for instances with 10 and 14 locations (5 and 7 requests, respectively) and where transfers are allowed at every location. This is due, in part, to the symmetry (vehicles are indistinguishable) and the use of big-M constraints, which results in weak linear relaxations. Therefore, we develop an Adaptive Large Neighborhood Search (ALNS) algorithm which is able to handle large instances in short computation times.

ALNS algorithms have been successfully applied to many hard combinatorial problems, in particular to many variants of the vehicle routing problem (see Pisinger and Ropke (2007)). An ALNS algorithm specifically designed for solving instances of the PDPTW is described in Ropke and Pisinger (2006). A successful ALNS algorithm relies



(a) (b)

Figure 3.2: A PDPTW-T instance where transfers result in significant benefits.

on the availability of several fast-to-explore local search neighborhoods for modifying a solution. We focus on neighborhoods that remove requests from the current solution and reinsert them to create a modified solution. At each iteration, one of the neighborhoods is selected, giving priority to neighborhoods that have been successful in earlier iterations. Improving solutions are always accepted, but a diversification mechanism is incorporated which sometimes allows for the acceptance of worse solutions. The basic flow of our ALNS algorithm is given in Algorithm 1.

An initial solution, in which some of the requests are transferred, is obtained as described in Section 3.4.1. A removal and insertion operator are chosen (line 4) based on weights that reflect past performance. The removal and insertion operators are applied in sequence (line 5). The acceptance of the resulting solution is controlled by a scheme that is similar to those found in simulated annealing (SA) algorithms. More specifically, given the current solution S^c, a solution S is accepted with probability $e^{-(z(S)-z(S^c))/T}$ where $T > 0$ is the current temperature. The temperature starts at T_0 and is decreased at every iteration of Algorithm 1 by the cooling rate $0 < \alpha < 1$, that is $T_{i+1} = T_i\alpha$. T_0 is given by $\frac{\omega}{-\ln(0.5)} z_0$, where z_0 is the cost of the initial solution and $0 < \omega < 1$, i.e., a solution with cost $(1+\omega)z_0$ has probability 0.5 of being accepted. A higher value ω (and consequently T_0) allows for more diversification throughout the search, especially during the early stages, but may still be insufficient to avoid getting trapped in local optimum. Therefore, we take a similar approach to Stenger et al. (2013) and reset the temperature to $\frac{\omega}{-\ln(0.5)} z^*$ after v iterations in which no improving solution has been found, where z^*

Algorithm 1: Overview of the main steps in the proposed solution method (ALNS)

Input : Set $R = \{(i^+, i^-) | i^+ \in P, i^- \in D\}$ with pickup and delivery requests; Set of transfer locations, T

Output: Routing plan S^*

$I \leftarrow$ Construct an initial solution having a subset of the requests transferred;
$S^*, S^c \leftarrow I$;
while *Termination criteria not satisfied* **do**
    Choose a removal (O^-) and an insertion (O^+) operators;
    $S \leftarrow O^+(O^-(S^c))$;
    if $z(S) \le z(S^c)$ **then**
        $S^c \leftarrow S$;
        if $z(S) \le z(S^*)$ **then**
            $S^* \leftarrow S$;
    else
        $S^c \leftarrow accept(S, S^c)$;
    Update weights used for operators selection;
return S^*;

is the cost of the best solution found so far.

3.4.1. INITIAL SOLUTION

A critical aspect of an ALNS algorithm for solving the PDPTW-T concerns the strategies embedded for deciding which requests to transfer and at which locations. Such strategies have to be embedded in the solution improvement framework (i.e., removal/insertion operators that explicitly consider transfers), but starting with an initial solution in which some requests are being transferred might help guide the search – requests are considered one at a time and introducing a (new) transfer is not likely to look attractive as it involves detours to visit a transfer location. Thus, by having (some) transfers in the initial solution, an ALNS algorithm is more likely to make effective use of transfer options.

Any solution, s, is characterized by a subset $R'_s \subset R$ of requests being transferred. We create an initial solution, s_0, by forcing a subset of requests to be transferred in the initial solution. Next, we describe how we obtain $R'_{s_0} \subset R$ (3.4.1) and how we decide on the transfer locations to use (3.4.1) when constructing s_0 (3.4.1).

SELECTING REQUESTS TO TRANSFER

For a request $r_i = (i^+, i^-) \in R$, let τ_i be the (direct) travel time between i^+ and i^-. Selecting the requests that are forced to use a transfer in the initial solution is controlled by a threshold τ'. We let $\mathscr{R} = \{(i^+, i^-) \in R; \tau_i \ge \tau'\}$ be the candidate set for transferred requests in the initial solution.

Transferring of requests having relatively long travel times might yield cost savings as it may induce "regional" vehicle routes in which requests with pickup and delivery locations in a region are served without a transfer and requests with pickup and delivery locations in different regions are served using a transfer. The threshold τ' is defined based on the characteristics of an instance, e.g., half of the longest possible travel time.

SELECTING A TRANSFER LOCATION

Given the set of candidate requests to be transferred in the initial solution, the next decision concerns which transfer location to use. To prevent transfers that require a large detour (compared to traveling directly from pickup to delivery location), transferring $r_i \in \mathscr{R}$ via $t \in \Gamma$ is considered only when $\tau_{i^+,t} + \tau_{t,i^-} \leq \gamma\tau_i$, where $\gamma > 1$ is a parameter controlling the maximum allowed ratio between the detour $\tau_{i^+,t} + \tau_{t,i^-}$ and the direct travel time τ_i. Moreover, transferring request r_i via t has to be feasible for at least one combination of depots m_a and m_b, i.e., feasible routes $m_a - i^+ - t - m_a$ and $m_b - t - i^- - m_b$ that can be timed and synchronized to induce a feasible transfer at t have to exist. If there is no feasible transfer location for a request in the candidate set, the request is removed from the candidate set.

Among the feasible transfer locations $t \in \Gamma$ for request (i^+, i^-) we select the one that minimizes the difference between $\tau_{i^+,t}$ and τ_{t,i^-}. This strategy favors transfer locations that are located more centrally between the pickup and delivery locations of a request, thus better balancing the detour required to visit the transfer between the two vehicles serving the request. Figure 3.3 illustrates the possible transfer assignment for a request (i^+, i^-).

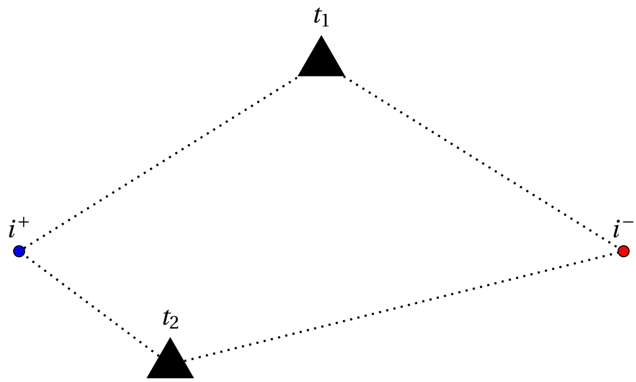


Figure 3.3: Selecting a transfer location for request (i^+, i^-). Location t_1 is selected, since it gives more balanced detours.

CONSTRUCTING THE INITIAL SOLUTION

The initial solution, s_0, is constructed such that all requests in R'_{s_0} are served by two vehicles and transferred at the location selected using the strategy presented in Section

3.4.1. All other requests $r_i \in R \setminus R'_{s_0}$ are served by a single vehicle. To construct a feasible solution, all requests in $R \setminus R'_{s_0}$ are put in the "request bank" and one iteration of the ALNS algorithm for the PDPTW by Ropke and Pisinger (2006) is executed to obtain a partial solution for those requests. Then, for each request $r_i \in R'_{s_0}$, we use a "Greedy insertion with transfer" operator of our ALNS framework (described in Section **??**) to (try and) insert r_i using the selected transfer location. If unsuccessful, two new routes are created: $m_a - i^+ - t - m_a$ and $m_b - t - i^- - m_b$, where depots $m_a, m_b \in M$ are chosen such that the total travel time of the two new routes is minimized.

3.4.2. IMPROVEMENT PHASE

At the heart of any ALNS algorithm is a set of *operators*, each modifying the current solution in a specific and limited way, i.e., making small modifications to the current solution, and, thus, defining a neighborhood. As is common in many ALNS algorithms for routing problems, our operators either remove or insert sets of locations (either pickup or delivery locations).

For a given solution, let $\rho(i)$ and $\sigma(i)$ denote the direct (on the same route) predecessor and successor nodes, respectively, of node i. The detour cost associated with node i is $\delta_i = c_{\rho(i),i} + c_{i,\sigma(i)} - c_{\rho(i),\sigma(i)}$. The detour cost associated with request $r = (i^+, i^-)$ is $\delta_r = \delta_{i^+} + \delta_{i^-}$ when $\sigma(i^+) \neq i^-$, and $\delta_r = c_{\rho(i^+),i^+} + c_{i^+,i^-} + c_{i^-,\sigma(i^-)} - c_{\rho(i^+),\sigma(i^-)}$ when $\sigma(i^+) = i^-$. Similarly, if request r is not yet served in the solution, then the cost to insert its pickup i^+ (its delivery i^-) between nodes u and v ($v = \sigma(u)$) is the detour cost $c_{u,i^+} + c_{i^+,v} - c_{u,v}$ ($c_{u,i^-} + c_{i^-,v} - c_{u,v}$), which we also denote as δ_{i^+} (δ_{i^-}). The insertion cost of request $r = (i^+, i^-)$, given insertion positions (u^+, v^+) and (u^-, v^-) for the pickup and delivery nodes, respectively, is the detour cost $\delta_r = \delta_{i^+} + \delta_{i^-}$ when $(u^+, v^+) \neq (u^-, v^-)$, and $c_{u,i^+} + c_{i^+,i^-} + c_{i^-,v} - c_{u,v}$ when $(u^+, v^+) = (u^-, v^-) = (u, v)$.

REMOVAL OPERATORS

A removal operator deletes a number of requests from the current solution and adds them to a request bank (from which insertion operators select requests to be inserted). Let $O_k^-(s)$ denote the (partial) solution that results after removal operator O_k^- is applied to solution s, and let $R_k^-(s)$ denote the set of requests deleted from solution s by operator O_k^-. Algorithm 2 illustrates the steps performed by a removal operator.

1. Worst request removal (O_1^-): The operator deletes q requests in non-decreasing order of their detour cost.

2. Random request removal (O_2^-): The operator deletes q requests selected at random.

3. Route removal (O_3^-): The operator uses a biased selection procedure to choose a route to delete from the current solution. Only routes that do not visit a transfer are considered. The probability that a route is selected to be deleted is proportional to the ratio of the waiting time in the route and the number of requests served in the

Algorithm 2: Steps of removal operator O_k^-

Input : Feasible solution, s
Output: Partial solution $O_k^-(s)$ and set of deleted requests $R_k^-(s)$

$q \leftarrow$ Number of requests to delete;
$O_k^-(s) \leftarrow s$;
$R_k^-(s) \leftarrow \emptyset$;
for $1, 2..., q$ **do**
  Select a request $r \in R \setminus R_k^-(s)$ to be deleted;
  $O_k^-(s) \leftarrow O_k^-(s) \setminus r$;
  $R_k^-(s) \leftarrow R_k^-(s) \cup r$;
return $O_k^-(s), R_k^-(s)$;

route. Thus, the operator favors the deletion of less efficient routes. All requests in the chosen route are deleted. If the number of requests deleted is less than q, then another route is selected. The process continues until at least q requests are deleted.

4. Transfer-based request removal (O_4^-): This operator considers requests that have been transferred in one or more previous solutions. For each request $r_i = (i^+, i^-)$, let t_i^* denote the number of times request r_i has been transferred in an incumbent solution encountered during the search. The probability that a request r_i is selected is proportional to $1 - \frac{t_i^*}{I^*}$, where I^* denotes the (total) number of incumbent solutions encountered during the search. Note that requests that are seldom transferred in incumbent solutions are more likely to be deleted than requests that are often being transferred in incumbent solutions.

5. Cluster removal (O_5^-): This operator is an adaptation of an operator used in Masson et al. (2013, 2014) and based on the idea that if the pickups (deliveries) of a set of requests form a cluster, it may be beneficial if they are picked-up (delivered) by the same vehicle and dropped off at (collected from) a transfer location. In our implementation, we randomly pick a "root" request (i^+, i^-) and compute sets C_i^+ and C_i^-, where C_i^+ contains pickup locations within radius $\mu = 60$ units of distance from i^+ being serviced by a different vehicle (and the delivery location is not within μ from i^+) and C_i^- contains delivery locations within radius μ from i^- being serviced by a different vehicle (and the pickup location is not within μ from i^-). The requests in the larger set are deleted. The parameter q is ignored.

After the deletion of requests, it may happen that a vehicle visits a transfer location, but no transshipment of any request takes place. We allow such unnecessary visits to remain in the current solution for at most ϕ iterations. If in ϕ consecutive iterations a transfer location t is not used to transfer any request, it is removed from the route(s).

Allowing unnecessary visits to transfer locations can be beneficial, because inserting a request involving a transfer location that is not yet in the solution results in additional detour costs and, consequently, a deterioration in solution quality. If the transfer location is already in the solution, even if it is currently not used to transfer requests, the increase is only due to the detour cost incurred for the inserted request itself. Moreover, it also increases the likelihood that the transfer location is already visited by multiple vehicles. A downside of keeping transfer locations with unnecessary visits in the solution is that the time required to visit the transfer location may render an otherwise feasible insertion infeasible. Moreover, more insertions with transfers are evaluated and, as a consequence, execution time is longer. Despite these downsides, our experiments have shown that keeping transfer locations with empty visits provides a good mechanism for exploring transfer opportunities. A post-processing step at the end of the search removes any unnecessary visits at transfer locations in the best solution found.

INSERTION OPERATORS

The objective of an insertion operator is to reintroduce requests in the set $R_i^-(s)$, i.e., the requests deleted by removal operator O_i^-. Let $O_j^+(s)$ denote the solution obtained after applying insertion operator O_j^+ to a (partial) solution s. Algorithm 3 presents the steps performed by an insertion operator that does not consider transfers, and Algorithm 4 presents the steps performed by an insertion operator that does consider transfers.

1. Greedy insertion without transfer. Requests are sequentially inserted in the least-cost position and route. If a request cannot feasibly be inserted, it remains in the request bank. The sequence in which requests are inserted defines an operator:
   - O_1^+: Requests are selected in decreasing order of direct travel time from pickup to delivery location.
   - O_2^+: Requests are selected in increasing order of insertion cost. More formally, if Δ_{rk} represents the least cost to insert request r in route k ($\Delta_{rk} = \infty$ if the insertion is infeasible), then the operator selects request r' to be inserted in route k' as $(r', k') = \text{argmin}_{r \in R_i^-(s),\, k \in K} \Delta_{rk}$. Values Δ_{rk} are updated after the selected request has been inserted in its associated route.
   - O_3^+: Requests are selected in decreasing order of regret, i.e., the absolute difference between least insertion cost and the second least insertion cost. More formally, if Δ_{rk}^1 is the least-cost insertion of request r in a route k and Δ_{rk}^2 is the second least-cost insertion of request r in a route k (where k may be the same or a different route), then the operator selects request r' to be inserted in route k' as $(r', k') = \text{argmax}_{r \in R_i^-(s),\, k \in K}(\Delta_{rk}^2 - \Delta_{rk}^1)$. Values Δ_{rk}^1 and Δ_{rk}^2 are updated after the selected request has been inserted in its associated route.
   - O_4^+: Similar to operator O_2^+, but the request is selected randomly among the best ψ insertions, where, after initial experiments, ψ is set to 30% of the feasible insertions.

Algorithm 3: Steps of insertion operator that does not consider transfers

Input : Partial solution s and set of requests R to be inserted
Output: A complete solution $\bar{s}$, or a partial solution $\bar{s}$ plus a set $\bar{R}$ of requests that were not inserted

$L \leftarrow$ requests in R ordered accordingly to some criterion;
$\bar{R} \leftarrow \varnothing$;
$\bar{s} \leftarrow s$;
$I \leftarrow \varnothing$ // list of feasible insertions;
for $r = (i^+, i^-) \in L$ **do**
    feasible $\leftarrow$ False;
    for *each route v in s* **do**
        for $k_1 \in \{0, 1, ..., K\}$, *positions within route v* **do**
            evaluate insertion of i^+ at position k_1 in v;
            if *insertion is feasible* **then**
                for $k_2 \in \{k_1, ..., K\}$ **do**
                    evaluate insertion of i^- at position k_2 in v;
                    if *insertion is feasible* **then**
                        feasible $\leftarrow$ True;
                        $I \leftarrow I \cup (k_1, k_2, v)$;
    if *feasible=False* **then**
        $\bar{R} \leftarrow \bar{R} \cup r$
    insertion r in $\bar{s}$ using least-cost insertion $i \in I$;
return $\bar{s}, \bar{R}$;

Algorithm 4: Steps of insertion operator that considers transfers.

Input : Partial solution s and set of requests R to be inserted

Output: A complete solution $\bar{s}$, or a partial solution $\bar{s}$ plus a set $\bar{R}$ with requests that were not inserted

```
L ← requests in R ordered accordingly to some criterion;
R̄ ← ∅;
s̄ ← s;
I ← ∅ // list of feasible insertions;
for r = (i⁺, i⁻) ∈ L do
    feasible ← False;
    for each route v and transfer location t visited by v in s do
        n ← position of t in v;
        for k_1 ∈ {0, 1, ..., n − 1}, positions in v do
            evaluate insertion of i⁺ at position k_1 in v;
            if insertion is feasible then
                for each route w visiting t do
                    m ← position of t inh w;
                    for k_2 ∈ {m + 1, ..., K} positions in route w do
                        evaluate insertion of i⁻ at position k_2 in w;
                        if insertion is feasible then
                            feasible ← True;
                            I ← I ∪ (k_1, k_2, t, v, w)
    if feasible=False then
        R̄ ← R̄ ∪ r
    insert r in s̄ using least-cost insertion i ∈ I;
return s̄, R̄;
```

2. Greedy insertion with transfer. Requests are sequentially inserted in the least-cost positions and routes. If a request cannot feasibly be inserted, it remains in the request bank. Given a request $r = (i^+, i^-)$, an operator seeks two routes, k_a and k_b, and a transfer location, t, such that the possible visit times at t for k_a and k_b overlap (i.e., there exist a time when both vehicles can be at t). If it is possible to insert i^+ before t in k_a and i^- after t in k_b (properly accounting for any change in the arrival time at t of route k_a due to the insertion of i^+), the insertion is considered feasible. The sequence in which requests are inserted defines an operator:

   - O_5^+: Requests are selected in decreasing order of direct travel time from pickup to delivery location.
   - O_6^+: Requests are selected in increasing order of insertion cost. More formally, if $\Delta_{rk_1k_2}$ is the cheapest cost to insert request $r = (i^+, i^-)$, where i^+ and i^- are inserted in routes k_1 and k_2, respectively ($\Delta_{rk_1k_2} = \infty$ if the insertion is infeasible), then the operator selects a request r' and routes k_1' and k_2' by $(r', k_1', k_2') = \text{argmin}_{r \in R_i^-(s),\, k_1, k_2 \in K} \Delta_{rk_1k_2}$. Values $\Delta_{rk_1k_2}$ are updated after the selected request has been inserted in its associated routes.
   - O_7^+: Requests are selected in decreasing order of regret insertion with transfers. More formally, if $\Delta^1_{rk_1k_2}$ represents the least-cost insertion of request r using a transfer, and $\Delta^2_{rk_1k_2}$ represents the second least-cost insertion of request r using a transfer (where the routes in which the pickup and delivery location are inserted may be the same or may differ), then the operator selects request r' and its associated routes as $\text{argmax}_{r \in R_i^-(s),\, k_1, k_2 \in K} (\Delta^2_{rk_1k_2} - \Delta^1_{rk_1k_2})$. Values $\Delta^1_{rk_1k_2}$ and $\Delta^2_{rk_1k_2}$ are updated after the selected request has been inserted in its associated routes.

3. Transfer insert O_8^+: This operator facilitates the inclusion of transfers in a solution by creating routes with an unnecessary visit at a transfer location. Requests are selected in decreasing order of direct travel time between pickup and delivery location. For a request $r = (i^+, i^-)$, let C_i be the circle with radius $\frac{\tau_i}{2}$, where τ_i is the distance between i^+ and i^-, centered at the midpoint of the line segment joining i^+ and i^-. If there is a transfer location t within C_i and a feasible route $m_v - i^+ - t - i^- - m_v'$ for vehicle $v \in V$ (not yet in the solution), this route is created. If there are multiple transfer locations in C_i, the one minimizing the difference $|\tau_{i^+,t} - \tau_{t,i^-}|$ is chosen.

It is possible that an insertion operator does not succeed to insert all requests deleted by a removal operator. In this case, the remaining requests are inserted as follows: find a route k_a with transfer location t_a maximizing the number of delivery locations that can be feasibly inserted in k_a after t_a. Let the set of these delivery locations be denoted by D. Similarly, find a route k_b with transfer location t_b maximizing the the number of pickup locations that can be feasibly inserted in k_b before t_b. Let the set of these pickup locations be denoted by P. If $|D| \geq |P|$, the delivery locations in D are inserted in k_a (if feasible) after t_a. For each inserted delivery location, if it is not possible to insert the corresponding pickup in a route already in the solution, a new route $m_v - i^+ - t_a - m_v$ is created (assuming a vehicle v is available). If $|D| < |P|$, a similar process is used, but starting with the pickup locations. If, at the end, there are still remaining requests, these are inserted in new routes without transfers.

OPERATOR SELECTION

The adaptive weight mechanism used to control the selection of a removal and insertion operator is implemented by a roulette wheel procedure, as described in Pisinger and Ropke (2007). If the weight of operator O_i, $i \in 1, \ldots \Pi$, is π_i, expressing the desirability of selecting operator O_i, then the probability of selecting O_i is given by $\pi_i / \sum_{j=1}^{\Pi} \pi_j$. The removal and insertion operators are chosen independently, i.e., with different roulette wheels. At the start of the search, all operators have equal weights. During the search, the weights are updated every κ iterations, similarly to Pisinger and Ropke (2007). As discussed above, in our ALNS algorithm, two types of insertion operators are considered: operators in which transfers are considered, and operators considering transfer operations. Since the insertion of a node (request) can be feasible for one type, but not for the other, our ALNS algorithm selects two insertion operators, one of each type. The operator with the largest weight is applied first, and if there are remaining nodes (requests), then the other operator is applied.

The performance of each operator is recorded in its weight based on the solutions obtained after the operator is applied. In particular, the score $\bar{\pi}_i$ for operator O_i starts at 0 at the start of each segment of κ iterations, and is incremented by: σ_1 if an overall best solution is found, σ_2 if an improving solution is found, and σ_3 if a worse solution is found and accepted (as part of the annealing scheme). If a_i is the number of times operator O_i was used during the current segment (of κ iterations), then the new weight π_i is given by $\pi_i(1-\rho) + (\bar{\pi}_i / a_i)\rho$, where $\rho \in [0,1]$ adjusts the importance given to scores in past segments. Since two insertion operators may be applied in one iteration, their weights are updated independently.

EFFICIENCY

The performance of an insertion-based neighborhood search heuristic relies heavily on its ability to efficiently evaluate the feasibility of an insertion, as a very large number of insertions will be attempted. In Appendix B, we discuss the information that is kept with the current solution to allow fast (constant time) evaluation of the feasibility of an insertion. Moreover, we show how to update this information when the current solution changes.

3.5. COMPUTATIONAL EXPERIMENTS

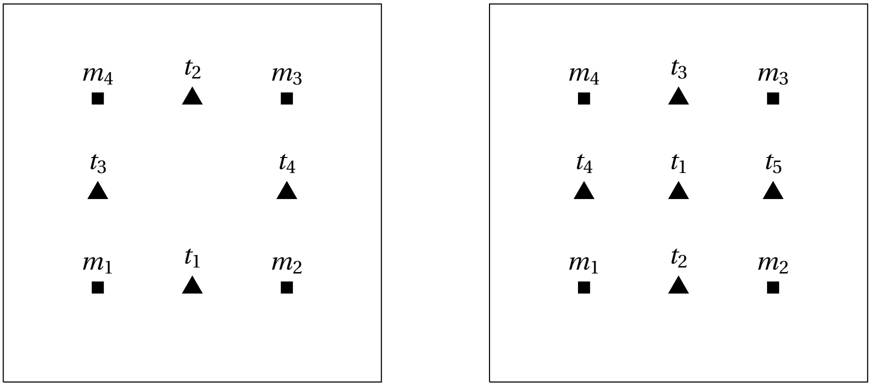
In this section, we present the results of a set of computational experiments conducted to evaluate the performance of the ALNS algorithm and to characterize transportation settings which might benefit from the introduction of transfers. The ALNS algorithm was coded in C++ and the experiments were conducted using an Intel Core i5-2450M machine, running at 2.5 GHz x 4 with 4 GB of RAM, under Ubuntu 16.04. To investigate the benefits of transfers, we compare the solutions obtained by our ALNS algorithm to the solutions obtained by the ALNS algorithm of Ropke and Pisinger (2006), which does not consider transfers. Furthermore, we analyze the impact of different problem characteristics, in particular, the vehicle's shift length, the number of transfer locations, the geographic distribution of transfer locations, and the geographic distribution of customers.

3.5.1. INSTANCE GENERATION

To assess the potential benefits of transfers in freight transportation systems employing crowd-sourced drivers, we generate sets of instances that capture characteristics of such systems. We consider a squared geographical area of 120×120 units of distance and assume that one unit of distance can be traveled in one time unit (e.g. 60 km/h). Four depots are located in the area as shown in Figure 3.4a, requests can be serviced from any depot, and the number of vehicles in each depot is sufficient to service all requests. Pickup and delivery locations are drawn uniform randomly in the area, but we consider different scenarios, $\mathscr{C}$, for origin-destination proximity: long-distance requests only (L), i.e., at least 60 units between a pickup and a delivery location, short-distance requests only (S), i.e., no more than 60 units of distance, but at least 30 units between a pickup and a delivery location, and a third scenario having both long-distance and short-distance requests (M).

We generate instances with 50, 75, and 100 requests, each with a 180 units of time (e.g., 3-hour) time window, i.e., $E_{i^+} = 0$ and $L_{i^-} = 180$ for all $r_i \in R$, for each class L, M and S. If $\gamma = {}^1/_{|R|} \sum_{r_i \in R} \tau_i$ indicates the origin-destination proximity of an instance, then for the generated instances in classes L, M, and S, we have, on average, $\gamma = 74.67$, $\gamma = 57.69$ and $\gamma = 45.82$, respectively. We consider vehicle shift lengths 180, 240, and 300 (i.e., drivers operate for three, four, or five hours). The instances are created such that even for the shortest shift length ($H = 180$), all requests can be served (i.e., there is at least one depot such that a vehicle from that depot can feasibly visit the pickup and delivery locations of a request on a route). Moreover, we consider two geographies for the transfer locations, MD-4T and MD-5T, shown in Figures 3.4a and 3.4b, respectively. In the former there are four transfer locations, and in the latter there are five transfer locations, with one additional transfer location in the center of the region. The full set of instances can be downloaded at `http://dx.doi.org/10.17632/pywzcgyzrv.1`.

We note that Mitrović and Laporte (2006) generated instances of the PDPTW-T motivated by an application in last-mile freight transportation, and used these in their computational experiments. Furthermore, Masson et al. (2013) generated instances of the PDPTW-T motivated by an application in which transportation requests of disabled people need to be served. As indicated earlier, our research is motivated by an environment in which crowd-sourced drivers, working for short periods of time, transport shipments



(a) MD-4T (b) MD-5T

Figure 3.4: Basic settings for multi-depot instances generation. Squares and triangles represent depots and transfers, respectively.

from origin to destination. As a consequence, the instances available in the public domain are not representative of the operating environment we are exploring, and, thus, the instances are not well-suited for testing our methodology. Nevertheless, for the sake of completeness and as a validation of our methodology, we present results for a subset of the instances proposed by Mitrović and Laporte (2006) in the appendix.

3.5.2. ALNS ALGORITHM PARAMETER TUNING

In Table 3.1, we show the parameters (column Par.) and the values used in our ALNS. In the left side of the table, we present the parameters that are set to similar values used in Ropke and Pisinger (2006). In the right side of the table, we present the parameters we believe have a significant impact on the performance of our proposed ALNS algorithm and for which the values have been carefully calibrated.

Table 3.1: ALNS Parameters and values

Par.	Description	Value	Par.	Description	Value
ι	ALNS iterations	25K	ψ_1	Min. removed requests	$0.1\|R\|$
α	SA cooling rate	0.98	ψ_2	Max. removed requests	$0.2\|R\|$
ω	Initial temperature adjustment	0.15	σ_1	Score for incumbent solution	40
κ	Level length	250	σ_2	Score for improving solution	10
ν	Max. non-improving iterations	250	σ_3	Score for worse, new solution	1
ρ	Reaction factor	0.2	ϕ	Max. empty transfers iterations	100
μ	Cover radius used in removal O_5^-	60			

Parameter tuning was done using a set of six instances, three with 30 and three with 50 requests – one instance for each scenario of origin-destination proximity, L, S and M –, as described in 3.5.1. All parameters to be tuned are given a default value. Then, a single parameter value (or single parameter set values) is varied while the others are kept fixed. For each parameter value considered (of the parameter being tuned), we solve each of the six instances five times. The parameter value yielding the best results is chosen. The process continues until all parameters have been assigned a value. The parameters to be tuned, and their respective default values, are $((\psi_1,\psi_2),(\sigma_1,\sigma_2,\sigma_3),\phi) = ((6,10),(10,10,10),100)$, where parameters $0 < \psi_1 < \psi_2 < 1$ control the fraction of requests to be removed, $\sigma_1,\sigma_2,\sigma_3$ are the reward values for operators and ϕ is the maximum number of iterations an unused transfer is allowed to remain in the current solution. In the initial solution for an instance, each request is served individually by a single vehicle without a transfer. To assess the quality of different parameter values, we assign a score to each solution obtained: ${}^{z^*}/_{z}$, where z is the cost of the solution and z^* is the cost of best solution encountered during an experiment. Because we also want the proposed ALNS algorithm to be reasonably efficient, we adjust the score to take run times into account as well: $0.7\,{}^{z^*}/_{z} + 0.3\,{}^{t^*}/_{t}$, where t is the run time and t^* is the fastest run time encountered during an experiment. Thus, a value close to 1.0 for a parameter value (the average score over the five runs) indicates superior performance.

Table 3.2: Calibration of ALNS parameters.

(ψ_1,ψ_2)			$(\sigma_1,\sigma_2,\sigma_3)$			ϕ	
(0.1,0.2)	(0.2,0.3)	(0.3,0.4)	**(40,10,1)**	(4,2,1)	(1,1,1)	0	**100**
0.98	0.89	0.85	*0.96*	0.95	0.93	0.88	*0.96*
0.87	0.82	0.76	*0.88*	0.86	0.85	*0.90*	0.88
0.87	0.79	0.73	*0.95*	0.92	0.92	0.88	*0.95*
0.95	0.88	0.83	0.96	0.97	*0.98*	*0.97*	0.96
0.96	0.87	0.83	*0.95*	0.91	0.88	0.93	*0.95*
0.93	0.84	0.77	*0.90*	0.74	0.78	0.72	*0.90*

[1] Each row corresponds to an instance and each column to the average score over five runs for the relevant parameter value(s). The best parameter value(s) is shown in bold.

First, we tune the interval from which the number of requests to be removed by a removal operator is drawn uniform randomly, i.e., $[\psi_1|R|,\psi_2|R|]$, where parameters $0 < \psi_1 < \psi_2 < 1$ control the fraction of requests to be removed. The results show that $(\psi_1,\psi_2) = (0.1,0.2)$ performs best. Allowing for a larger fraction of requests to be removed may lead to a wider exploration of the solution space, but it also increases time per remove-and-reinsert combination as requests are inserted one by one and execution time primarily depends on the number of requests inserted.

Next, we tune the reward values for the operators, where we note that their relative difference is more important than the values themselves. We consider three sets of parameters: one in which finding new best and better solutions is highly rewarded, i.e., $(\sigma_1,\sigma_2,\sigma_3) = (40,10,1)$, one in which success in finding new best and better solutions is recognized as a positive and rewarded, but not as much, i.e., $(\sigma_1,\sigma_2,\sigma_3) = (4,2,1)$ and, finally, one in which we do not actively “encourage” the search for new best and better solutions, but rely on the randomness in the search process to encounter high-quality new best and better solutions, i.e., $(\sigma_1, \sigma_2, \sigma_3) = (1,1,1)$. With the exception of one instance, the best results are obtained by highly rewarding success in finding new best and better solutions. This indicates that the SA acceptance mechanism, in combination with the restarting procedure, is enough to provide a satisfactory level of diversification in the search, and avoids getting trapped in local optima.

Finally, we calibrate the maximum number of iterations, ϕ, that an unnecessary transfer location visit is allowed to remain in the current solution. We consider two scenarios: one in which an unnecessary transfer location visit is removed immediately, i.e., $\phi = 0$, and one in which an unnecessary transfer location visit is kept much longer, i.e., $\phi = 100$. Even though run times increase when an unnecessary transfer location visit is kept longer, it appears to be worthwhile to do so. This is likely due to the fact that it significantly simplifies the search for beneficial transfer opportunities.

3.5.3. STYLIZED INSTANCES

We have used the proposed ALNS algorithm to solve stylized instances as shown in Figure 3.1, with 4, 5, 6, 7 and 8 sided polygons inscribed in a circle with radius 100 (where instances for 7 and 8 sided polygons are created as expected). We compare the solutions obtained by the proposed ALNS algorithm, where the center of the circle can be used as transfer location, to the solutions obtained by the ALNS algorithm of Ropke and Pisinger (2006) (i.e., without any transfers), and to the optimal solutions. The results can be found in Table 3.3. For each instance, we report the total travel distance (Dist), the number of vehicles used (Veh), the total computational time in seconds (Time) and the number of requests transferred (Trans).

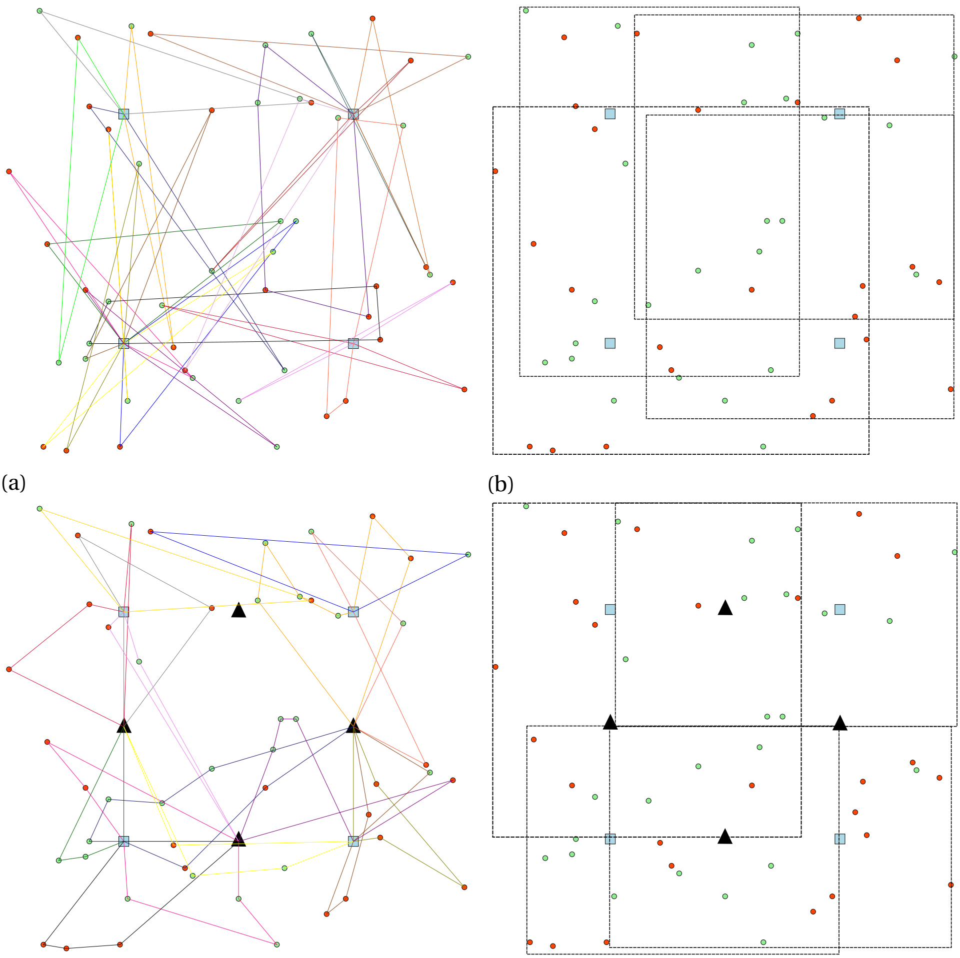
Our proposed ALNS is able to obtain the optimal solution for all but one of the instances. In constructing the polygons, we took distance and time to be equivalent (traveling one unit of distance requires one unit of time) and forced distances and travel times to be integer valued. As a result, for the 7-sided polygon it was not possible to obtain a completely regular shape. This hinders synchronization at the transfer points, and hence a lower quality solution resulted. The higher run times of our proposed ALNS algorithm compared to the one of Ropke and Pisinger (2006) are the result of more complex and more time consuming insertion operators, because the need to consider transfers. This is especially true for these stylized, highly symmetrical instances, where proper synchronization of transfers is critical.

Table 3.3: Algorithm comparison using stylized instances.

Instance		ALNS Ropke			Proposed ALNS				Optimal	
Polygon	$\|R\|$	Dist	Veh	Time	Dist	Veh	Trans	Time	Dist	Veh
4-sided	12	4336	12	0.9	1600	4	10	11.3	1600	4
5-sided	20	7104	20	1.5	2000	5	15	25.5	2000	5
6-sided	30	7244	18	2.2	2400	6	24	48.3	2400	6
7-sided	42	10652	28	13.6	2992	9	36	32.6	2800	7
8-sided	56	14988	40	17.8	3200	8	49	94.6	3200	8

3.5.4. AN EXAMPLE OF TRANSFER BENEFITS

We first illustrate the potential benefits of transferring requests by analyzing the results of a single instance of class L with 25 requests, shift length 180, and transfer location geography MD-4T. Figure 3.5 shows two solutions, one in which transfers are not considered (Figure 3.5a) obtained with the ALNS by Ropke and Pisinger (2006), and one in which transfers are considered (Figure 3.5c) obtained with our proposed ALNS (squares indicate depots, triangles indicate transfer locations, and circles indicate pickup and delivery locations).



(a) (b) (c) (d)

Figure 3.5: Overview of a solution without transferred requests (a), and a solution having requests transferred (c).

In the solution without transfer (Figure 3.5a) all but six requests are served by a dedicated vehicle (i.e., a vehicle serving a single request), and 22 vehicles are used. In the solution with transfers (Figure 3.5c) only two requests are served by a dedicated vehicle and all other requests are transferred, and only 10 vehicles are used. To facilitate comparing the solutions with and without transfers and to better understand the benefits of transfers, we show, in Figures 3.5b and 3.5d, four rectangles, one for each depot, representing the area containing the routes originating from that depot. Observe that in Figure 3.5b, corresponding to the solution without transfers, there is a much larger overlap between the rectangles than in Figure 3.5d, corresponding to the solution with transfers. This illustrates that when transfers are possible, vehicles can stay closer to the depots and can perform pickups and deliveries of multiple requests within the shift; it is as if the transfer locations induce sub-regions within the region. When transfers are not possible, a vehicle performing a pickup has to also perform the corresponding delivery and consolidating multiple requests within the shift becomes more difficult.

Similar settings i.e., transfer location geography MD-4T, 3-hour time window, shift length 180, are used to generate a set of 15 instances with 10 requests, where five instances have long-only, five instances have short-only and five instances have mixed-distance requests. These instances are solved using the IP formulation presented in Section 3.3 and employing Gurobi 8.01 as the IP solver with a time limit of two hours. For each instance, we compute a solution without transfers, using the ALNS algorithm by Ropke and Pisinger (2006), and provide it as an initial solution to the solver. We also compute a solution using our proposed ALNS algorithm and provide it as an initial solution. The results can be found in Table 3.4, where we report the total distance (*Dist*), the number of vehicles used (*Veh*), and the optimality gap after two hours (*Gap*). Solutions in which one or more requests are transferred are labeled with a superscript t.

The results show that for most instances, Gurobi is unable to improve the initial solution provided in two hours (note the large optimality gaps after two hours of computing). If a solution without transfers is provided, a better solution is found for five instances, where in four of them the improved solution has at least one request that is transferred. The solutions produced by our ALNS algorithm are equal or better (in terms of total distance and number of vehicles used) in 14 out of the 15 instances. When the solutions produced by our ALNS algorithm are provided as initial solution, Gurobi again finds a better solution for five instances, where in three of them the improved solution uses fewer vehicles.

3.5.5. RESULTS FOR PROPOSED GENERATED INSTANCES

A set of 10 instances was generated for each combination of number of requests, origin-destination proximity, and vehicle shift length. Each instance in a set is solved five times using the ALNS algorithm by Ropke and Pisinger (2006) and our proposed ALNS algorithm, and the average cost of the solutions as well as the lowest cost among them are reported. Two initial solutions were considered in the runs with our proposed ALNS algorithm: the solution obtained with the ALNS algorithm by Ropke and Pisinger (2006) and

Table 3.4: Experiments for instances with 10 requests.

	without transfers		Gurobi			with transfers		Gurobi		
Inst	Dist	Veh	Dist	Veh	Gap(%)	Dist	Veh	Dist	Veh	Gap(%)
1	1566	10	1566	10	68.3	1319^t	9	1289^t	8	61.4
2	1358	9	1358	9	74.5	863^t	6	863^t	6	58.8
3	1590	10	1590	10	77.5	1117^t	7	1117^t	7	65.4
4	1353	9	1187^t	8	67.2	1022^t	7	1022^t	7	64.0
5	1536	10	1536	10	76.7	1130^t	8	1033^t	6	64.5
6	645	5	645	5	32.7	645	5	645	5	32.7
7	734	5	734	5	46.4	718^t	5	718^t	5	47.2
8	613	4	613	4	37.2	613	4	613	4	37.2
9	840	6	816^t	6	38.1	801^t	6	801^t	6	39.5
10	803	5	747^t	5	42.7	793^t	5	747^t	5	46.0
11	1190	8	1190	8	71.7	658^t	4	619^t	4	37.3
12	1165	8	1100^t	7	70.8	985^t	7	957^t	6	61.1
13	1077	8	1077	8	67.5	845^t	6	845^t	6	57.8
14	903	6	903	6	52.5	725^t	5	725^t	5	40.8
15	935	6	911	6	50.3	868^t	6	868^t	6	60.0

^{t} Solution has transferred requests.

3

the initial solution constructed as described in Section 3.4.1 using $\tau' = 0.8 \max_{r_i \in R}\{\tau_i\}$ and $\gamma = 1.25$. To evaluate the potential benefits of transfers, we compare the gap between solutions with transfers and solutions without transfers: $\Delta = (c^w - c)/c$, where c is the cost of the solution without transfers and c^w is the cost of the solution with transfers. Note that, because the solution found by the ALNS algorithm of Ropke and Pisinger (2006) is not necessarily optimal, we cannot claim that the benefits are solely due to adding transfers.

First, we consider transfer location geometry MD-4T. In Table 3.5 we report the results for instances with 50 requests for each origin-destination proximity class ($\mathscr{C}$) and vehicle shift lengths 180, 240 and 300 (H). Each row in the table presents average results over all instances for the given combination $(H, \mathscr{C})$. Column *RP ALNS* reports the cost (c) and the fleet size (Veh) of the solutions obtained with the ALNS algorithm by Ropke and Pisinger (2006) (in each of the five runs, the same solution was found). Column *Proposed ALNS* reports the results obtained using our proposed ALNS algorithm. Column *Best* reports results for the best among the five solutions, whereas column *Average* reports the average over the five solutions. Column s_0 indicates the initial solution used: s_r, solution produced by the ALNS algorithm of Ropke and Pisinger (2006), or s_t, solution produced by the scheme described in Section 3.4.1. Column Δ_D(%) reports the reduction in distance, column Δ_V(%) reports the reduction in the number of vehicles used, column *Trans* reports the number of transferred requests, and column *t(s)* reports the run time in seconds, excluding the time to obtain the initial solution. Note that scheme s_t requires little time (less than one second) whereas scheme s_r requires running the ALNS algorithm by Ropke and Pisinger (2006) which takes a considerable amount of time (on average 60% of the running time of our proposed ALNS algorithm). Detailed results can be found in the appendix.

We observe that transfers can provide significant benefits, especially when the driver shift length is short ($H = 180$) and the distance between pickup and delivery locations is long (L) – we see a reduction of almost 50% for both the total distance and the number of vehicles used. When the distance between pickup and delivery location is short (S), the benefits are minor, in the order of 1-2%, regardless of driver shift length. As expected, the benefits decrease when driver shift lengths increase, because with longer driver shift lengths routes can cover larger distances and serve more requests in the same route, which tends to be less costly than using transfers.

The effect of the initial solution appears to be minor. Starting from an initial solution in which requests are transferred tends to result in final solutions with slightly more requests being transferred than starting from an initial solution in which no requests are transferred. This is likely due to the fact that insertions with transfer operators are more likely to be rewarded during the early stages of the search when the initial solution already has some transfers. When starting from a high-quality (locally optimal) solution without transfers, introducing transfers in the solution is likely to increase the total distance and insertions with transfer operators are less likely to be rewarded. For instances in which the distance between pickup and delivery location is short (S), and where trans-

Inst.	RP ALNS				Proposed ALNS							
					Best				Average			
$H, \mathscr{C}$	c	Veh	t(s)	s_0	Δ_D(%)	Δ_V(%)	Trans	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)
180, L	6322.1	39.0	18.4	s_r	-47.7	-46.2	47.2	58.5	-45.8	-44.1	45.5	59.5
				s_t	-46.7	-45.1	45.7	58.7	-45.2	-43.4	44.8	61.5
180, M	4116.3	24.8	23.6	s_r	-30.7	-27.4	40.1	50.9	-27.4	-23.8	32.9	50.0
				s_t	-30.9	-27.8	39.6	55.0	-28.3	-24.8	36.8	53.8
180, S	2459.2	15.0	29.8	s_r	-2.3	-2.7	8.6	35.8	-1.1	-0.4	7.2	38.0
				s_t	-4.9	-1.3	20.4	44.1	-2.6	0.0	16.0	42.1
240, L	3864.7	18.8	31.7	s_r	-25.6	-13.3	47.7	54.0	-19.4	-10.6	36.2	51.1
				s_t	-26.2	-16.0	47.2	55.1	-22.5	-11.6	43.3	54.1
240, M	3012.0	15.1	34.9	s_r	-11.3	-5.3	26.3	50.3	-6.5	-2.4	15.4	44.3
				s_t	-15.6	-5.3	41.1	54.2	-9.3	-1.3	27.3	49.7
240, S	2195.1	11.8	39.1	s_r	-1.8	0.0	12.2	45.6	-0.6	0.0	4.1	41.9
				s_t	-1.5	2.5	22.1	55.3	1.4	5.1	19.6	49.2
300, L	3653.1	16.6	33.8	s_r	-20.4	-3.0	46.5	55.2	-13.5	-2.2	30.8	50.8
				s_t	-21.7	-3.6	47.3	49.8	-17.6	-1.9	42.1	53.4
300, M	2952.8	14.4	35.9	s_r	-8.6	0.7	29.2	53.5	-4.1	-0.6	12.5	45.7
				s_t	-10.7	-0.7	33.4	45.8	-6.8	1.5	24.8	48.5
300, S	2188.9	11.7	39.8	s_r	-0.8	-1.7	5.8	43.6	-0.3	-0.7	2.4	41.2
				s_t	-1.0	0.9	12.9	42.4	1.4	4.3	15.3	47.3

Table 3.5: Results for instances containing 50 requests with 3h time windows in the MD-4T setting. Column RP ALNS reports results obtained with Ropke and Pisinger (2006) (no transfers) and results obtained with our proposed ALNS (with transfers) are reported in column Proposed ALNS. In row s_r, the initial solution to our method is given by the solution found with the ALNS by Ropke and Pisinger (2006) and, in row, s_t, the initial solution is a solution in which some requests are initially transferred, as described in Section **??**. Results for an instance in each class is the average over five runs. The ALNS of Ropke and Pisinger (2006) provided the same solutions in all runs, thus we do not present Best of Average values for c or Veh.

fers have a limited value, especially for larger shift lengths ($H = 240$ and $H = 300$), starting from an initial solution with transfer is in fact detrimental to the quality of the final solution. We also note that the run time is slightly higher when starting from an initial solution with transfers, because, as noted earlier, the more time-consuming insert with transfer operations are performed more often.

Next, we consider transfer location geometry MD-5T. The results can be found in Table 3.6. The benefits are similar to what we have seen for transfer location geometry MD-4T, but for the fact that, on average, more requests are transferred, and, consequently, slightly higher CPU times are observed.

To show that our proposed ALNS algorithm scales well, we present results for instances with 75 and 100 requests in Tables 3.7 and 3.8, respectively. In each table, we show the results obtained for transfer location geometry MD-5T and using an initial solution with transfers. We observe once more that for instances in which there are requests with a long distance between pickup and delivery location (L and M) and the shift lengths are relatively short (180 and 240), the proposed ALNS algorithm is able to find solutions that reduce both the total distance and the number of vehicles used.

3.5.6. EVALUATION OF REMOVE AND INSERT OPERATORS

The performance of the remove and insert operators employed in our ALNS algorithm has been evaluated on two representative instances. The first instance, I_1, has 50 long-only requests, shift length 180, and transfer location geometry MD-4T. As we have seen, such an instance is likely to benefit from transfers. The second instance, I_2, has 50 short-only requests, shift length 300, and transfer location geometry MD-4T – an instance less likely to benefit from transfers. We analyze which operators are used throughout the search when solving these two instances and also if this depends (strongly) on the initial solution.

Recall that the likelihood of an operator being selected is proportional to its weight, where the weight is initialized and updated during each segment of κ iterations (within a segment, operators are rewarded scores based on whether or not an operator advances the search, i.e., obtains a new best solution, obtains an improving solution, or obtains a worse, but accepted solution). Therefore, to analyze the performance of an operator during the search, we report how the weight of the operator changes during the search and its contribution to advancing the search (the ratio of the number of times the operator advanced the search and the total number of times the search advanced).

Figures 3.6 and 3.7 show the progressing of the operators for instances I_1 and I_2, respectively. In each figure, we show, separately, the progression of the removal and the insertion operators. We use a stacked bar graph to show the progression of the contribution to advancing the search and a line plot to show the progression of the weight. The values are collected at the end of every 500 iterations (2κ). The graphs on the left (labeled (a) on both Figures 3.6 and 3.7) show results obtained when using an initial solution without transfers, and the graphs on the right (labeled (b) on both Figures 3.6 and 3.7) show results obtained when using an initial solution with transfers.

Inst.	RP ALNS			Proposed ALNS								
					Best				Average			
$H, \mathscr{C}$	c	Veh	t(s)	s_0	Δ_D(%)	Δ_V(%)	Trans	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)
180, L	6322.1	39.0	18.4	s_r	-46.6	-45.9	44.4	62.1	-45.1	-43.7	43.7	60.9
				s_t	-47.1	-46.2	45.2	55.8	-45.1	-43.6	44.0	60.9
180, M	4116.3	24.8	23.6	s_r	-31.2	-28.2	37.4	54.9	-29.2	-26.2	35.7	56.2
				s_t	-31.4	-29.0	37.6	55.8	-29.0	-25.6	36.2	55.9
180, S	2459.2	15.0	29.8	s_r	-4.9	-4.0	14.2	38.2	-2.7	-2.1	8.1	36.7
				s_t	-5.4	-2.7	20.5	47.2	-2.8	-0.3	15.8	44.5
240, L	3864.7	18.8	31.7	s_r	-28.5	-18.1	47.4	54.4	-23.2	-12.7	42.4	53.4
				s_t	-28.5	-17.6	46.3	55.5	-25.3	-13.8	45.3	57.6
240, M	3012.0	15.1	34.9	s_r	-15.2	-6.6	34.7	53.1	-8.4	-2.6	19.0	47.0
				s_t	-16.7	-7.3	38.4	58.9	-10.5	-2.1	29.1	53.1
240, S	2195.1	11.8	39.1	s_r	-2.2	0.0	12.5	47.6	-0.9	0.0	7.2	43.9
				s_t	-2.0	0.0	15.3	50.2	0.4	3.1	12.6	47.2
300, L	3653.1	16.6	33.8	s_r	-22.8	-5.4	46.7	56.9	-17.7	-2.2	40.9	54.3
				s_t	-23.1	-6.6	47.0	54.1	-17.5	-3.0	40.9	55.0
300, M	2952.8	14.4	35.9	s_r	-11.6	-1.4	33.5	55.0	-6.0	-1.3	17.5	45.7
				s_t	-14.4	-2.8	38.7	59.6	-9.0	1.8	30.2	53.1
300, S	2188.9	11.7	39.8	s_r	-1.3	0.0	4.3	42.3	-0.4	-0.5	2.2	41.6
				s_t	-2.2	0.9	17.9	53.1	0.9	4.4	19.1	51.7

Table 3.6: Results for instances containing 50 requests and 3h time windows in the MD-5T setting. Column RP ALNS reports results obtained with Ropke and Pisinger (2006) (no transfers) and results obtained with our proposed ALNS (with transfers) are reported in column Proposed ALNS. In row s_r, the initial solution to our method is given by the solution found with the ALNS by Ropke and Pisinger (2006) and, in row, s_t, the initial solution is a solution in which some requests are initially transferred, as described in Section **??**. Results for an instance in each class is the average over five runs. The ALNS of Ropke and Pisinger (2006) provided the same solutions in all runs, thus we do not present Best of Average values for c or Veh.

Inst.	RP ALNS			Proposed ALNS							
				Best				Average			
$H, \mathscr{C}$	c	Veh	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)
180, L	8,883.3	53.8	97.6	-50.0	-47.6	70.8	142.1	-48.0	-45.7	68.4	159.5
180, M	5,166.8	30.8	151.2	-31.0	-26.6	57.0	134.9	-27.8	-23.2	49.6	137.8
180, S	3,268.6	19.6	184.2	-3.3	2.0	42.5	127.0	-0.5	3.8	36.7	133.7
240, L	5,162.3	24.7	204.2	-29.1	-17.0	71.4	164.7	-25.9	-13.5	68.7	162.3
240, M	3,743.1	18.7	212.4	-14.2	-1.1	62.5	156.6	-9.3	0.2	46.1	146.0
240, S	2,932.6	15.5	183.6	0.6	7.7	26.4	127.6	3.3	11.6	38.0	143.5
300, L	4,941.4	22.6	190.9	-25.5	-11.9	71.8	177.0	-20.3	-6.8	63.5	158.8
300, M	3,668.5	17.8	238.0	-12.2	1.7	64.7	180.4	-7.1	4.3	47.2	161.5
300, S	2,924.9	15.7	180.6	0.1	5.7	50.1	136.2	4.5	11.0	49.8	141.4

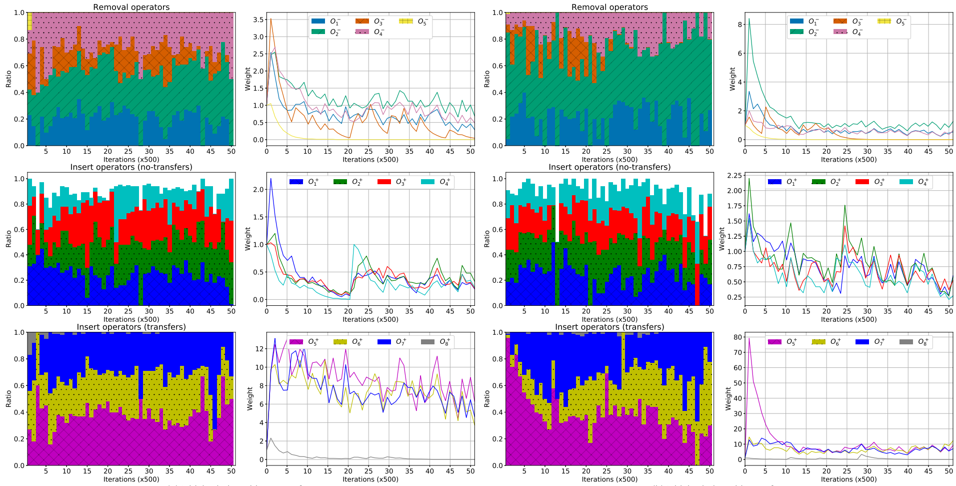
Table 3.7: Results for instances containing 75 requests and 3h time windows in the MD-5T setting. Column RP ALNS reports results obtained with Ropke and Pisinger (2006) (no transfers) and results obtained with our proposed ALNS (with transfers) are reported in column Proposed ALNS. Results for an instance in each class is the average over five runs. The ALNS of Ropke and Pisinger (2006) provided the same solutions in all runs, thus we do not present Best of Average values for c or Veh. The initial solution for the proposed ALNS is a solution with transfers (s_t).

3

Inst.	RP ALNS			Proposed ALNS							
				Best				Average			
$H, \mathscr{C}$	c	Veh	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)	Δ_D(%)	Δ_V(%)	Trans	t(s)
180, L	11613.5	69.7	163.9	-53.1	-50.5	97.6	277.6	-51.1	-48.2	95.4	300.2
180, M	7309.3	43.3	256.4	-35.4	-30.9	90.6	261.2	-35.2	-30.5	87.2	285.9
180, S	4071.9	24.0	335.3	-5.9	0.8	67.9	280.3	-2.4	3.0	50.6	280.8
240, L	6481.8	30.6	383.7	-30.9	-17.3	96.4	295.5	-27.3	-13.5	94.4	290.8
240, M	5069.5	24.9	450.4	-20.8	-7.2	93.0	310.8	-16.7	-4.3	85.9	322.1
240, S	3618.9	19.3	483.2	-3.0	6.2	73.3	298.0	0.0	6.4	53.3	294.8
300, L	6186.0	28.2	432.6	-26.0	-9.9	98.0	332.8	-22.3	-6.0	94.1	322.1
300, M	4985.4	23.7	461.2	-17.9	-1.7	93.7	386.2	-12.1	0.5	73.5	326.7
300, S	3619.7	19.1	461.5	-2.1	7.3	59.7	317.1	1.2	7.0	45.0	288.5

Table 3.8: Results for instances containing 100 requests and 3h time windows in the MD-5T setting. Column RP ALNS reports results obtained with Ropke and Pisinger (2006) (no transfers) and results obtained with our proposed ALNS (with transfers) are reported in column Proposed ALNS. Results for an instance in each class is the average over five runs. The ALNS of Ropke and Pisinger (2006) provided the same solutions in all runs, thus we do not present Best of Average values for c or Veh. The initial solution for the proposed ALNS is a solution with transfers (s_t).

3



(a) Initial solution without transfers

(b) Initial solution with transfers

Figure 3.6: Progression of operators during the search when solving instance I_1 starting from an initial solution without transfers (left) and with transfers (right). Removal operators: O_1^- – worst request; O_2^- – random request; O_3^- – route; O_4^- – transfer-based request; O_5^- – cluster. Insertion operators without transfers: O_1^+ – travel time; O_2^+ – insertion cost; O_3^+ – regret; O_4^+ – random. Insertion with transfers: O_5^+ – travel time; O_6^+ – insertion cost; O_7^+ – regret; O_8^+ – transfer.

Note that the stacked bars for the removal operators always sum up to 1, whereas some of the stacked bars for an insertion operator (with or without transfers) sum up to less than 1. This is due to the fact that two insert operators are selected at each iteration. If the first operator (with the largest weight) succeeds inserting all requests, then the second one is not used. Thus, for some iterations the search is advanced after applying only one of the selected operators.

Figure 3.6 shows that for both initial solutions, insertion operators using transfers are the ones contributing the most for advancing the search (note that the scales of the vertical axis of the weight charts are not same). The stacked bars show that, in many iterations, an insertion with transfer operator alone is able to advance the search (because in many cases the stacked bars for the insertion without transfer operators sums up to less than 1, indicating that the they were not used). The larger weights of the insertion with transfer operators also demonstrate that these operators are more successful in finding new best and better solutions. We see that insertion with transfer operator O_8^+ is mostly used at the start of the search process and more so when the initial solution does not have transfers. None of the other three insertion with transfer operators clearly dominates the others. However, we see that when the initial solution already has transfers, insertion with transfer operator 0_5^+ is able to quickly find good transfer options for the requests with long distances between pickup and delivery locations (the initial spike in the weight chart shows the operator finds new best solutions). Regarding removal opera-

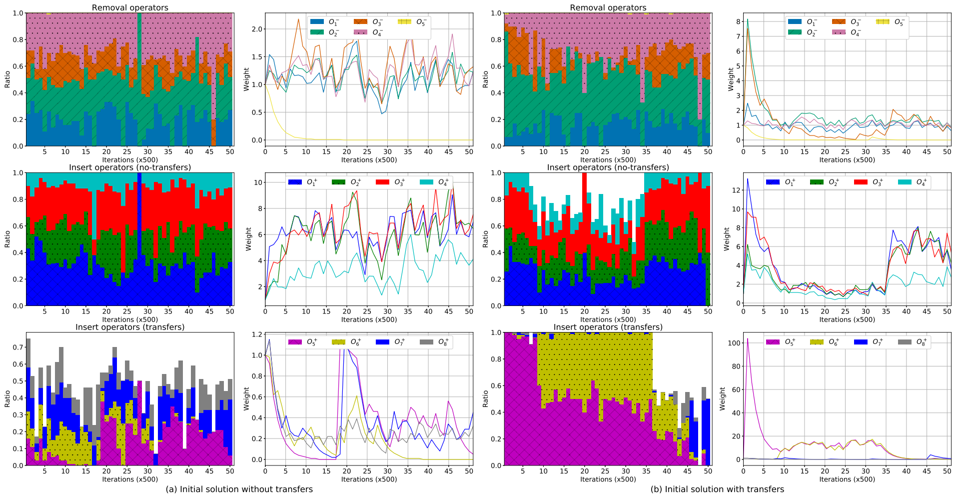


Figure 3.7: Progression of operators during the search when solving instance I_2 starting from an initial solution without transfers (left) and with transfers (right). Removal operators: O_1^- – worst request; O_2^- – random request; O_3^- – route; O_4^- – transfer-based request; O_5^- – cluster. Insertion operators without transfers: O_1^+ – travel time; O_2^+ – insertion cost; O_3^+ – regret; O_4^+ – random. Insertion with transfers: O_5^+ – travel time; O_6^+ – insertion cost; O_7^+ – regret; O_8^+ – transfer.

tors, note the important contribution of the transfer-based removal O_4^-, especially when starting with a solution without transfers.

When an instance benefits little from transfers, as is the case for instance I_2, the proposed ALNS algorithm recognizes this and adjusts accordingly (Figure 3.7). When the initial solution has no transfers, the search advances mainly using insertion without transfer operators. Even though insertion with transfer operator O_8^+ introduces a few unnecessary transfer visits early in the search, the transfer location is eventually removed from the solution. When the initial solution has transfers, improving solutions involving transfers are found in the early iterations, but as the search progresses and the insertion without transfer operators successfully insert more and more requests, the search advances primarily due to these operators (the turning point appears somewhere around iteration 17000).

3.5.7. THE IMPORTANCE OF THE ADAPTIVE LAYER

In a recent work by Turkeš et al. (2020), the authors conduct a meta-analysis to gain insights into the importance of the adaptive layer in the ALNS framework. In particular, 136 studies are assessed regarding whether or not adaptiveness actually contributes to the performance of an ALNS algorithm. Most of the works assessed consider routing or scheduling problems. The study shows that, surprisingly, the addition of an adaptive layer in an ALNS algorithm improves the objective function value by 0.14%, on average.

We conducted a similar experiment to evaluate the average improvement of the objective function caused by the adaptive layer in our ALNS implementation. To this end, we run our ALNS having the adaptive layer disabled and, thus, operators are selected (uniform) randomly throughout the execution of the algorithm, without accounting for their past performance.. We compare the results obtained over a set of 18 instances (9 with 50 requests and 9 with 75 requests) with the results obtained for the same instances solved using our ALNS with the adaptive layer enabled. For each instance, we execute the algorithms three times. With the adaptive layer enabled, average improvements in the objective value of 5.5% were observed. For some instances, the objective value improved by 10.8%. The larger improvements are observed on instances for which a particular type of operator is more suitable for the given instance characteristics. That is, on instances benefiting from transfers, the operators exploiting transfer opportunities, and for instances not benefiting from transfers, the operators without transfers. Thus, the adaptive layer is able to capture the underlying structure of the instance (benefiting or not from transfers) by selecting the more successful operators and improve the algorithmic performance. Finally, we observe that in the work by Turkeš et al. (2020), the authors highlight two outliers in the study, for which the adaptive layer provided improvements of 15.5% and 6.5%.

3.5.8. BLOCK-BASED CROWD-SOURCED TRANSPORTATION SYSTEMS

In this section, we evaluate the impact of transfers in a block-based crowd-sourced systems, i.e., system in which drivers are available for certain periods of time (blocks) during the planning. In particular, drivers commit to provide their services in one or more offered blocks, time slots of a given duration.

In the following experiments, we consider only instances with short and long requests (class M) and with only long requests (class L), as instance with only short requests (class S) do not benefit from transfers. Given a request $r = (i^+, i^-)$, the time window $[E_{i^+}, L_{i^-}]$ is generated as follows: $L_{i^-} = F + \tau_{i^+,i^-}$, where F is a number drawn uniform randomly from $[60, 480 - \tau_{i^+,i^-}]$, and $E_{i^+} = F - 60$ (i.e., there is one hour of flexibility to perform the request). The planning horizon is nine hours, and crowd-sourced drivers sign up for three-hour blocks. We consider a geography similar to the one depicted in Figure 3.4b, but without locations $m_1 - m_4$, and where crowd-sourced drivers start and end their blocks at location t_1.

Firstly, we consider a setting in which the planning horizon $[0, 540]$ is divided into three non-overlapping blocks $[0, 180], [180, 360]$, and $[360, 540]$. To assess the possible benefits provided by considering transfers, we compare the number of requests not served in a solution with and without the use of transfers (these requests would have to be served by dedicated, professional drivers). A visualization of the results can be found in Figure 3.8, which is based on 10 instances with 50 and 100 requests for class L and M (a total of 40 instances). We see, as expected, that the number of requests that cannot be served decreases when drivers are able to transfer requests. More requests with long direct travel times between pickup and delivery locations can be served using a transfer and drivers in different blocks. On average, 22% more requests can be satisfied by crowd-sourced drivers when transfers are utilized.

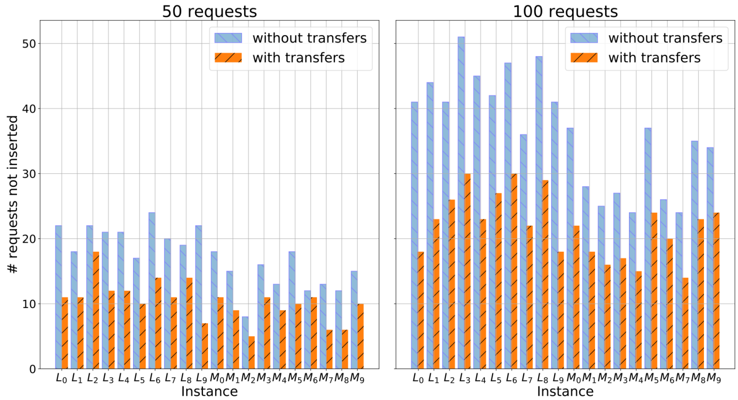


Figure 3.8: Number of requests not served when drivers sign up for blocks in the set $\{[0,180],[180,360],[360,540]\}$

Next, we consider a setting in which the planning horizon $[0,540]$ is divided into overlapping blocks $[0,180]$, $[90,270]$, $[180,360]$, $[270,450]$, and $[360,540]$. The additional flexibility provided by overlapping blocks results in all requests being served in the instances, both with and without transfers. Therefore, to assess the benefits of transfers, we focus on the number of drivers required in each of the blocks. Figure 3.9 shows results for two specific instances, one in class M and the other in class L, both with 100 requests. For each instance, the figure shows the number of open requests in the system at time t (x-axis), where a request $r = (i^+, i^-)$ with time windows $[E_{i^+}, L_{i^-}]$ and $E_{i^+} \leq t \leq L_{i^-}$ is considered open, and the number of drivers used in a solution with and without the use of transfers. (Additional results, in which we average over all instances in a class, can be found in the appendix.) The two instances, however, are representative of the results observed.

First, we see that fewer drivers are needed when sharing is allowed, especially when there are only long requests. Second, we see that when transfers are allowed, the number of drivers in the system in the first block is larger than when transfers are not allowed. This signals that when transfers are allowed, many requests are picked up in the first block and brought to transfer points to be consolidated and delivered in subsequent blocks (which requires more drivers in the first block). As a consequence, fewer drivers are required in subsequent blocks, as it takes less time to deliver consolidated requests. As a result, which may be advantageous, the number of drivers required across the blocks is more balanced.

Figure 3.9: Instances with mixed (M) and long (L) requests and the required crowd-sourced capacity for service with and without the use of transfers.

3.6. FINAL REMARKS

Our research has focused on investigating the potential benefits of using transfers in urban freight delivery systems. Our results show that these benefits can be significant, especially in settings where pickup and delivery locations are relatively far apart and driver shifts are relatively short. Thus, transfers may be especially valuable for urban freight delivery systems that rely, in part or completely, on crowdsourced delivery capacity, because crowdshippers tend to work for relatively short periods of time. Our results also show that the ability to transfer requests can substantially reduce the number of drivers required to serve a given number of requests. This too may be especially valuable in urban freight delivery systems that rely on crowdsourced delivery capacity, because crowdshippers are often compensated based on the number of requests they serve, and when the number of requests served per driver increases, it will become easier to attract crowdshippers to the system.

4

Vehicle Routing with Roaming Delivery Under Stochastic Travel Times

In the face of ambiguity, refuse the temptation to guess.
There should be one– and preferably only one –obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.

Tim Peters, The Zen of Python

We address a stochastic variant of the Vehicle Routing Problem with Roaming Delivery Locations. In this model, direct-to-consumer deliveries can be made in the trunk of the customer's car, while the vehicle is parked at a location along the customer's itinerary. The stochasticity arises from the uncertainty in travel times and the problem is formulated as a two-stage stochastic model. We propose a scenario-based sample average approximation to obtain a heuristic solution. Several experiments to assess the effect of our solution approach compared to a pure deterministic solution approach, using expected travel times, show that a cost savings of on average more than 30% can be obtained. Furthermore, it is shown that the flexibility provided by using alternative roaming delivery locations as a recourse to avoid missed deliveries can provide, on average, costs savings of 25% compared to a recourse staying with the locations chosen in the a priori first stage plan.

4.1. INTRODUCTION

PRivate cars represent a large share of the total flow of vehicles within urban areas. Given current population and urbanization growth, they will likely continue to have a prominent role in the fulfillment of transportation activities for years to come. However, some studies reveal that the average car is parked most of the time and only being used for a relatively short period – parked at home for 80% of the time, 16.5% elsewhere and moving for 3.5% (Bates and Leibling, 2012). Improving the period of time when the vehicle is actually moving e.g., minimize travelled distance to reduce costs/emissions, was the focus of attention for most academic studies. However, a comprehensive view seeking for more cost-effective, sustainable solutions to minimize flows of goods and people needs to leverage all available resources at all times.

4

Trunk delivery refers to using the trunk of a customer's car, when the vehicle is parked, as a possible location for the final delivery of goods to the consumer. As indicated, cars are parked away from home for a significant amount of time, thus providing opportunities that could be seized to avoid, for example, missed deliveries and, consequently, multiple visits at home. In fact, given the explosion of e-commerce and on-line shopping, the last-mile supply chains of even the largest e-tailers are strained by the sheer volume increase of direct-to-consumer orders. This challenge is even amplified by the increased customer service levels offered by e-tailers to compete against the instant gratification of brick-and-mortar stores. Companies are evaluating new and innovative business models, such as trunk delivery, that could help improving last-mile operations. As an example, Amazon recently introduced a free in-car delivery service for some of its customers. In a partnership with a selection of car manufacturers, the service connects to the vehicle allowing the Amazon delivery person an one-time access to the trunk of the customer's car. Whereas neither manufacturers nor Amazon expects monetary gains from the service, they see it as an added delivery option to market (Hawkins, 2018).

The Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) is introduced by Reyes et al. (2017). In the VRPRDL, each customer has a given, fixed, itinerary specifying one or more locations where his/her car will be parked, within corresponding time windows i.e. a roaming delivery location. While the work by Reyes et al. (2017) showed the benefits of trunk delivery, specially combined with traditional home delivery, a fully deterministic environment is assumed. Nevertheless, companies may face many uncertain disruptions that could affect travel times e.g., accidents, vehicle breakdowns, weather and, consequently, hinder the fulfillment of a delivery within the time windows during which the customer's car will be parked at a given location. Moreover, since a customer is also moving from one location to another along his/her itinerary, the time during which the customer's vehicle will be available for trunk delivery at a given location might also be delayed due to increased travel times. Stochastic travel times in the VRPRDL is first considered by Lombard et al. (2018). The authors propose a Monte-Carlo method in which many deterministic VRPRDL instances, one for each travel time realization which is sampled out of the considered travel time distribution, are solved until the variation of the costs found is marginal. A frequency analysis is performed to assess the (probability of) occurrence of the best solutions and the costs of the most frequent solutions found. However, for contexts where the routing plan has to be decided before travel times are known (e.g. to ensure correct and timely loading of the delivery vehicles), such an approach might yield very poor solutions since the routing plans computed in the method are optimized towards a particular travel time matrix, which might not necessarily realize.

In this work, we also consider the VRPRDL and Stochastic Travel Times (VRPRDL-STT). We assume a context in which the service provider has to decide on the operational routing plans before the uncertain travel times are revealed. This is necessary to ensure all items to be loaded into the correct delivery vehicles. We propose tackling the problem as a two-stage, stochastic optimization problem. Conceptually, in the first stage, we obtain the *a priori* VRPRDL routing plan (also known as first-stage decision) which abides to the time windows of any location visited, considering deterministic travel time values. In the second stage, travel times are revealed and the a priori plan is modified by a given recourse policy whenever a failure, i.e. a late visit to a customer's location, occurs. The objective is to obtain an a priori routing plan minimizing both the expected travel costs (routing) and the costs incurred in the second stage following the recourse policy. For many different types of Vehicle Routing Problems with Time Windows (VRPTW), servicing a location outside its associated time windows is acceptable, to some extent, although incurring in waiting time in case of early arrivals and a penalty in case of late arrivals. For the VRPRDL, the delivery locations, the trunk of a car, are only present within a given time window. Waiting is allowed, but late arrivals are not feasible, since the car will not be present anymore at the location. We propose different recourse policies to recover feasibility in the a priori plan given realized travel times in the second-stage. For example, if due to the realized travel times a particular customer location cannot be visited within its corresponding time windows, another location in the customer itinerary might be selected.

To solve the two-stage problem, we propose a solution framework which combines an extension of the local search heuristic devised by Reyes et al. (2017), to account for uncertain travel times, with an implementation of the Sample Average Approximation (SAA) approach proposed by Kleywegt et al. (2002). We assume that travel times follow a known probability distribution, allowing for the generation of (a very large number of) different possible *scenarios*. The SAA method is an iterative procedure that solves the problem restricted to a small set of scenarios on each iteration and evaluates the obtained solution over a (considerable) larger set of (independent) scenarios to approximate the expected objective function value. We note that, even if the chosen probability distribution of the travel time follows the convolution property, integrating the evaluation of the recourse policies proposed in this work in an exact approach is challenging. The SAA method is suitable to deal with extremely large scenarios sets, thus able to approximate the optimal value of the stochastic problem. Moreover, exact and heuristic solution approaches can be applied to solve each restricted problem in a SAA iteration.

The contributions of this research are:

- We consider a last-mile application with roaming locations, where the company is able to access the trunk of a customer car, and taking into account stochastic travel times. We present a two-stage stochastic formulation with recourse to model the problem. We propose a scenario-based stochastic approximation (SAA) method to solve the problem combined with an adaptation of the local search heuristics proposed in Reyes et al. (2017) considering recourse costs to evaluate neighborhood solutions. In our computational experiments, we explicitly evaluate implementation decisions taken into the design of the SAA method (choice of sample size to estimate better lower bounds, trade-off between solution quality and computational time, number of iterations).

- We exploit the flexibility introduced by trunk-delivery i.e., a delivery can be made at different locations at different times. Specifically, we define a variety of recourse policies reducing the number of failed deliveries caused by travel time stochasticity.

- We show the benefits of using a stochastic solution approach over the deterministic counterpart are twofold. First, the evaluation of different travel time scenarios in the a priori stage allows for a better assessment of the customer availability and to hedge against the travel time uncertainty. Results show that, compared to a deterministic solution approach, which takes average travel times as input, the proposed methodology provides savings of, on average, more than 30%. Secondly, exploiting the flexibility to visit customers at other locations than planned for in the a priori plan, leads on average to cost savings of 25%.

The remainder of the chapter is organized as follows. Section 4.2 reviews some of the works related to the VRPRDL as well as issues regarding the consideration of stochastic elements in vehicle routing problems, in particular stochastic travel times. The mathematical notation used throughout the text is introduced in Section 4.3, as well as a formal problem definition and the mathematical programming formulation proposed by Reyes et al. (2017), considering deterministic travel times. We extend the formulation to a two-stage stochastic model to account for uncertainty in travel times. In Section 4.4, we describe our implementation of a scenario-based Sample Average Approximation. Section 4.5 presents computational experiments carried out to evaluate the proposed methodology and to assess the gains of considering uncertainty in travel times when planning the schedules of drivers in the context of trunk delivery. Finally, our conclusions and directions for further research are presented in Section 4.6.

4.2. Related Literature

The VRPRDL was first introduced in Reyes et al. (2017). The authors developed construction and improvement heuristics and assessed the benefits of trunk delivery, showing that a reduction of up to 50% on total travelled distance can be achieved for certain environments. A followup work by Ozbaygin et al. (2017) was the first to tackle the problem with an exact approach, introducing a Branch-and-Price method for the VRPRDL able to solve instances of up to 120 customers. In both works, all relevant information (travel times, customers, demands) are assumed to be deterministic and not changing over time (static). More recently, Ozbaygin and Savelsbergh (2019) proposed a dynamic but deterministic variant of the problem in which customers' planed routes for the entire service day are known to the service provider beforehand, but can change during the execution of the delivery schedules. In particular, the authors assume that a customer always visits the locations in his/her planned route for the day but arrival and departure times at these locations might deviate from those previously informed to the provider.

The problem relates to both the VRPTW and to the Generalized Vehicle Routing Problem (GVRP). For the former, the reader is referred to Toth and Vigo (2014) for an extensive coverage of many problem variants and state-of-the-art methods. The latter was introduced by Ghiani and Improta (2000) and consists of a generalization of the VRP for which the set of customers is partitioned in clusters and exactly one customer of each cluster has to be visited in the solution. The reader is referred to Bektaş et al. (2011) for formulations and an exact method to solve the GVRP and to Kovacs et al. (2015) for an application of the problem and a heuristic solution approach. Moccia et al. (2012) introduced the GVRP with Time Windows (GVRPTW) and proposed a tabu search heuristic to solve the problem. Observe that the GVRPTW reduces to the VRPRDL when all the locations of a given cluster define the customer's itinerary and the associated time windows of each location within a cluster are non-overlapping. The VRPRDL also relates to a problem recently proposed by Gambella et al. (2018), the Vehicle Routing Problem with Floating Targets. The problem is a dynamic variant of the VRP, and the objective is to obtain a route schedule visiting a set of target points that can freely move in the plane. As an example, the target points are ride-sharing users who share a common destination. Instead of agreeing on meeting at a fixed location (e.g., home), the driver might meet an individual asking for transport at a location midway on the route from the fixed location to the common destination.

Recently, increased attention has been given to numerous stochastic variants of the VRP(TW). The uncertain events addressed in most of the works comprise customers availability, when the presence or absence of a customer is a random event; demand volumes, when the exact amount of a commodity to collect or deliver is unknown; operation times, when travel or service times, for example, are considered stochastic. Gendreau et al. (2016) provide an overview of the state-of-the-art for those main classes of stochastic VRPs, the modelling issues and the exact and approximate methods that have been proposed to solve them. The uncertainty on travel times addressed in this work stem mainly from unexpected disruptions affecting expected values e.g., increased traffic due to accidents or weather conditions. Expected variations on travel times such as due to congestion during peak-hours (e.g., early morning and late afternoon rush) can be tackled using deterministic models by considering time-dependent travel times. The reader is referred to Gendreau et al. (2015) for an overview of such problems.

Usually, a VRPTW with stochastic travel times (VRPTW-STT) takes into account deviations from customer's time windows in the computation of the expected recourse costs. In problems with soft time windows, late arrivals are allowed in the routing plan, but usually incurring in a penalty proportionally to their tardiness and a recourse action is not necessary. As observed by Gendreau et al. (2016), this allows for the use of closed-form expressions to compute the expected total penalty of a route – i.e., the expected earliness and tardiness – if the probability distribution follows the convolution property. A shifted gamma distribution (Erlang) is considered by Russell and Urban (2008). The authors propose different functions to evaluate penalty costs (fixed, linear, quadratic) and a tabu-search heuristic to solve the problem. Li et al. (2010) considered both stochastic travel and service times modeled by a normal distribution with parameters (mean, variance) depending on the arcs and customers, respectively. The authors propose a chance constrained programming and a stochastic programming model and solve both models with an adaptation of a tabu-search heuristic. Soft time windows are also considered by Taş et al. (2013). The authors assume the travel time of one unit of distance as a random variable with gamma distribution by given parameters (scale, shape). The travel time for an arc is then obtained by scaling the unit travel time with respect to the arc distance. A tabu-search heuristic is proposed to solve the problem and the authors consider different coefficients of variation for the travel time per unit distance to assess solutions obtained regarding variability. The same problem is solved exactly through a branch-and-price method by Taş et al. (2014b) and under time-dependency of the travel times by Taş et al. (2014a). Deadlines for visiting (a subset of) customers under stochastic travel times are considered by Adulyasak and Jaillet (2016). The problem is extended to consider soft time windows, and a branch-and-cut framework is proposed.

For problems with hard time windows, such as the VRPRDL(STT), late arrivals at customers are not allowed, requiring a recourse action to be applied to recover feasibility. Modelling expected arrival times at customers cannot be done, generally, by applying convolution properties of the distribution used to model travel times, as hard windows tend to truncate the distribution. Only a few works have addressed problems with hard-time windows and stochastic times and, due to the difficulty in computing expected times, the recourse policies proposed are usually restrictive e.g., assume that only one disruption occurs in a route, and only one recourse action is necessary (Errico et al., 2016). A VRPTW with stochastic travel times and stochastic demands is considered by Branda (2012), where the problem was modelled as a stochastic programming problem with chance constraints and a sample average approximation technique was used to derive estimates on the sample sizes required to obtain a feasible solution. Normal distributed travel times are considered by Ehmke et al. (2015) and a chance-constrained model is proposed in order to guarantee a given service level for customers (a probability on respecting the time windows). Those features can be embedded in any algorithm for the VRPTW with stochastic travel times and demands, and the authors show how to apply them on a tabu search heuristic. Binart et al. (2016) address a variation of the stochastic VRPTW, considering both stochastic travel and service times, in which customers are split in mandatory and optional. The first have to be served within their time windows whereas the latter can be serviced at any time during the planning horizon or not be serviced at all. The objective is to minimize total travel time servicing as many optional customers as possible. Stochastic travel and service times are dealt through dynamic programming where optional customers are used as buffers to hedge against the variations on those elements. Finally, Miranda and Conceição (2016) propose a statistical model to compute the cumulative probability function for the arrival times over customers when travel times are normally distributed.

Only a few works addressing stochastic events in the context of the GVRP(TW) exist. Similarly to the work by Laporte et al. (2002) on the VRP with stochastic demands, Biesinger et al. (2016) consider a recourse policy which consists of returning trips to the depot whenever a failure – a stock-out – occurs and propose an Integer *L*-Shaped Method (Laporte and Louveaux, 1993) and branch-and-cut framework to solve a GVRP with stochastic demands.

In a related stream of literature, dynamic and online routing problems are considered, where the information (e.g., customer availability, travel time, demand, etc) is revealed concomitantly to the operating period, requiring (almost) immediate response to update the routing and scheduling decisions taken so far. In particular, *same-day delivery* problems for online purchases are a recent trend, where customers place orders on the same day that they should be delivered. The reader is referred to Voccia et al. (2019) where the authors consider a same-day delivery problem and identify when it is beneficial for the vehicles to stay at the depot, waiting for more information (customer requests). In the same vein, Klapp et al. (2018) conduct a study on the trad-offs in same-day delivery operations by considering dispatching waves of vehicles at pre-defined times. For a general overview of dynamic routing problems, the reader is referred to Pillac et al. (2013a).

Most of the works in the literature focus either on solving a (static) stochastic problem, as in our work, or a (deterministic) dynamic problem. Only a few works exist in which stochastic information is used within a method to solve a dynamic problem. Bent and Van Hentenryck (2004) showed that dynamic routing approaches can benefit from including stochastic information when decisions are taken concomitantly to the execution of the routes. Our work contributes to the stochastic VRP literature by considering the stochastic version of a novel VRP application. Our results indicate the benefits of stochastic information even for static contexts, and provide a baseline for assessing the dynamic version of the problem. Moreover, the recourse policies described in the next section can also be used within dynamic contexts.

4.3. PROBLEM DESCRIPTION

We first provide a general description of the VRPRDL and the MIP formulation proposed by Reyes et al. (2017), adapted to our notation, in Sections 4.3.1 and B, respectively. In Section 4.3.3, we consider that travel times are uncertain and define the VRPRDL with Stochastic Travel Times (VRPDL-STT). We then extend the MIP formulation proposed by Reyes et al. (2017) and describe a two-stage stochastic formulation for the VRPDL-

STT.

In Section 4.3.3, we consider that travel times are random variables and describe a two-stage stochastic formulation for the VRPRDL with Stochastic Travel Times (VRPDL-STT).

4.3.1. The VRP with Roaming Delivery Locations

In the VRPRDL, a set of customers $C = \{1,2,...,n\}$ is serviced by a homogeneous fleet of vehicles having a fixed capacity, Q, operating during the planning period $[0,T]$. Each customer $c \in C$ has a non-negative demand d_c and is associated with a unique set of locations N_c, specifying the itinerary of customer c i.e., the locations and times at which his/her car is available for trunk delivery. Assuming $N_c = \{i^c_1, i^c_2, ..., i^c_{|N_c|}\}$ and that $i^c_1 = i^c_{|N_c|}$ represent the home location for any customer c, his/her itinerary is defined by the directed graph $G(N_c, A_c)$, where $A_c = \{(i^c_1, i^c_2), (i^c_2, i^c_3), ..., (i^c_{|N_c|-1}, i^c_{|N_c|})\}$. A delivery vehicle servicing a customer $c \in C$ visits exactly one of the delivery locations in N_c i.e., does not traverse arcs in $A_C = \bigcup\limits_{c\in C} A_c$. Let $N = \left(\bigcup_{c\in C} N_c\right) \cup \{0, n+1\}$ be the set of *service locations*, where 0 is the source depot and $n+1$ the target depot of the vehicles. The route of a vehicle is defined over the directed graph $G(N, A_R)$, where $A_R = \Big(\bigcup\limits_{\substack{c_1,c_2\in C\\ c_1\neq c_2}} (N_{c_1} \times N_{c_2})\Big) \cup \Big(\bigcup\limits_{c\in C} (\{0\} \times N_c)\Big) \cup \Big(\bigcup\limits_{c\in C} (N_c \times \{n+1\})\Big)$.

A non-negative travel time, t_{ij}, and cost w_{ij} (e.g. distance) for traversing the arc are associated with each arc $(i,j) \in A = A_C \bigcup A_R$, and arc's travel times are the same for customers and delivery vehicles. Servicing customer $c \in C$ can occur at any of the locations $i \in N_c$, but only during a location-dependent time-interval $[a_i, b_i]$. In particular, the time windows for any customer $c \in C$ are non-overlapping, defining an ordering on the location set N_c, and are defined as:

$$a^c_1 = 0; \qquad b^c_{|N_c|} = T \tag{4.1}$$

$$a^c_\ell = b^c_{\ell-1} + t_{i^c_{\ell-1}, i^c_\ell} \quad \forall \ell = 2, ..., |N_c| \tag{4.2}$$

that is, a customer starts at home and moves from one location to another in his/her itinerary throughout the planning period $[0,T]$, being available for trunk delivery at location $i^c \in N_c$ when the car is parked (not moving), between times a_i and b_i. It is assumed that, for a pair of distinct customers $c, c' \in C$, $N_c \cap N_{c'} = \emptyset$, i.e., customers do not share locations. This assumption is readily satisfied by introducing duplicate locations when necessary. Since there exists a unique correspondence between a location and a customer, we define the function $\pi : N \setminus \{0, n+1\} \mapsto C$ to map a location $i \in N$ to the unique customer it belongs to, i.e. $i \in N_{\pi(i)}$. All routes start at the source depot, 0, and end at the target depot, $n+1$, within the planning period i.e., $a_0 = a_{n+1} = 0$ and $b_0 = b_{n+1} = T$, and should not exceed vehicle capacity. A solution to the VRPRDL constitutes a set of routes in which each customer, $c \in C$, is serviced exactly once within the time windows of the selected location for service, $i \in N_c$. The objective is to find a set of routes minimizing the total costs e.g., distance travelled. By definition, the problem generalizes to the well-known Generalized VRP with time window constraints. Moreover, when $|N_c| = 1$ for all $c \in C$, the problem reduces to a standard Capacitated VRP with Time-Windows. It follows that the VRPRDL belongs to the class of NP-Hard problems.

Throughout this paper, the following notation is used. Given a subset of vertices $S \subseteq V$, the cutset $\delta(S)$ denotes the set of edges with exactly one endpoint in S. The cutset $\delta^+(S)$ denotes the set of directed edges (arcs) having their tail in S and their head not in S. Similarly, the cutset $\delta^-(S)$ denotes the set of edges with their head in S and their tail outside S. For undirected edges $\delta(S) = \delta^+(S) = \delta^-(S)$. A summary of the notation used throughout this paper is given in Table 4.1.

Parameter	**Description**
C	Set of customers
N_c	Set of locations for customer $c \in C$
Q	Vehicle capacity
d_c	Demand of customer $c \in C$
t_{ij}	Travel time from i to j
w_{ij}	Cost to travel from i to j
$[a_i, b_i]$	Time window associated with location $i \in N$

Table 4.1: Mathematical notation used throughout the text

4.3.2. DETERMINISTIC MIP MODEL

To unambiguously define the problem, we re-state the MIP model proposed by Reyes et al. (2017), adapted to our notation. The model uses binary routing variables x_{ij}, indicating whether a delivery vehicle traverses arc $(i,j) \in A_R$, continuous start time variables τ_c to record the service start time of customer $c \in C$, and continuous load variables y_c to track the vehicle capacity remaining after visiting customer $c \in C$.

$$\min \sum_{(i,j)\in A_R} w_{ij} x_{ij} \tag{4.3}$$

$$\text{s.t.} \sum_{(i,j)\in\delta^+(N_c)} x_{ij} = 1 \qquad \forall c \in C \tag{4.4}$$

$$\sum_{(j,i)\in\delta^-(N_{\pi(i)})} x_{ji} = \sum_{(i,j)\in\delta^+(N_{\pi(i)})} x_{ij} \qquad \forall i \in N \setminus \{0, n+1\} \tag{4.5}$$

$$y_c \geq y_{c'} + d_{c'} - Q\Big(1 - \sum_{i\in N_c}\sum_{j\in N_{c'}} x_{ij}\Big) \qquad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \tag{4.6}$$

$$\tau_{c'} \geq \tau_c + \sum_{i\in N_c}\sum_{j\in N_{c'}} t_{ij} x_{ij} - T\Big(1 - \sum_{i\in N_c}\sum_{j\in N_{c'}} x_{ij}\Big) \qquad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \tag{4.7}$$

$$\sum_{i\in N_c} a_i \sum_{(i,j)\in\delta^+(N_c)} x_{ij} \le \tau_c \le \sum_{i\in N_c} b_i \sum_{(i,j)\in\delta^+(N_c)} x_{ij} \qquad \forall c\in C \qquad (4.8)$$

$$x_{ij}\in\{0,1\} \qquad \forall (i,j)\in A_R \qquad (4.9)$$

$$0\le y_c\le Q-d_c \qquad \forall c\in C \qquad (4.10)$$

$$0\le \tau_c\le T \qquad \forall c\in C \qquad (4.11)$$

The objective function minimizes the total travel cost for the vehicles. Constraints (4.4) ensure that all customers are serviced and constraints (4.5) ensure that flow conservation is preserved over all locations. Capacity constraints are imposed by inequalities (4.6) – the sum of customer demands served in a single trip cannot exceed the vehicle capacity. Inequalities (4.7) guarantee that the correct amount of driving time is spent between servicing customers c and c'. Finally, service within one of the availability time windows of the customer is imposed by (4.8). A vehicle is allowed to wait if it arrives early at a customer location, but it cannot be late.

4

4.3.3. THE VRPRDL WITH STOCHASTIC TRAVEL TIMES

The VRPRDL-STT considered in this work can be defined as follows. We assume that the service provider takes actions at two distinct times. At planning time, the provider has to decide on a set of *a-priori* routes, each starting and ending at the depot, servicing each customer exactly once within the time windows of the selected location for service, that is, a solution to the VRPRDL. After the *a-priori* routing plan is defined, it is carried out during the operational time. When deciding the set of *a-priori* routes, the provider uses deterministic information regarding travel times (e.g., assuming there is no traffic). Obviously, travel time variations might occur during operational time. Consequently, customers might not be serviced within their corresponding time windows or vehicles might return late at the depot (overtime), when vehicles follow the *a-priori* plan. Let the *a-posteriori* routing be the set of routes obtained at operational time which resulted from the *a-priori* plan being modified by the service provider, as a result of the revealed travel times, in order to minimize the costs incurred by not serviced customers and drivers overtime. Thus, the service provider targets the design, at planning time, of *a-priori* routes which, on expectation, requires the least cost set of modifications during operational time, when travel times are revealed

Conceptually, the decision on the *a-priori* routing plan, taken before the uncertain travel times are known, is a first-stage decision. After uncertainty is revealed, the *a-priori* routes can be modified by second-stage actions, or recourse actions. For example, one possible recourse action (in the next section, we elaborate more on all recourse actions considered in this paper) could be to deliver to a different location within the customer's itinerary in case revealed travel times render it to be infeasible visiting the customer within the time windows of the location chosen in the *a-priori* route. Note, however, that a customer's vehicle moves from one location to another in his/her itinerary and is subjected to the *same* variations in travel times experienced by the delivery vehicles. In particular, the extent of a disruption along the arcs in the itinerary, A_c, of a customer

$c \in C$ might change the availability for service in the locations $i \in N_c$ i.e., change the time windows $[a_i, b_i]$. Let $\delta_{ij} \geq 0$ be the length of the disruption in arc $(i, j) \in A$, $\tilde{\omega}_{ij} = t_{ij} + \delta_{ij}$ the realized travel time during the second-stage and $[\tilde{a}_i, \tilde{b}_i]$ the time window for location i after disruptions realize. As per example, after the travel times are revealed in the 2nd stage, disruptions in the customer's itinerary could have caused the customer to arrive late at location $i \in N_c$, i.e. at time $\tilde{a}_i > a_i$. Consequently, the time window during which the customer can be serviced at location $i \in N_c$ shrinks or the location becomes unavailable for servicing in the extreme case that $\tilde{a}_i > b_i$. Figure 4.1 illustrates the itinerary of a customer $c \in C$, with four locations $N_c = \{i_1^c, i_2^c, i_3^c, i_4^c\}$. In the example, location i_2 is not available for delivery anymore in the second stage. The time window for location i_1 remains the same and the time windows for locations i_3 and i_4 are shortened.

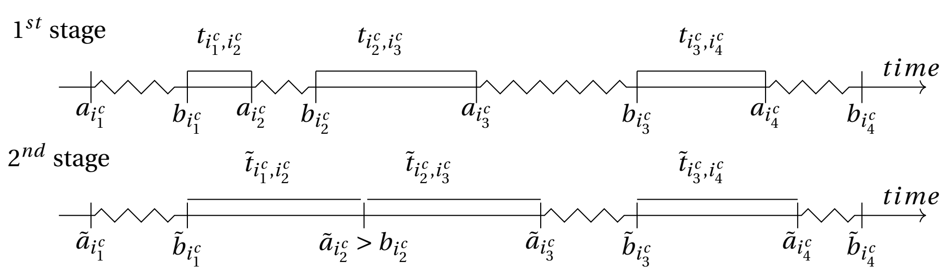


Figure 4.1: Changes in location's time windows for a particular travel time scenario. $\tilde{\omega}_{i,j} = t_{i,j} + \delta_{i,j}$. Squiggly line segments correspond to time windows during which the customer can be visited at a given location.

A TWO-STAGE STOCHASTIC MODEL

We now extend the the MIP formulation (4.3) – (4.11) to include stochastic travel times. First, let $\boldsymbol{\omega} = (\omega_{ij})_{(i,j)\in A}$ be a random variable vector, where ω_{ij} is the stochastic travel time for traversing arc (i, j), and $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_{ij})_{(i,j)\in A}$ is a particular realization of $\boldsymbol{\omega}$. To represent recourse actions taken after the realization of the travel times (second-stage variables), we use similar variables as proposed by Hvattum et al. (2006) for a VRP with stochastic customers. Let x^+_{ij}, $(i, j) \in A_R$, be a binary variable indicating whether ($x^+_{ij} = 1$) a vehicle traverses arc (i, j) in the second-stage but not in the first-stage solution $\mathbf{x} = (x_{ij})_{(i,j)\in A_R}$. Similarly, let x^-_{ij}, $(i, j) \in A_R$ be a binary variable indicating whether ($x^-_{ij} = 1$) a vehicle traverses arc (i, j) in the first-stage but not in the second-stage solution, after recourse. The continuous variable τ^+_c track the (potentially new) service time of customer $c \in C$ after recourse. We now let x^+_{ij}, x^-_{ij} and τ^+_c all depend on the random event $\boldsymbol{\omega}$ to emphasize that those decision variables are defined for each travel time realization vector $\tilde{\boldsymbol{\omega}}$. Similarly, let $[a_i(\boldsymbol{\omega}), b_i(\boldsymbol{\omega})]$ be the time windows for location $i \in N$ as a function of the random travel time variable $\boldsymbol{\omega}$ and denote by $[\tilde{a}_i, \tilde{b}_i]$ the time windows for a realized vector $\tilde{\boldsymbol{\omega}}$.

The structure used to define the recourse actions in the second-stage is said to be relatively complete if for every first-stage solution and every possible realization of random data, the second-stage problem is feasible. Observe that the structure $(\mathbf{x}^+, \mathbf{x}^-)$ does not provide relatively complete recourse, as some customers might not be serviced in time at any of his/her locations depending on the realized arc travel times. However, a model with relatively complete recourse can be achieved if skipping service for some customers is allowed, incurring in a high penalty (Hvattum et al., 2006). Let $\lambda_c(\boldsymbol{\omega})$ be a binary variable indicating whether customer $c \in C$ is skipped ($\lambda_c(\boldsymbol{\omega}) = 1$) in the second stage solution, and Λ_c the penalty for skipping customer c. When vehicles return to the depot, each unit of time exceeding the time horizon, T, incurs in a penalty cost of β. Moreover, let binary variables $u_{c,c'}$ indicate whether customers $c, c' \in C$ are serviced by the same first-stage route.

$$\min \sum_{(i,j)\in A} w_{ij} x_{ij} + \mathbb{E}[R(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-, \boldsymbol{\tau}^+, \boldsymbol{\lambda}, \boldsymbol{\omega})] \tag{4.12}$$

s.t. (4.4) – (4.11)

$$\sum_{(i,j)\in\delta^+(N_c)} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) + \lambda_c(\boldsymbol{\omega}) = 1 \qquad \forall c \in C \tag{4.13}$$

$$\sum_{(j,i)\in\delta^-(N_{\pi(i)})} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) = \sum_{(i,j)\in\delta^+(N_{\pi(i)})} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) \qquad \forall i \in N \setminus \{0, n+1\} \tag{4.14}$$

$$\tau^+_{c'}(\boldsymbol{\omega}) \geq \tau^+_c(\boldsymbol{\omega}) + \sum_{i\in N_c}\sum_{j\in N_{c'}} \omega_{ij}(x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) - T(1 - \sum_{i\in N_c}\sum_{j\in N_{c'}} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega}))) \qquad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \tag{4.15}$$

$$\tau^+_c(\boldsymbol{\omega}) \geq \sum_{i\in N_c} a_i(\boldsymbol{\omega}) \sum_{(i,j)\in\delta^+(N_{\pi(i)})} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) \qquad \forall c \in C \tag{4.16}$$

$$\tau^+_c(\boldsymbol{\omega}) \leq \sum_{i\in N_c} b_i(\boldsymbol{\omega}) \sum_{(i,j)\in\delta^+(N_{\pi(i)})} (x_{ij} + x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) \qquad \forall c \in C \tag{4.17}$$

$$x^+_{ij}(\boldsymbol{\omega}) + x^-_{ij}(\boldsymbol{\omega}) \leq 1 \qquad \forall (i,j) \in A_R \tag{4.18}$$

$$c, c' \text{ on the same first-stage route} \iff u_{c,c'} = 1 \qquad \forall c, c' \in C \tag{4.19}$$

$$x^+_{ij}(\boldsymbol{\omega}) \leq u_{\pi(i),\pi(j)} \qquad \forall (i,j) \in A_R,\ i \neq 0,\ j \neq n+1 \tag{4.20}$$

$$x^+_{ij}(\boldsymbol{\omega}) \in \{0,1\} \qquad \forall (i,j) \in A_R \tag{4.21}$$

$$x^-_{ij}(\boldsymbol{\omega}) \in \{0,1\} \qquad \forall (i,j) \in A_R \tag{4.22}$$

$$0 \leq \tau^+_c(\boldsymbol{\omega}) \leq T \qquad \forall c \in C \tag{4.23}$$

$$\lambda_c(\boldsymbol{\omega}) \in \{0,1\} \qquad \forall c \in C \tag{4.24}$$

$$u_{c,c'} \in \{0,1\} \qquad \forall c, c' \in C \tag{4.25}$$

where

$$R(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-,\boldsymbol{\tau}^+,\boldsymbol{\lambda},\boldsymbol{\omega}) = \sum_{(i,j)\in A_R} w_{ij}(x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega})) + \sum_{c\in C} \Lambda_c \lambda_c(\boldsymbol{\omega}) \qquad +$$
$$\beta \sum_{c\in C}\sum_{i\in N_c} \max\{0, \tau^+_c(\boldsymbol{\omega}) + \omega_{i0} - T(1 - \sum_{i\in N_c}(x_{i0} + x^+_{i0}(\boldsymbol{\omega}) - x^-_{i0}(\boldsymbol{\omega}))) - T\}$$

The objective is to minimize total routing cost for the first stage solution and expected (random) recourse costs $R(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-,\boldsymbol{\tau}^+,\boldsymbol{\omega})$, encompassing the cost of changes required in the first stage routing and penalties for skipping customers and overtime after recourse. Constraints (4.13)-(4.14) ensure that customers are either visited or skipped in the second stage solution. Cost Λ_c, $c \in C$, should be set to a high value, such that skipping customer c in the second-stage only occurs when it is not feasible to service c within its corresponding time windows. Observe that time windows for customer locations in the second stage, $a_i(\boldsymbol{\omega})$ and $b_i(\boldsymbol{\omega})$, depend on the realization of the random vector $\boldsymbol{\omega}$ and are, thus, random variables as well. Inequalities (4.15)-(4.17) ensure that service times for not skipped customers in the second stage occur within their corresponding time windows, $\tilde{a}_i$ and $\tilde{b}_i$, given the revealed travel times $\tilde{\boldsymbol{\omega}}$. Inequalities (4.18) are added to make the definitions of second stage variables $x^+_{ij}(\boldsymbol{\omega})$ and $x^-_{ij}(\boldsymbol{\omega})$ consistent, but are not necessary since they are only used together in the form $(x^+_{ij}(\boldsymbol{\omega}) - x^-_{ij}(\boldsymbol{\omega}))$. Note that, if arc $(i,j) \in A_R$ is traversed both in the first and second stage, then $x_{ij}(\boldsymbol{\omega}) = 1$ and $x^+_{ij}(\boldsymbol{\omega}) = x^-_{ij}(\boldsymbol{\omega}) = 0$ (or, also, $x^+_{ij}(\boldsymbol{\omega}) = x^-_{ij}(\boldsymbol{\omega}) = 1$, if inequalities (4.18) are not enforced). Inequalities (4.19)-(4.20) impose that a second stage route services only customers assigned to the same first stage route. Such restriction could be, for example, due to sorting and bundling processes being done considering the first stage solution and which are not easily or quickly changed after recourse. Thus, a complete rescheduling is not allowed – reassignment of customers to a different vehicle route or the start of new routes is not allowed after recourse – and we only allow that customers may be rearranged within their corresponding first stage route and that a different location from the one chosen to service a customer in the first-stage may be selected to service that customer after recourse. Consequently, vehicle capacity does not need to be enforced in second-stage constraints. For the sake of simplicity, inequalities (4.19) are stated as informal implications in the model but can be formulated as linear constraints, as shown in the Appendix.

In formulating the two-stage model, we assume that the random variable $\boldsymbol{\omega}$ has finite support. We have one recourse structure $(\mathbf{x}^+,\mathbf{x}^-,\boldsymbol{\tau}^+,\boldsymbol{\lambda})$ and constraints (4.13)-(4.25) defined for each possible realization $\tilde{\boldsymbol{\omega}}$. Even if the particular distribution used to model travel times is not discrete, we can approximate the optimal value of the problem by considering $\boldsymbol{\omega}$ taking values over a discrete, finite subset of its support i.e., we consider a (potentially large) set of realizations of the distribution. Another difficulty with the formulation is that the recourse actions allowed by the model are less restrictive than those usually considered in the literature, specially regarding hard-time windows. For example, Errico et al. (2016) consider that a route needs at most one recourse action. More intricate actions make evaluating the recourse cost more complex and, even when simpler actions are considered, computing the value R with integer recourse (in the case $(\mathbf{x}^+, \mathbf{x}^-)$), requires solving many similar integer NP-Hard problems (Schultz et al., 1998).

Not surprisingly, it is usually the case that solving two-stage stochastic VRPs to optimality is only possible, from a computational point of view, for problems with few customers. Thus, we resort to a heuristic scenario-based method to solve the VRPRDL-STT considering fixed recourse actions applied to a first-stage solution to compensate for the possible infeasibilities arising after the observation of $\boldsymbol{\omega}$ (e.g. customers being serviced outside their time windows). In particular, one of the recourse actions consists of servicing a customer on a different location when the location given by the first-stage is not feasible given the revealed travel times.

In the following section, we describe each considered recourse action.

4.3.4. RECOURSE ACTIONS

Each non-serviced customer (location) potentially leads to an extra cost (Λ_c in the two-stage formulation) representing, for example, the cost of outsourcing the service. Moreover, routes exceeding the time horizon T incur an overtime penalty proportionally to the extra time. For all proposed actions, a vehicle initially follows its a priori route visiting customer locations within their time windows (in case of early arrival, waiting is allowed). Travel times are assumed to be revealed at once in the two-stage formulation (at the start of the second-stage). However, in evaluating the value R of a recourse action, values of travel times realizations $\widetilde{\omega}_{ij}$, $(i, j) \in A$, are assessed in an ordered manner, as dictated by the first-stage route. In particular, at a location $i \in N$, only travel time realizations of arcs adjacent to i, $\delta^-(\{i\})$ are used in evaluating the action to take. In this way, only values of arcs that could potentially be traversed by the vehicle from its current location are assumed to be known, and not values of arcs visited later in the route (e.g., non-anticipativity is preserved).

First, we describe three recourse actions not utilizing opportunities to service a customer at a different location than that of the first-stage route.

- *Do nothing* (R_0): No corrective action is taken i.e., the second-stage route remains the same as in the first-stage. The penalty Λ_c is incurred for all customers $c \in C$ serviced outside their time windows.

- *Skip next customer* (R_1): Given an a priori route, the realized travel times could lead to infeasibilities serving one or more customers within their time windows. If serving the next customer in the route is not possible (within the time window of the selected location), that customer is skipped and a penalty is incurred. Customers in the route continue to be skipped until a customer location able to be visited within its corresponding time windows is found (given the a priori route). If no customer location is found, the vehicle returns to the depot. Figure 4.2a illustrates an example for a route visiting four customers. After servicing customer

c_2, visiting c_3 within the time windows of the selected location is not possible and the customer is skipped. Similarly, it is not possible to visit c_4 (from c_2's location) within the corresponding time window of the selected location and customer c_4 is also skipped. Visiting customer c_5 at the selected location is feasible and, after servicing c_2, the vehicle visits customer c_5.

- *Skip customers* (R_2): Similar to the previous recourse action, customers are skipped whenever the first-stage route leads to a late visit, but the decision on which customers to skip is optimized. Given a first stage route $r = (v_0 = 0, v_1, v_2, \dots v_{|r|} = n+1)$, where $v_i \in N \cup \{0, n+1\},\ i = 0, \dots, |r|$ are the locations visited by r, and the realized travel times $\tilde{\boldsymbol{\omega}}$, define the following dynamic program:

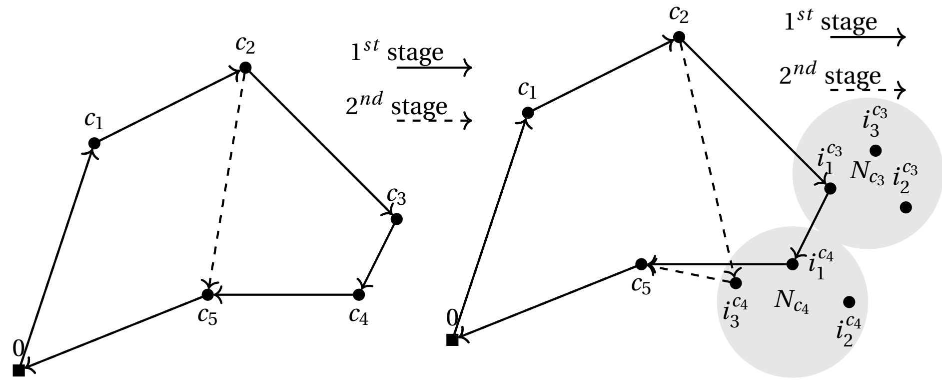
$$z(v_i, v_j, \tau_i) = \begin{cases} \beta \max\{\tau_i + \tilde{\omega}_{v_i,v_j} - T, 0\} & \text{if } v_j = n+1 \\ \min\{z(v_j, v_{j+1}, \max\{\tau_{\pi(v_i)} + \tilde{\omega}_{v_i,v_j}, \tilde{a}_{v_j}\}), & \text{if } \tau_{\pi(v_i)} + \tilde{\omega}_{v_i,v_j} \le \tilde{b}_{v_j} \\ \qquad z(v_i, v_{j+1}, \tau_{\pi(v_i)}) + \Lambda_{\pi(v_j)}\} & \\ z(v_i, v_{j+1}, \tau_{\pi(v_i)}) + \Lambda_{\pi(v_j)} & \text{otherwise} \end{cases} \tag{4.26}$$

where $z(v_i, v_j, \tau_i)$ is the minimum total penalty cost incurred for skipping customers in the (partial) route $(v_i, \dots, v_j)$, where the earliest time at v_i is τ_i. For route r, the minimum value is obtained by solving $z(0, v_1, a_0)$ and the corresponding set of skipped customers can be obtained by backtracking the optimal path in the recursion. Observe that the dynamic program in Equation (4.26) assumes the travel times of all arcs in the first-stage route to be revealed, thus violating non-anticipativity. We use this recourse as a means to evaluate the value of having all information available to the decision-maker and compare to the previous recourse R_1.

Since customers in the VRPRDL-STT have different possible locations for delivery (with non-overlapping time-windows), a natural recourse action consists in attempting delivery at another location in the customer's itinerary.

- *Reschedule next customer* (R_3): Similar to the *skip next customer* recourse, customers are visited following the locations given by the first stage route. If servicing the next location in the a priori route, within the location's time window, is not possible, instead of skipping this customer, a delivery to a different location is evaluated. If service is possible within the time window of another location in the customer's itinerary, then this location is selected and the customer is visited there. In case multiple locations can be selected, the one with the minimum detour time is chosen. If no location can be selected, the customer is skipped. Figure 4.2b depicts the same a priori route as in Figure 4.2a but applies this recourse in the second stage. Customer c_3 is skipped, as no other location in the set N_{c_3} can be visited within the corresponding time windows. Customer c_4 would be skipped if only the location selected in the first stage route could be visited. However, by selecting a different location in the set N_{c_4} (in the example, visiting $i_3^{c_4}$ instead of $i_1^{c_4}$), the vehicle can still deliver to customer c_4.

- *Reschedule customers* (R_4): Similarly to the previous recourse action, a different location other than that in the first-stage solution is selected for customers that would otherwise not be serviced by following the first-stage route. However, the set of customers to skip, and evaluate a new location for servicing, is obtained by solving the dynamic program defined in Recourse 2. All locations corresponding to skipped customers are removed from the route before trying to reinsert those customers again, but at a different location, in the route. Given a skipped customer, $c \in C$, a location $i \in N_c$ is inserted at the best position in the route, minimizing the required detour, and the best feasible insertion is executed. If no location $i \in N_c$ is feasible to be inserted, customer c is skipped in the route. Similarly to recourse R_2, recourse $R4$ also violates non-anticipativity, but we use it to assess the value of information and compare against recourse R_3.



(a) Skipping customers (c_3 and c_4).

(b) Changing the location of an otherwise skipped customer (c_4).

Figure 4.2: Recourse actions applied to a first-stage route visiting five customers. In 4.2a, customers are skipped if delivery occurs outside the location time windows (R_1) and, in 4.2b, a different location might be selected for an otherwise skipped customer (R_3).

In the following section, we elaborate on how we (heuristically) solve the two-stage model with recourse. It is worth mentioning that we are mainly interested in the first-stage solution from the model (the decision taken by the provider at planning stage). The second-stage recourse actions for a given realization $\tilde{\omega}$ does not lead to an *a-posteriori* solution, obtained during operational time, but to a *simulation* of it. Moreover, we assume that $\widetilde{\omega}_{ij}$ become known at the beginning of the second-stage, even though it might be only traversed at a later point in the operational time. Thus, a travel time scenario $\widetilde{\boldsymbol{\omega}}$ specifies values for traversing the arcs regardless of the period in the horizon $[0, T]$. Observe that this is in contrast with a multistage approach, wherein travel times are observed at different points over the horizon, and decisions (recourse actions) are taken to react to the outcomes after each observation.

4.4. SAMPLE AVERAGE APPROXIMATION

The Sample Average Approximation is a framework to solve stochastic discrete optimization problems by Monte-Carlo simulation proposed by Kleywegt et al. (2002). A computational study using the method for solving two-stage stochastic routing problems in the form $\min_{\mathbf{x}\in X} \mathbf{w}'\mathbf{x} + \mathbb{E}[R(\mathbf{x}, \boldsymbol{\omega})]$ was conducted by Verweij et al. (2003). The idea is to approximate the expected value function $\mathbb{E}[R(\mathbf{x}, \boldsymbol{\omega})]$ by the corresponding sample average function $z(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} R(\mathbf{x}, \widetilde{\boldsymbol{\omega}}^i)$, where $\Omega = \{\widetilde{\boldsymbol{\omega}}^1, \ldots, \widetilde{\boldsymbol{\omega}}^N\}$ is a set of realizations (scenarios) of the random vector $\boldsymbol{\omega}$, and solve a Sample Average Approximation (SAA replication) problem $\min_{\mathbf{x}\in X} v_\Omega(\mathbf{x})$, $v_\Omega(\mathbf{x}) = \mathbf{w}'\mathbf{x} + z(\mathbf{x})$, considering not the full support of $\boldsymbol{\omega}$ – which can grow exponentially with the dimension of $\mathbf{x}$ – but a smaller set of realizations. A large set of realizations, $\Omega' = \{\widetilde{\boldsymbol{\omega}}^{1'}, \ldots, \widetilde{\boldsymbol{\omega}}^{N'}\}$, $N' \gg N$, is then used to approximate the true expected value (solution gap) of the solution obtained by solving the SAA problem. The procedure is repeated until a certain criteria is met (e.g. the maximum number of replications, the optimality gap is small enough).

Algorithm 5 illustrates the main steps of our SAA implementation. Usually, implementations of the SAA framework proposed by Kleywegt et al. (2002) solve a fixed number, M, of SAA replications, generally using a fixed sample size $|\Omega| = N$. In contrast, in our implementation, the number of replications solved is not fixed and the sizes of the samples used to solve each replication is adjusted accordingly to the performance of the method for a current value, namely, the gap estimator, ϵ_m, after solving the m^{th} SAA replication. Kleywegt et al. (2002) utilize $\epsilon = v_{\Omega'}(\mathbf{x}) - \hat{v}_\Omega$ as an estimator of the (true) optimality gap $v_{\Omega'}(\mathbf{x}) - v^*$, where v^* is the optimal cost of the problem considering all possible realizations of $\boldsymbol{\omega}$. By solving each SAA replication to optimality, it can be shown that $v^* - \mathbb{E}[\hat{v}_\Omega]$ is monotonically decreasing in N i.e., $\hat{v}_\Omega$ is a statistical lower bound for v^* (Verweij et al., 2003). Since in our approach individual SAA problems are solved heuristically, $\hat{v}_\Omega$ is not necessarily a valid lower bound and, moreover, the lower bound estimator $\hat{v}_\Omega$ tends to overestimate the true lower bound. However, we still compute and use the estimator $\hat{v}_\Omega$ in order to evaluate the performance of the SAA problems given the current sample size, N, and to adjust (increase) the sample size throughout the method, but only after a certain level of convergence (variance of the values observed in the past L replications) is achieved.

Choosing the sizes N and N' is a trade-off between solution quality and computational efficiency in solving the SAA problems. Observe that solving each SAA problem becomes more time consuming as N increases, but the estimated lower bound, $\hat{v}_\Omega$, tends to be stronger and, consequently, the SAA gap tends to be smaller. In our implementation,

Algorithm 5: Overview of the (proposed) SAA framework implementation

Input : Initial sample size N; N', the large set of realizations $\Omega' = \{\tilde{\boldsymbol{\omega}}^{1'}, \ldots, \tilde{\boldsymbol{\omega}}^{N'}\}$;
Number of replications L to check for convergence

Output: Solution $\mathbf{x}^*$

$m \leftarrow 1$
Generate $\Omega^m = \{\tilde{\boldsymbol{\omega}}^1, \ldots, \tilde{\boldsymbol{\omega}}^N\}$
Solve the SAA problem over Ω^m, with objective v_{Ω^m} and solution $\mathbf{x}^m$
Evaluate $\mathbf{x}^m$ on Ω':
  $v_{\Omega'}(\mathbf{x}^m) \leftarrow \mathbf{w}'\mathbf{x}^m + \frac{1}{N'}\sum_{i=1}^{N'} R(\mathbf{x}^m, \tilde{\boldsymbol{\omega}}^{i'})$
Compute the average of solutions found in previous iterations
  $\hat{v}_\Omega \leftarrow \frac{1}{m}\sum_{i=1}^{m} v_{\Omega^i}$
Compute the SAA gap estimate:
  $f^* \leftarrow \min_{i=1,\ldots,m} v_{\Omega'}(\mathbf{x}^i)$;
  $\epsilon_m \leftarrow \frac{f^* - \hat{v}_\Omega}{f^*}$
Compute variance of $\hat{v}$ over the past L replications
  $\sigma^2_{\hat{v}_\Omega} = \frac{1}{(L-1)}\sum_{i=m-l}^{m}(\hat{v}_\Omega - v^i_\Omega)^2$
  $\sigma^2_{\epsilon_m} = \frac{1}{(L-1)}\sum_{i=m-l}^{m}(\hat{\epsilon} - \epsilon_i)^2$
Check convergence and update the sample size:
if $\sigma_{\hat{v}_\Omega} \leq 0.01$ **then**
  if $\epsilon_m \leq 0.05$ **then**
    Generate a new (independent) sample set Ω'
    return $\mathbf{x}^* = \min_{i=1,\ldots,m} v_{\Omega'}(\mathbf{x}^i)$
  else
    $N \leftarrow N + \Delta$
$m \leftarrow m + 1$
goto 2

we use a fixed $|\Omega'| = N'$ throughout the algorithm, and start solving the SAA problems over small sample sets $|\Omega| = N$. An integer parameter Δ is used to control the extent by which to increase N (how many additional scenarios to consider). When the solutions obtained by replications with the same sample size N converge, the gap estimator ϵ_m is evaluated and, in case it is not yet below the tolerance, N is increased by Δ and new replications are solved, until convergence. Otherwise, all solutions obtained by solving each SAA replication are evaluated over a new (independent) set Ω' and the best solution is returned.

In Section 4.5, we conduct a series of experiments to assess the implementation decisions taken in our proposed SAA.

4.4.1. SOLVING THE SAA PROBLEM

To solve the SAA problem $\min_{\mathbf{x}\in X}\mathbf{w}'\mathbf{x} + \frac{1}{N}\sum_{i=1}^{N} R(\mathbf{x},\tilde{\boldsymbol{\omega}}^i)$, we resort to an adaptation of the heuristics proposed by Reyes et al. (2017) which considers travel time uncertainty in the arcs. We did so by adding the sample average function to the objective and, as a result, we also modified the local search operators such that the marginal cost of an insertion considers both the changes in distance and recourse costs over N scenarios $\Omega = \{\tilde{\boldsymbol{\omega}}^1,\ldots,\tilde{\boldsymbol{\omega}}^N\}$. In particular, computing the change in the recourse costs before and after an operator is applied requires the evaluation of the modified solution (route) on the sample set Ω. Algorithm 6 gives an overview of how second-stage recourse costs are computed for a given first-stage solution and recourse policy, R, considering a sample of K travel time realizations.

Algorithm 6: Evaluating second stage costs for a first-stage solution $\mathbf{x}$.

Input : First stage routing solution $\mathbf{x}$; a sample $\Omega = \{\tilde{\boldsymbol{\omega}}^1,\ldots,\tilde{\boldsymbol{\omega}}^K\}$ of travel time realizations; a recourse action $R \in \{R_0, R_1, R_2, R_3, R_4\}$.

Output: The average recourse cost $\frac{1}{K}\sum_{i=1}^{K}, R(\mathbf{x},\tilde{\boldsymbol{\omega}}^i)$

1 $R(\mathbf{x},\Omega) \leftarrow 0$

2 **for** *each scenario* $\tilde{\boldsymbol{\omega}}^i \in \Omega$ **do**

3   **for** *each route, r, in* $\mathbf{x}$ **do**

4   Apply recourse R to r, considering travel times $\tilde{\boldsymbol{\omega}}^i$, to obtain a new route r'

5         $U \leftarrow$ set of customers visited by r but not visited following r'

6   **for** *each customer* $c \in U$ **do**

7             $R(\mathbf{x},\Omega) \leftarrow R(\mathbf{x},\Omega) + \Lambda_c$

8         $\lambda_{n+1} \leftarrow$ arrival time at the target depot for route r'

9         $R(\mathbf{x},\Omega) \leftarrow R(\mathbf{x},\Omega) + \beta\max\{0, \lambda_{n+1} - T\}$

10 **return** $\frac{R(\mathbf{x},\Omega)}{K}$

Given a realization of travel times, one of the recourse decisions described in Section 4.3.4 is applied to each a priori route in the first-stage solution $\mathbf{x}$ (Line 4), and a new route (second-stage route) is obtained in which customers are skipped and/or a different customer location is selected for service. The penalty Λ_c is is incurred for every customer $c \in U \subseteq C$ not visited in the second-stage route (Line 7) as well as the overtime cost (Line 9). Finally, Algorithm 6 returns the average recourse cost for solution $\mathbf{x}$ computed over all travel time scenarios in Ω (Line 10).

4.5. COMPUTATIONAL EVALUATION

In this section, we present a set of computational experiments conducted to evaluate the performance of the proposed framework in solving the VRPRD-STT and to assess the benefits of trunk delivery considering unknown travel times at the time of planning. Our algorithms are coded in Java and all experiments are executed on an Intel Xeon E5-2666 v3 CPU @3.5GHz machine, 15GB, running Ubuntu Server 18.04.

In our experiments, unless explicitly stated otherwise, we use the following input parameters. The number of samples used to solve each SAA replication is initially set to $N = 1$ and the increase parameter Δ is set to $5\lceil \frac{\epsilon_m}{0.05} \rceil$, i.e., the higher the relative difference between current and desired (estimate) gaps the larger the increase in the number of samples. The set Ω' has size $|\Omega'| = 10000$. Convergence of the estimators (gap and lower bound) are checked considering the variance over the last $L = 15$ replications and the algorithm terminates when convergence is attained and the gap estimator is below 0.05. We consider a congestion level of $\eta = 0.35$ (representative for the average congestion level of the top 100 congested cities worldwide (Tom Tom, 2016). The transportation cost matrix $\mathbf{w} = (w_{ij})_{(i,j)\in A_R}$ is defined by the travel distance between any two locations i and j. The overtime penalty is $\beta = 2$ and the maximum allowed overtime is set to 120 time units (two hours). We define the penalty cost of not servicing customer $c \in C$, in the second stage, as $\Lambda_c = 10F_c$, where F_c is the total (distance) cost of a single route from the depot to the home location of customer c and back to the depot e.g., the cost of a subcontracted, dedicated service. For a solution $\mathbf{x}$, we define $\mathbf{w}'\mathbf{x}$ and $\mathbb{E}[R, \mathbf{x}]$ as the corresponding first and second-stage costs, respectively.

4.5.1. Instance Description

To evaluate the impact of stochastic travel times in solving the VRPRDL, we perform computational experiments using problem instances introduced by Reyes et al. (2017). In particular, we consider their set of *general* instances, with 15, 20 and 30 customers, each with up to five roaming delivery locations, a time horizon of $T = 720$ (12 hours), a single depot (locations 0 and $n+1$ are the same) at the center of the region under consideration, and vehicles with capacity $Q = 750$. The geographic profile of each customer, $c \in C$, is as follows: the roaming delivery locations of c, N_c, are centered around the customer's home location and all are reachable, within the time horizon, from the depot; the itinerary of c starts at its home location, visits each roaming location and ends at home. Time windows (i.e., the time customer c spends at each location, being available for delivery) are generated by subtracting the total time spent by travelling from the time horizon and allocating the remaining time, in uniformly random lengths, to each location $i \in N_c$. Observe that if $|N_c| = 1$ then customer c is available during the whole time horizon exclusively at home.

Table 4.2 reports the characteristics of each instance considered in the computational experiments. Each row represents an instance and depicts the number of customers ($|C|$), the total number of locations ($\sum_c N_c$) and the average time, as a percentage of the time horizon T, that customers are available for delivery (Av.), where customer time windows are defined by deterministic (expected) travel times. We also distinguish between customers available only at home ($|N_c| = 1$, available over the full time horizon) and customers available at roaming delivery locations ($|N_c| > 1$). Thus, in the column Roaming, we report the average (Avg.), the minimum (Min) and maximum (Max) percentage of time that customers with roaming locations are available for delivery.

A disruption on an arc might alter the customer availability at the locations for trunk delivery (see Figure 4.1). In an extreme case, this means that delivery at some locations becomes infeasible. To give an insight on how changes in travel times might affect the availability (time windows at each location in the itinerary) we also report, in column Stochastic, the number of locations (Avg_m) that are unavailable once travel times are revealed and the time available for delivery (Av), as the average values over 10000 different travel time realizations.

	All customers (deterministic)			Roaming customers (deterministic)			Stochastic	
Instance	$\|C\|$	$\sum_c N_c$	Av.(%)	Avg(%)	Min (%)	Max (%)	Avg_m	Av(%)
I_0	15	63	63.1	47.7	4.9	81.4	5.4	60.9
I_1	15	58	69.1	52.2	31.0	90.4	3.9	66.2
I_2	15	53	71.3	53.5	7.9	98.6	7.4	68.1
I_3	15	51	64.2	39.1	8.3	82.4	6.1	61.6
I_4	15	53	69.9	55.7	14.2	96.9	3.6	69.8
I_5	20	67	69.9	49.1	16.3	99.7	5.2	68.1
I_6	20	69	67.7	40.7	7.5	97.2	9.3	65.8
I_7	20	81	60.0	41.3	4.4	64.2	10.3	57.3
I_8	20	77	53.9	36.6	1.0	85.0	10.3	51.6
I_9	20	64	69.5	52.1	14.2	98.6	6.9	67.3
I_{10}	30	104	64.2	50.2	3.1	98.1	6.7	61.8
I_{11}	30	114	56.4	41.8	0.6	89.4	14.9	53.6
I_{12}	30	119	57.0	44.9	3.8	83.3	13.6	54.0
I_{13}	30	108	63.6	47.1	12.4	91.7	9.4	61.7
I_{14}	30	125	56.7	46.8	4.0	96.3	13.7	54.1
I_{15}	30	120	61.5	48.7	2.8	93.8	6.3	58.8
I_{16}	30	131	53.0	46.3	7.5	93.5	13.3	50.2
I_{17}	30	107	61.1	43.4	0.6	64.4	9.9	58.7
I_{18}	30	99	65.0	49.1	7.6	85.3	10.3	62.5
I_{19}	30	76	78.1	50.0	4.9	91.4	7.9	76.8

Table 4.2: Description of a subset of the general VRPRDL instances proposed by Reyes et al. (2017).

4.5.2. GENERATING TRAVEL TIMES SCENARIOS

For this work, we follow an approach similar to Taş et al. (2013), Jabali et al. (2015) and Vareias et al. (2017). We model the (stochastic) time to traverse an arc $(i, j) \in A$ as a random variable given by $t_{ij} + \delta_{ij}$, where $\delta_{ij} \geq 0$ is the duration of the stochastic disruption on arc (i, j) and t_{ij} is a deterministic travel time, used in the first stage. Specifically, we consider that δ_{ij} follows a gamma distribution with a given shape parameter, k, and scale, θ_{ij}, depending on the deterministic travel time value of the arc. Gamma distributions are used to describe stochastic travel times in the literature, as it follows convolution and non-negativity properties. The parameters k and θ_{ij} allow for the generation of scenarios considering the degree by which travel times vary, adjusted by the *coefficient of variation* ($\hat{c_v}$).

Let $\eta \geq 0$ be a congestion level, representing the (expected) increase in travel time proportional to the travel time used in the first stage i.e., the travel time of arc (i,j) after a disruption incurs an expected increase of $\mathbb{E}[\delta_{ij}] = \eta t_{ij}$. Thus, if $\delta_{ij} \sim G(k, \theta_{ij})$, we have:

$$\mathbb{E}[\delta_{ij}] = k\theta_{ij} = \eta t_{ij} \tag{4.27}$$

$$Var(\delta_{ij}) = k\theta_{ij}^2 \tag{4.28}$$

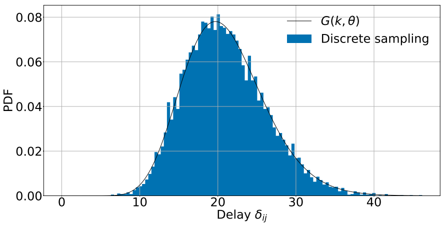
We derive parameters k and θ_{ij} for a given value $\hat{c_v}$ as follows:

$$\hat{c_v} = \frac{\sqrt{k\theta_{ij}^2}}{k\theta_{ij}} \implies k = \frac{1}{\hat{c_v}^2}, \; \theta_{ij} = \eta t_{ij} \hat{c_v}^2 \tag{4.29}$$

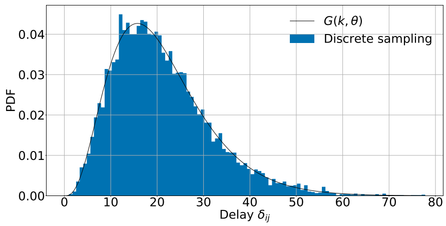
A scenario $\tilde{\boldsymbol{\omega}}$ is, then, an assignment of a random value to each arc $(i,j) \in A$, with $\tilde{\omega}_{ij} = t_{ij} + \delta_{ij}$, where δ_{ij} is drawn from the gamma distribution $G(k, \theta_{ij})$, given a value of $\hat{c_v}$. Figure 4.3 illustrates the probability density of the gamma distribution used to model the disruption on a particular arc $(i,j) \in A$ with $t_{ij} = 60$ and $\eta = 0.35$, considering different (squared) coefficient of variation values: $\hat{c}_v^2 = 0.0625$ (Figure 4.3a), $\hat{c}_v^2 = 0.25$ (Figure 4.3b) and $\hat{c}_v^2 = 1.00$ (Figure 4.3c). Within the SAA framework, the objective value (4.12) of the solution $\mathbf{x^i}$ obtained after solving the i^{th} SAA replication, is approximated by $v_{\Omega'}(\mathbf{x^i})$, taking a very large sample Ω'. Thus, Figure 4.3 presents an interval frequency for $|\Omega'| = 10000$ values sampled from the distribution.

In this specific example, the disruption δ_{ij} has the same expected value, $\mathbb{E}[\delta_{ij}] = 21$, regardless of the coefficient value used to obtain the parameters k and θ_{ij} of the gamma distribution. However, the samples derived from each distribution differ significantly. In particular, observe that the maximum disruption observed for $\hat{c}_v^2 = 1.00$ is approximately four (resp. two) times the maximum disruption observed for $\hat{c}_v^2 = 0.0625$ (resp. $\hat{c}_v^2 = 0.25$). The higher $\hat{c}_v^2$, the higher the number of samples with low probability, but with larger disruption values. Scenarios drawn from a distribution with $\hat{c}_v^2 = 0.0625$ represent short length disruptions occurring frequently (e.g., sudden increase in traffic as a consequence of events on adjacent streets), whereas values sampled from a distribution with $\hat{c}_v^2 = 1.00$ reflect severe disruptions with low probability of happening (e.g., accidents blocking one or more lanes, severe speed reduction due to road condition).

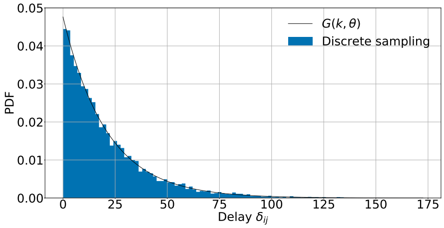
We consider that the travel times of arcs in the instances introduced by Reyes et al. (2017) already account for expected disruptions i.e., the deterministic travel time, $\bar{t}_{ij}$ of arc $(i,j) \in A$ is $\bar{t}_{ij} = t_{ij} + \mathbb{E}[\delta_{ij}]$. When solving an instance taking into account the stochastic disruptions, the travel time of an arc $(i,j) \in A$ in the first-stage is $t_{ij} = \frac{\bar{t}_{ij}}{(1+\eta)}$, and the stochastic disruption, δ_{ij}, in the second-stage is sampled from a gamma distribution, as shown previously, i.e., t_{ij} is the travel time to traverse arc $(i,j) \in A$ without any disruption, and $\mathbb{E}[\delta_{ij}] = \eta t_{ij}$. Figure 4.3d illustrates the travel and disruption times for an arc $(i,j) \in A$. In the remaining sections, unless specified otherwise, we use $\hat{c}_v^2 = 0.25$.



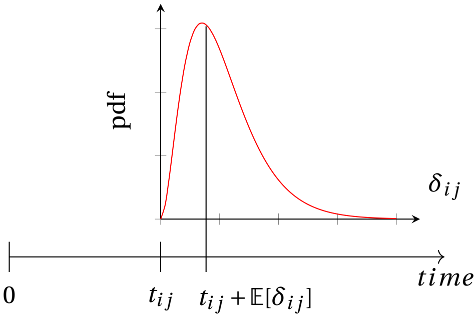
(a) $\hat{c}_v^2 = 0.0625$



(b) $\hat{c}_v^2 = 0.25$



(c) $\hat{c}_v^2 = 1.00$



(d) Changes in first-stage t_{ij} due to disruption.

Figure 4.3: Disruption, δ_{ij}, on arc (i, j), $t_{ij} = 60$ and $\eta = 0.35$, sampled with different $\hat{c}_v^2$ values.

4.5.3. The Value of a Stochastic Solution

To assess the value of stochastic information, we obtain a stochastic solution, $\mathbf{x}^s$, using our SAA framework, as well as a deterministic solution, $\mathbf{x}^d$, in which all random variables are replaced by their *expected values*. Next, we compare the routing costs (first stage), as well as the recourse costs (second stage) for both solutions. The expected recourse cost for a solution is obtained through Algorithm 6, using a large scenario sample size $|\Omega| = 10000$. Through this experiment, we show the extent to which first-stage decisions and costs of a stochastic solution change as a consequence of incorporating uncertainty in the travel times. To facilitate a fair comparison, we require that the number of routes in the stochastic solution does not exceed the number of routes used in the deterministic solution. As elaborated in Section 4.4, we use our adapted implementation of the local search heuristic proposed by Reyes et al. (2017) to compute both $\mathbf{x}^d$ and $\mathbf{x}^s$. To calculate $\mathbf{x}^s$, we use $\mathbf{x}^d$ as a starting solution to our heuristic.

Figure 4.4 depicts the comparison of $\mathbf{x}^d$ and $\mathbf{x}^s$ for the instances in Table 4.2 using recourse R_0. Recall that in R_0 no corrective actions are taken: the second-stage costs solely include penalties incurred for missed customers and overtime. In this experiment, the scenarios are generated using a coefficient of variance $\hat{c}_v^2 = 0.25$. The results for $\mathbf{x}^s$ are averaged over 3 runs of the SAA framework.

For each instance, Figure 4.4 shows the routing costs $\mathbf{w}'\mathbf{x}$, and the expected recourse costs, $\mathbb{E}[R_0, \mathbf{x}]$, for both the deterministic solution $\mathbf{x}^d$ and the stochastic solution $\mathbf{x}^s$. The total cost of a solution is given by $\mathbf{w}'\mathbf{x} + \mathbb{E}[R_0, \mathbf{x}]$. Figure 4.4 also shows the value of the stochastic solution computed as $vss = 100 \times \frac{(v_{\Omega'}(\mathbf{x}^d) - v_{\Omega'}(\mathbf{x}^s))}{v_{\Omega'}(\mathbf{x}^d)}$, where $v_{\Omega'}(\mathbf{x}^d)$ and $v_{\Omega'}(\mathbf{x}^s)$ are the total costs of the deterministic and stochastic solutions, respectively.

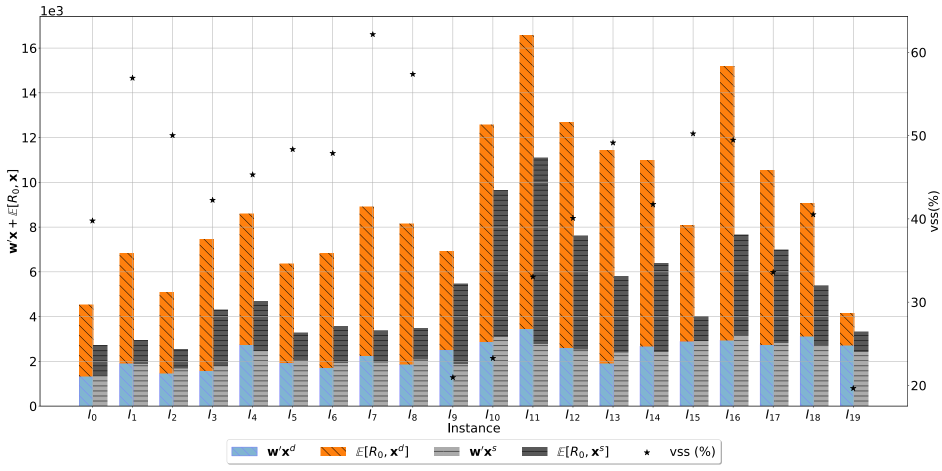


Figure 4.4: The value of incorporating stochastic information. Second-stage costs are derived from recourse R_0, travel times scenarios are derived from distributions having coefficient of variation $\hat{c}_v^2 = 0.25$.

As can be observed in Figure 4.4, the total costs of the deterministic solutions are significantly higher than the costs of the stochastic solutions. In fact, using our stochastic framework, we realize a cost reduction of nearly 42% compared to deterministic solutions. The difference in first-stage routing costs is small (on average 1%) but the routing plans differ significantly. The main changes stem from vehicle-customer assignment decisions, the order of service and the selected service locations.

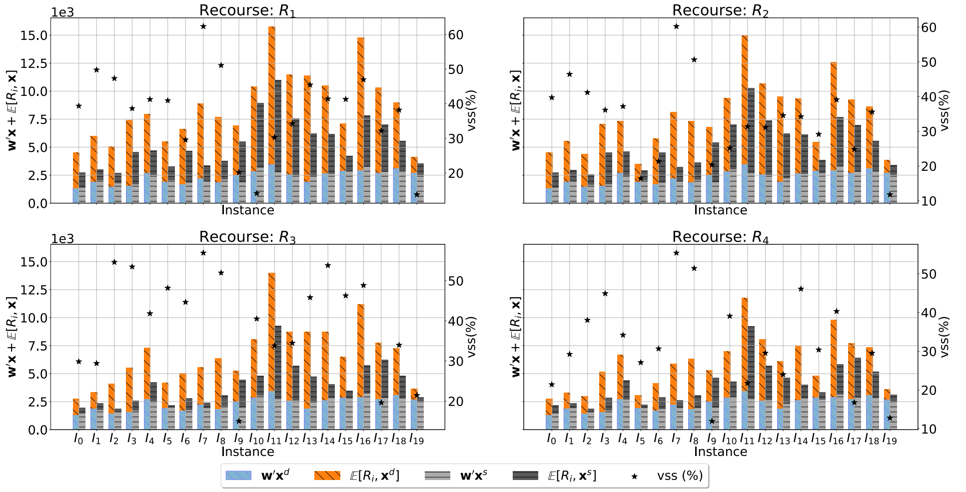


Figure 4.5: The value of incorporating stochastic information to solve the VRPRDL using recourse actions R_1, R_2, R_3 and R_4. Travel times scenarios derived from distributions having coefficient of variation $\hat{c}_v^2 = 0.25$.

In Figure 4.5 we repeat the same experiment for recourse actions R_1-R_4. Again, we observe that the stochastic solutions drastically improve over the deterministic solutions: the average cost reductions are 38%(R_1), 33%(R_2), 40% (R_3), and 32%(R_4). Note that in these experiments, the deterministic solution, $\mathbf{x}^d$, and its routing cost, $\mathbf{w}'\mathbf{x}^{\mathbf{d}}$, remain constant independent of the recourse action being used; only the expected recourse cost $\mathbb{E}[R_i, \mathbf{x}^{\mathbf{d}}]$ changes due to its dependency on R_i. When comparing the expected recourse costs $\mathbb{E}[R_i, \mathbf{x}^{\mathbf{d}}]$ for $i = 1, \ldots, 4$ and fixed $\mathbf{x}^d$, we observe that on average the recourse costs of R_4 are lowest, followed by R_3, R_2 and finally R_1.

4.5.4. RECOURSE POLICY COMPARISON

A mutual comparison of recourse policies R_1 and R_2, as well as R_3 and R_4 is provided in Figure 4.6. Per instance, the average results over three runs of the SAA algorithm are shown.

Recall from Section 4.3.4 that recourse policies R_1 and R_3 *greedily* skip or reschedule customers that cannot be reached within their time windows, whereas policies R_2 and R_4 use a more involved DP to compute the best subset of customers in a given route to skip or reschedule, violating non-anticipativity of travel time realizations. From Figure

4.6, we confirm that recourse policy R_2 outperforms its simpler counterpart (R_1): the costs of R_2 are on average 4% lower than R_1. The difference in costs obtained for policies R_3 and R_4 are negligible, with R_4 performing slightly better. These cost reductions come with an increase of computation time of 50 – 60%, because solving the dynamic problem (4.26) is computationally relatively expensive. The limited performance of recourse actions assuming full knowledge of travel time realizations is explained by the fact that the average number of customers per route, and thus the number of traversed arcs, in the first-stage solution is relatively low (4-5 customers). As a result, recourse policies wherein deciding on which customer(s) to skip is made assuming only knowledge of travel times for the next arc to traverse (R_1, R_3) performs similarly to recourse policies assuming full knowledge at the start of the second-stage (R_2,R_4). More specifically, comparing the results on the instance for which recourse R_2 improves over R_1 the most (20%) and the results on the instance with the least improvement (< 1%), R_2 outperforms R_1 when the solution contains slightly longer routes (5-6 customers). Longer routes consolidating more customers might decrease routing costs for first-stage decisions, but are likely to require more recourse actions due to travel time realizations, since they contain more arcs.

Figure 4.7 compares policies R_1 and R_3, investigating the benefit of being able to visit a customer at a different location than the one selected in the a priori plan. Policy R_3, in contrast to R_1, attempts to reschedule a customer at an alternative location in case of a missed delivery. Solution costs obtained with policy R_3 are on average 25% lower than the costs obtained with policy R_1: both the first stage costs and the second stage costs are lower (resp. 1% and 44%). Do note that evaluating recourse R_3 is generally more complex than evaluating R_1, and, as a result, evaluating the second-stage costs using R_3 takes about twice the execution time required to evaluate recourse R_1. In practice, this computation time increases proportional with the number of alternative locations in the itinerary of a customer.

Based on these results, we conclude that a recourse policy (R_3) exploiting the flexibility of visiting customers at different locations provides routing schedules with overall higher quality than a policy which simply skips customers who cannot be serviced in time (R_1), but also incurs in relatively higher computational costs. If fast computation times are required, then recourse action R_1 is recommended, which still achieves a 37% improvement over a deterministic model which only takes expected values into account. Moreover, recall that only recourse actions R_0, R_1 and R_3 yield realistic policies that could be applied as travel times are revealed throughout the horizon, as R_2 and R_4 violate the non-anticipativity of travel times realization. It is however interesting to observe how close the performance of recourse policy R_3 is, which only uses partial information, in comparison to R_4. In practise, R_2 and R_4 could still be applied in the case when travel times are revealed, or at least a good estimation of them is made available, in periods (e.g., morning, afternoon, evening). Thus, the dynamic problem in Equation (4.26) could be solved using values of all arcs traversed in a given period by the a-priori route.

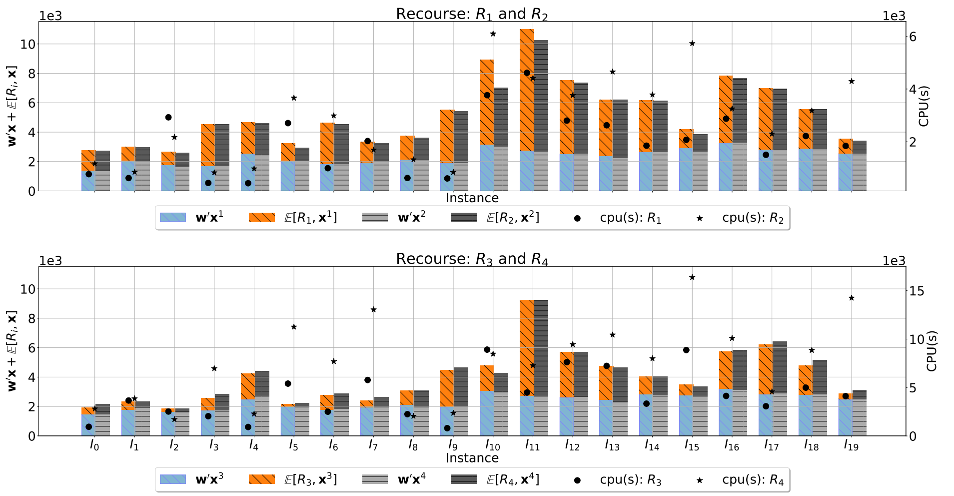


Figure 4.6: Comparison of solutions obtained by recourse actions R_1 and R_3 compared to solving the dynamic programming (R_2 and R_4) model (4.26).

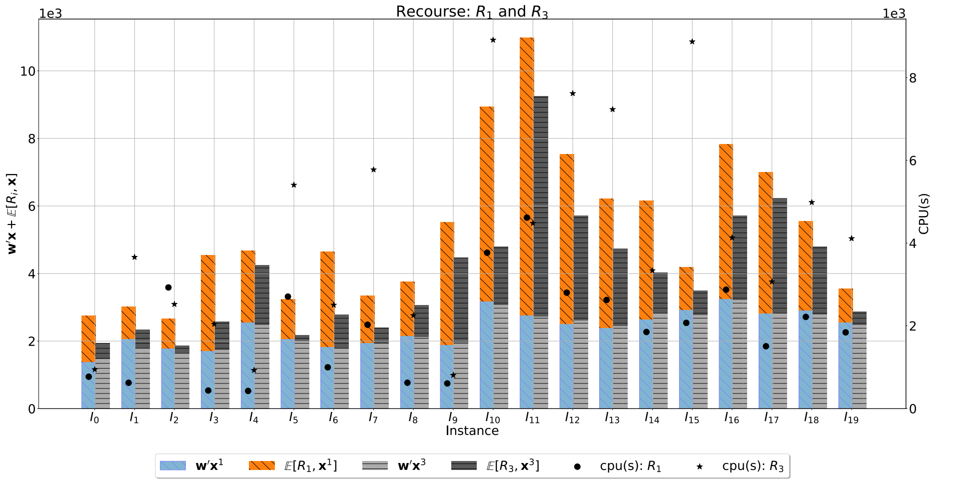


Figure 4.7: Savings by changing the service location of an otherwise skipped customer.

	$\hat{c}_v^2 = 0.0625$			$\hat{c}_v^2 = 1.00$		
	$v_{\Omega'}(\mathbf{x}^*)$	T(s)	$vss(\%)$	$v_{\Omega'}(\mathbf{x}^*)$	T(s)	$vss(\%)$
Average	3,107	1,920	38.4	5,546	7,881	32.3

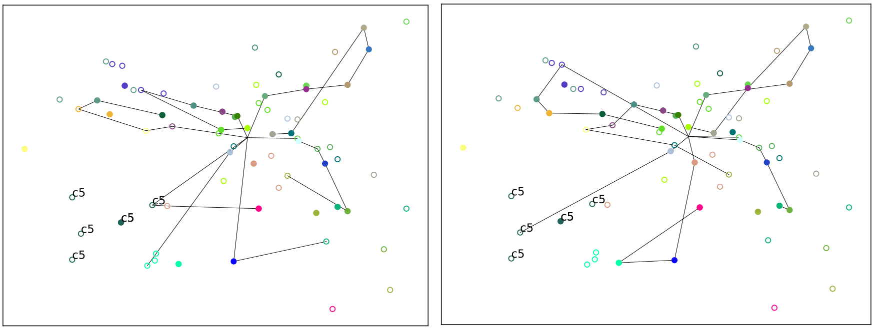
Table 4.3: Impact of travel time scenarios generated with different coefficient of variance. R_3 is applied as second-stage recourse action.

4.5.5. The Impact of Travel Time Variance

To assess the impact of travel time variance, we conduct an experiment in which the stochastic travel times are drawn from different gamma distributions: one with a *low* variance ($\hat{c}_v^2 = 0.0625$) and another one with a *high* variance ($\hat{c}_v^2 = 1.00$). The results averaged over *all* instances are reported in Table 4.3. Again the SAA algorithm has been invoked 3 times per instance.

As can be observed from Table 4.3, the solution costs $v_{\Omega'}(\mathbf{x}^*)$ increase significantly when the variance increases: $v_{\Omega'}(\mathbf{x}^*)$, for the high variance instances, is almost twice the value of the low variance instances. On average, the second-stage recourse costs account for 31% (low variance) resp. 63% (high variance) of these total costs. Next to an increase in costs, we witness a decrease in the Value of the Stochastic Solutions (VSS), from 38.4% for the low variance instances to 33.7% for the high variance instances. Finally, we observe a significant increase in computation times when the variance increases. This increase is attributed to the fact that in case of a high variance, more SAA replications are needed with a larger sample size (N) before the termination criteria of our SAA algorithm are met.

Figure 4.8 illustrates an example of how (first-stage) routing decisions can be affected when stochastic travel times are taken into account and the impact of those decisions on second-stage costs. In Figure 4.8a we show the deterministic solution routing for instance I_{18}. Figure 4.8b shows a stochastic solution obtained with the SAA using recourse policy R_3 under travel time scenarios with $\hat{c}_v^2 = 0.25$, for the same instance. The locations visited by a given customer have all the same color but are distinguished by solid (home location) and empty circles (the roaming delivery locations). In particular, we highlight customer c_5, visiting four roaming locations (most likely to be impacted by travel time disturbances). In the deterministic solution, the delivery to customer c_5 is made at the second roaming location visited by c_5, which is closer to the depot (at the center of the figures). For the stochastic solution, the sampling scheme in the SAA is able to drive the routing solution to deliver to c_5 at the third roaming location visited by the customer, farther from the depot but closer to the next roaming location and to home. The stochastic solution is thus able to cope better with travel time disturbances affecting customer c_5. Although the routing costs are similar (less than 4% difference), when both routes are evaluated over travel time scenarios with $\hat{c}_v^2 = 0.25$ and with recourse R_3, the second-stage cost for the stochastic solution is more than 50% cheaper than for the deterministic solution.



(a) Deterministic solution

(b) Stochastic solution. R_3 and $\hat{c}_v^2 = 0.25$

Figure 4.8: Routing solutions for the deterministic (a) and stochastic (b) solutions obtained for instance I_{18}. For the sake of clarity, the last leg of the routes (to the depot) are not shown.

4.5.6. MINIMUM VISITING TIME AT ROAMING LOCATIONS

In the previous experiments, we assumed that for a customer $c \in C$, when arriving at a roaming location $i \in N_c$ after the latest time, b_i, (that is, for a realized travel time scenario, $\tilde{a}_i > b_i$) the customer does not spend time at location i and drives directly to the next location in his/her itinerary. Thus, we conducted a few experiments where we impose that each customer spends at least a given amount of time, τ, at each roaming location in his/her itinerary, regardless of the time of arrival. Similarly to the previous experiments, an arrival after b_i implies that location i will not be available for delivery. We assume that τ is too short to make the service feasible. If the length of the time windows, $[\tilde{a}_i, \tilde{b}_i]$, at location i for a given realized scenario is smaller than τ, then we impose $\tilde{b}_i = \tilde{a}_i + \tau$.

Table 4.4 reports the results of the experiments, using $\tau = 10$ and travel time scenarios with $\hat{c}_v^2 = 0.25$. Column Instance shows the aggregate instances by the number of customers ($I_{1-4} : |C| = 15$, $I_{5-9} : |C| = 20$ and $I_{10-19} : |C| = 30$). In columns $v_{\Omega'}(\mathbf{x}^s)$ and $vss(\%)$, we report the average solution costs and the value of stochastic solution, respectively, over instances of the corresponding aggregation (recall that to compute the vss, the deterministic solution is evaluated over the same travel time scenarios and recourse policy as for the stochastic solution when evaluating second-stage costs). We observe similar results as those presented in Section 4.5.3, where the stochastic solutions clearly perform better when evaluated in different travel time scenarios. However, the total costs are larger compared to those observed previously, on average 30% more expensive. A delivery can be scheduled for a location $i \in N_c$, at which the realized time window $[\tilde{a}_i, \tilde{b}_i]$ is shorter than τ units of time. In case the recourse policy is not able to re-schedule the delivery to another location, the customer is missed, increasing second-stage costs. Moreover, once a customer is delayed, unless he/she spends longer at subsequent locations, it will be harder to deliver at a roaming location. Those scenarios were not considered for the experiments in Section 4.5.3.

	Recourse R_3		Recourse R_1	
Instance	$v_{\Omega'}(\mathbf{x}^s)$	$vss(\%)$	$v_{\Omega'}(\mathbf{x}^s)$	$vss(\%)$
I_{1-4}	3,554.4	29.2	4,376.1	36.2
I_{5-9}	4,526.4	32.3	5,174.6	36.8
I_{10-19}	6,715.1	26.9	8,119.7	28.0

Table 4.4: Value of stochastic solutions considering customers always spend time ($\tau = 10$ units of time) at the roaming locations along their itineraries, regardless of time of arrival.

4.5.7. Evaluating the SAA Framework

4

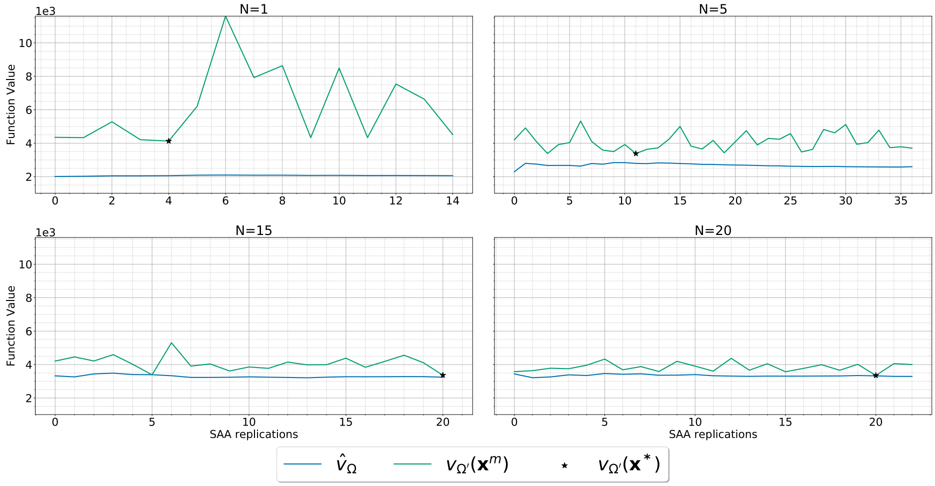
A common problem in the implementation of SAA algorithms is selecting the parameters N (number of scenarios in set Ω) and M (number of SAA replications). Both parameters are instance-dependent: low values of M and N lead to poor solutions, whereas high values of M and N significantly increase computation times. In this work, we aim to avoid this problem through an incremental update scheme for N and a termination criteria based on a gap estimate ϵ_m (Algorithm 5), instead of a fixed number of iterations M. In this section, we conclude with an empirical evaluation of these design choices.

To establish the impact of our incremental update scheme, we solve the benchmark instances (Table 4.2) for fixed values of N by setting $\Delta = 0$ in Algorithm 5. We terminate SAA as soon as $\sigma_{\hat{v}_\Omega}$ is less than 0.01 (line 15 in Algorithm 5). For these experiments, we use recourse policy R_0 and $\hat{c}_v^2 = 0.25$.

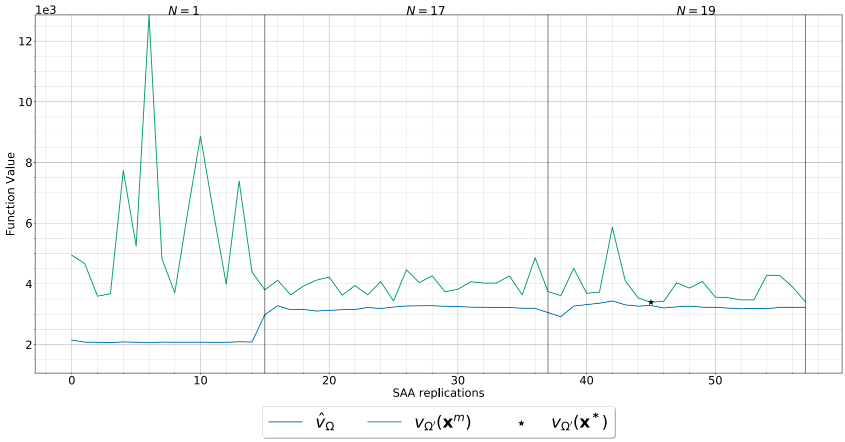
As an example, we solve instance I_5 for fixed $N = 1, 5, 15, 20$ with our incremental update scheme. The results are depicted in Figure 4.9a. For each SAA replication m (x-axis) the graphs show the value $\hat{v}_\Omega$ and the objective value $v_{\Omega'}(\mathbf{x^m})$. Clearly, when N is too small (e.g. $N = 1$), the objective value $v_{\Omega'}(\mathbf{x^m})$ fluctuates considerably. For larger values of N the objective value becomes more stable, indicating that the solution $\mathbf{x^m}$ obtained by solving the m^{th} SAA replication with $|\Omega| = N$ closely approximates the value $v_{\Omega'}(\mathbf{x^m})$. Figure 4.9b shows that our incremental scheme increases the value of N twice, and terminates after $m = 60$ iterations. Here, the best solution value is found for $N = 19$.

Table 4.5 summarizes the results, averaged over *all* instances with the same number of customers $|C|$. Similar to the experiments in the previous sections, the SAA algorithm has been invoked 3 times per instance. For a given $|C|$, Table 4.5 reports the average results obtained with fixed $N = 1, 5, 15, 30, 60, 90$, and with the proposed incremental update scheme ($\sim$). For a given $|C|$ and N, we state the best objective value found $v_{\Omega'}(\mathbf{x}^*)$ by the time the algorithm terminates, the time at which this best solution was discovered (TTB), the total execution time (T(s)), and the number of times ($\mathcal{N}$) SAA terminated with $\epsilon_m \leq 0.05$. Here, we remind the reader that when $\epsilon_m > 0.05$ our incremental update scheme would increase sample size N (line 19 in Algorithm 5).

Observe from Table 4.5 that when $|C|$ equals 15 and 20, the best results are obtained for $N = 30$, whereas for $|C| = 30$, the best result is found for $N = 90$. Manually picking the ideal value of N is hard without enumerating different values of N. Fortunately our



(a) SAA replications solved with a fixed sample size $N = 1, 5, 15, 20$.



(b) SAA replications with sample sizes adjusted throughout the method.

Figure 4.9: Comparison of the evolution of the SAA framework considering fixed sample sizes N (a) and the adjusted sizes (b)

incremental update scheme (~) finds solutions of comparable quality, without the need to manually select N, and relatively fast when compared against the computation times for larger N.

	$\|C\| = 15$				$\|C\| = 20$			
N	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)
1	4,108	0	15	32	5,182	0	23	41
5	3,528	0	24	70	4,039	1	54	108
15	3,453	0	58	136	3,915	1	122	218
30	3,414	3	70	215	3,829	1	108	383
60	3,427	3	145	463	3,895	3	783	994
90	3,434	5^a	175	816	3,933	5^a	397	1,301
$\sim$	3,444	5^a	108	297	3,834	5^a	445	994

	$\|C\| = 30$			
N	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)
1	9,212	0	62	95
5	7,304	1	108	260
15	6,878	0	360	582
30	6,796	5	680	1,174
60	6,785	8	1,081	2,177
90	6,774	10^a	2,396	4,737
$\sim$	6,796	10^a	2,143	3,373

Table 4.5: Running the SAA with a fixed N. $v_{\Omega'}(\mathbf{x}^*)$: best solution found; $\mathcal{N}$: the number of times SAA terminated with $\epsilon_m \leq 0.05$; TTB: time when best solution found; T(s): total execution time. $\hat{c}_v^2 = 0.25$ and R_0 as second-stage recourse action. [a]All SAA invocations terminated with $\epsilon_m \leq 0.05$.

4.6. CONCLUSIONS

We addressed a stochastic variant to a last-mile delivery problem, considering trunk delivery. In this case, a customer's car is used to facilitate the delivery process of direct-to-consumer orders. This problem is modeled as a Vehicle Routing Problem with Roaming Delivery Locations. Customers are assumed to be available for delivery at different locations as he/she moves along an itinerary, defining non-overlapping windows for servicing. By considering uncertain travel times when designing the routes for service, not only delivery vehicles are affected by possible differences between expected and realized times, but also customers, since travel times also affect the availability of customers at each location along their itinerary. Moreover, the presence of different locations for delivery to a customer might provide service providers with new approaches to handle uncertain events (e.g., road disruptions) while maintaining a desirable service level to customers.

We tackled the problem as a two-stage stochastic problem and implemented a hierarchical, scenario-based sample approximation method, in combination with an adaptation of local search heuristics, to take travel time uncertainty into account. Experiments conducted on a set of VRPDL instances showed that planning the delivery routes while explicitly taking stochasticity into account leads to significant savings compared to a deterministic approach which solely considers expected travel time values. Routing plans obtained by using expected values are too conservative when travel time realizations are shorter than expectations, or induce many missed deliveries in the advent of significant traffic disruptions. The SAA approach used in this paper resolves this issue by optimizing routes over a large number of potential travel time realizations.

Future research could involve the role of new technology (e.g., machine learning) to monitor and predict daily itineraries of receivers and drivers. This could allow, for example, delivery routes being adjusted dynamically, using real-time information provided by customers given the current status of the network.

5

Vehicle Routing with Dynamic Roaming Delivery Locations

For all its uncertainty, we cannot flee the future.
One meets (her)his destiny often in the road (s)he takes to avoid it.

Jean de La Fontaine

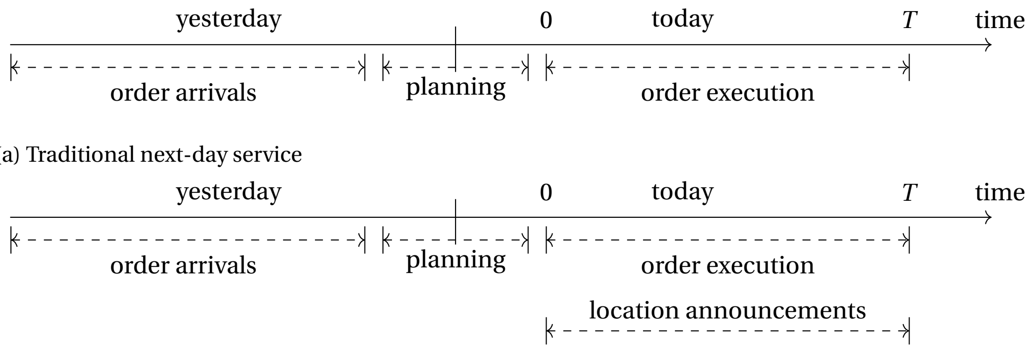
In this chapter, we consider a new dynamic variant of the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL). In this new variant, customers' itineraries are not known to the service provider beforehand but dynamically announced by customers throughout the time horizon. The only information required by the service provider from its customers are the home location and the corresponding customer's availability at home. A Multiple Plan Approach (MPA) is used in which multiple routing plans are maintained to provide alternative ways for reacting to dynamic information. Decisions to dispatch a vehicle servicing customers at home or at a dynamically announced location are taken by means of a consensus function, which selects an appropriate route in the pool. The results show that the solutions obtained with the MPA effectively make use of dynamically announced locations by customers. In our experiments, integrating dynamic locations into the routing plans led to improvements for almost all problem instances, with certain instances displaying improvements of more than 30%.

5.1. INTRODUCTION

Traditional brick and mortar stores were once the only option customers had to buy their products. The ubiquity of internet access and the increased penetration of mobility technology (e.g., mobile phones, smart devices, internet of things) not only dramatically increased on-line sales and e-commerce, but also shifted the power towards the consumer. Higher service levels are now offered as a means to provide customers more convenience while e-shopping. Fulfilling customer demands through multiple channels, the omnichannel logistics, have gained importance as a strategy to cope with higher customer expectations. However, offering and satisfying such high level requirements to enhance customer's experience do not come without challenges and costs.

In a conventional e-commerce setting, customers place orders on-line and have their products delivered at home, possibly within a preferred time windows. However, the small size of individual deliveries and the increased number of freight movements render home delivery an expensive option, both from a service provider and city logistics perspective (Savelsbergh and Van Woensel, 2016). Moreover, while speedy delivery options can increase customer satisfaction, many customers might not need or want such options, due to extra cost for the service (e.g., home delivery with a window length of two hours can be as costly as three times the delivery cost of an unattended delivery (Punakivi et al., 2001)). A study conducted by Retail System Research (Baird and Rosenblum, 2015) investigates the challenges faced by retailers to fulfill home-deliveries. In addition, the study evaluates the benefits perceived by the customer when opting for a delivery at home. Most of the retailers recognize the importance of increasing the speed of delivery to customers (52%) and offer differentiated services (47%), whereas many customers value narrower delivery windows (42%) and e-tailers that offer a delivery service which is faster than driving to a local store (33%). An option to service the customer at a more convenient location than home is to use the trunk of the customer's car. This option has gained traction recently after Amazon, in partnership with some car's manufacturers, launched its service, Amazon In-Car delivery (Hawkins, 2018), at some selected areas in the US. Using the trunk of customers' cars is also been considered by DHL and the Dutch postal service, PostNL, as one of the alternatives to provide more efficient last-mile services (Cohen, 2019). By having access to the trunk of the customers' cars, the service provider can fulfill delivery or pickup (e.g., returning of merchandising) requests without the physical presence of customers.

In this work, we consider a dynamic last-mile service system with roaming customer locations where the service provider is able to access the trunk of a customer's car. Most works assessing roaming last-mile systems consider that customers inform their full planned routes to the service provider. However, advances in communication technologies allow for options in which customers are not required to inform their full planned journeys in advance. Thus, customers can have a more convenient experience while e-shopping and service providers can deal with less uncertainty regarding customers planned routes during order execution. In particular, the provider offers a next-day service, in which orders are received and processed throughout the day before the service



(a) Traditional next-day service

(b) Next-day service with roaming locations

Figure 5.1: Time progression of logistics processes involved in next-day last-mile delivery services.

operation starts. When posing a request, a customer indicates only his/her home location and availability (as time windows) at home, for the next day. Figure 5.1b illustrates the service setting considered. During the planning phase, all orders received are taken into account to determine viable routes to service customers at their home locations. Note that new customers arriving during the execution of the routes are only considered for next-day service. Over the course of the day, customers in the planned routes can announce, dynamically, other location(s) where they will be available for service (e.g., the car will be parked at a given location for a specific duration). Vehicles are loaded and dispatched from a depot to visit customers at one of their locations and, at any given time during the day, the service provider might decide to dispatch vehicles for servicing customers either at home or at a dynamically announced location.

5.1.1. Literature Review

Reyes et al. (2017) first introduced the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL), developed construction and improvement heuristics and assessed the benefits of trunk delivery, thereby showing that a reduction of up to 50% on total travelled distance can be achieved for certain environments. An alternative solution approach for the VRPRDL was proposed by Ozbaygin et al. (2017). Their exact approach based on a branch-and-price framework is able to solve instances with up to 120 customers. In both works, all relevant information (e.g., travel times, customer itineraries, demands) is assumed to be *static and deterministic* – all parameters are known in advance and with certainty at planning, and do not change during operation. Advances in information and communication technologies enabled the development of many innovative transportation services and fostered the interest on *dynamic* routing problems, in which part or all of the information is revealed dynamically, concomitantly with the execution of planned routes. The reader is referred to Pillac et al. (2013b) and Bektas et al. (2014b) for extensive overviews on dynamic routing problems.

Recently, Ozbaygin and Savelsbergh (2019) considered a dynamic and deterministic variant of the VRPRDL. Customers planned routes for the entire day (customer itinerary) are known during planning stage but might change during the execution of the delivery schedules. In particular, the authors assume that a customer always visits the locations in his/her itinerary but arrival and departure times at these locations might deviate from the values used during planning stage. The authors propose an iterative framework which solves a (static) VRPRDL, with updated data from customers, using a branch-and-price method whenever a certain number of deviations from the input data have occurred. By using information collected during the solving of previous optimizations, the authors are able to efficiently solve the problem in short computational times.

To the best of our knowledge, Ozbaygin and Savelsbergh (2019) is the only study addressing a dynamic variant of the VRPRDL. The dynamic VRPRDL that we consider differs from the one addressed in that work regarding:

The source of dynamism: Ozbaygin and Savelsbergh (2019) assume that all customer locations are known beforehand and that customers might depart earlier or arrive later than the times specified on the a-priori itinerary (i.e., time windows at locations may change). We assume that customers only provide the home location and associated time windows during the planning stage, and that other locations and corresponding time windows along their itineraries are announced throughout the day.

Service setting: in Ozbaygin and Savelsbergh (2019), the focus is on collection settings, wherein the company collect packages from the trunk of a customer's car (e.g., return of goods). As a single commodity setting is assumed, a vehicle can service a customer which was previously assigned to a different vehicle. Consequently, dynamically arriving updates might be integrated in the current routing plan even when changing in assignment is required. In our work, we consider delivery applications (e.g., last-mile) and assume that each customer has a unique product to receive (multi-commodity). In particular, if the product of a customer is loaded into a vehicle, either that vehicle delivers to the customer or it has to come back to the depot before the item can be loaded into another vehicle.

Waiting strategy: in Ozbaygin and Savelsbergh (2019), a vehicle can wait at a location (before servicing the customer), and can be re-routed before service starts, when new information arrives and re-optimization is performed. Vehicles are dispatched at the start of the time horizon, fully loaded with the commodity. In our work, dispatching decisions are explicitly taken into account. In particular, delaying dispatching decisions may allow for the gathering of more information (e.g, a customer informs a location in his/her itinerary rendering a better overall routing plan). As noted, for delivery requests, consolidation opportunities are only possible when vehicles are in the depot and solely minimizing travel times may induce higher waiting times at customer locations. Thus, we aim for solutions with limited waiting times at customer locations and allow vehicles to perform multiple trips (return to the depot multiple times) within the operational horizon.

5.1.2. CONTRIBUTIONS

The main contributions of this work are: (i) we introduce a dynamic variant of the VRPRDL in which customers are not required to inform their full planned routes at the day of service. We show that while having customer itineraries revealed dynamically provides more flexibility to customers, compared to solutions considering only home visits, it can, however, potentially improve the operations of service providers. We indicate insights on the compromises that a delivery company can consider when offering such service compared to a traditional, only-home delivery option. (ii) We consider a dynamic strategy to decide on the actions to take during the operational day, in particular, to support vehicle dispatching decisions. (iii) We evaluate the potential benefits of including roaming locations as a mean to integrate the fulfillment of both delivery and return (pickup) flows.

The remainder of this chapter is structured as follows. In Section 5.2, we describe the problem, in particular, the dynamic settings considered. Section 5.3 describes the proposed Multiple Plan Approach (MPA) and the rolling horizon framework. We elaborate on the dynamic events and how the solution pool is maintained and updated after the triggering of each event. In Section 5.4, we present experimental results assessing the benefits of the solutions obtained with the methodology against a traditional, only home-solution. Finally, we conclude with some insights for further work.

5

5.2. PROBLEM FORMULATION

The VRPDRL considered in this work is characterized by a central depot, denoted by 0, from which a set of homogeneous vehicles $\mathcal{V} = \{1..., V\}$, each with capacity Q, is dispatched to provide last-mile pickup and delivery services within a given geographic area R. Let $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ be the set of customers whose orders the service provider received and will be serviced during the current operating day defined by the time-horizon $[0, T]$ (see Figure 5.1b). Each customer $c \in \mathcal{C}$ has a non-negative demand, d_c, that the service provider has to fulfill, either a delivery or a pickup (return of parcels) operation. When posing a request, besides the demand, customer c provides his/her home location, $\ell_h^c \in R$, and associated time windows. Customers might have multiple time windows at home (e..g, home availability at different times), $[a_{h_1}^c, b_{h_1}^c], [a_{h_2}^c, b_{h_2}^c], ..., [a_{h_M}^c, b_{h_M}^c] \subseteq [0, T]$, where these time windows are ordered and non-overlapping. During the planning phase, the service provider uses the informed home locations and time windows to determine a routing plan servicing the set of customers $\mathcal{C}$. At the start of the operating day the provider might dispatch vehicles for servicing customers at home, as decided in the routing plan. Alternatively, the provider can defer visiting some customers at home until some other locations are announced. Note that the provider is not certain about whether a given customer will announce a location. However, the provider does know the latest time to dispatch a vehicle for servicing the customer at home during his/her last time windows there, and can potentially wait until that time for an announcement.

During operation, customers who did not yet have their requests satisfied can announce to the service provider their current location and availability (i.e., where and for how long the customer's car is parked). Thus, the company might postpone dispatching decisions and wait for customers to provide alternative service locations that could potentially yield more favorable routes. For customer $c \in \mathscr{C}$, let $i \in \{1,2,3,...\}$ and ℓ_i^c be the i^{th} location visited by customer c throughout the operational horizon $[0,T]$, with availability $[a_i^c, b_i^c]$. Given the updated information provided by not-yet serviced customers, the provider could decide to service those customers at their present location, instead of at home, integrating the new location into the routing plan. Once the provider makes a dispatch decision and a vehicle leaves the depot following a given route, all customers assigned to the vehicle are visited at the locations prescribed by the route and the vehicle returns to the depot once all assigned customers have been serviced. Additional visits to customers requiring pickup of goods might still be added to the route after the vehicle is dispatched. A vehicle is allowed to wait at a customer location if the visit occurs before the earliest time for that location, but late visits are not allowed (hard time windows).

The dynamic operating scenarios are based on the following assumptions:

- Uncertainty comes from a single source. In particular, uncertainty comes from the possibility of customers announcing – dynamically, as time goes by – locations in their itineraries where service can also take place, besides home location. Decisions are taken without any knowledge of future customer locations i.e., without assuming stochastic information on customers' itineraries.

- Diversion is not allowed: once a driver is en route to his/her next destination, he/she must necessarily service that location.

- To fulfill delivery requests, a vehicle has to be loaded from the depot whereas pickup requests can be serviced by en-route vehicles.

The objective is to dynamically design a set of routes servicing customers either at home or at a dynamically announced location, minimizing system wide transportation costs, accounting for total travel distance and waiting times. We consider settings in which travel times are relatively short compared to the planning horizon, such that vehicles are able to perform multiple routes departing from the depot. Moreover, we assume all customers are serviced, either at home or at a dynamically announced location, and that a sufficient vehicle fleet is available for that.

At any point in time $t \in [0,T]$, the following information is available to the service provider for assisting decision making:

- The set $\mathscr{H} = \{\ell_h^{c_1}, ..., \ell_h^{c_n}\}$ of customers home locations as well as the associated time windows at which customers are at home.

- The set $\mathscr{L}^t = \{\ell^{c_1}, \ell^{c_2}, ..., \ell^{c_n}\} \cup \mathscr{H}$, containing all known valid locations at time t, where a location is valid at time t if it can still be visited, within its corresponding time windows, at time t.

- The set $\mathscr{C}^t$, containing the set of customers not yet serviced by time t. Let $\bar{\mathscr{C}}^t = \mathscr{C} \setminus \mathscr{C}^t$ be the set of customers serviced by time t.
- The set $\mathscr{V}^t$, containing the routes assigned to each dispatched vehicle that did not return to the depot by time t. In particular, for a route $r \in \mathscr{V}^t$, let $\phi(r)$ be the time that the vehicle assigned to route r returns to the depot.
- The travel time between two locations $i, j \in \mathscr{L}^t \cup \{0\}$ is deterministic and represented by τ_{ij}, which is proportional to the distance, d_{ij}, and cost, c_{ij}, incurred when a vehicle goes from location i to j.

Throughout the text, we employ the following terminology. A *routing plan*, or plan for short, consists of a set of *routes* covering all customers who were not yet serviced, either at home or at a dynamically announced location in a customer itinerary. Within a routing plan, a customer can be visited by at most two routes: one route visiting the customer at home and a different route visiting the customer at a dynamically announced location. Considering all information known to a customer can increase the ability of a routing plan to accommodate future information. A route starts at the depot, visits a given subset of the customers at most once, and returns to the depot. At some point in the operational day, a route in the routing plan can be assigned to a *vehicle*, and that vehicle is immediately dispatched from the depot. We assume that there is a sufficiently large amount of vehicles at the depot such that routes can always be assigned to a vehicle at the depot. All customers in a route assigned to a vehicle, a *dispatched route*, are considered serviced and removed from all routing plans. The *final solution* consists of all the routes selected for dispatching throughout the operating day.

5

5.3. SOLUTION METHOD

The proposed dynamic variant of the VRPRDL is solved through a *rolling horizon* framework using a *Multiple Plan Approach* (MPA) to include dynamic information and support decisions during the execution of the routes. In the framework, periodic re-optimizations are performed whenever there is an update to the input data (i.e., a customer announces a new location), keeping information regarding promising solutions on an adaptive memory (Taillard et al., 2001). Those solutions are maintained throughout the horizon and can be seen as alternative plans to better accommodate (future) information. In the following, we describe the framework in more detail and elaborate further on the procedures to maintain the solution pool and how it is used to assist on dispatching and routing policies.

5.3.1. MULTIPLE PLAN APPROACH

The MPA was introduced by Bent and Van Hentenryck (2004) and is a generalization of the adaptive memory framework proposed by Gendreau et al. (1999), who introduced a parallel tabu search using multiple solutions. The MPA generalizes the work by Gendreau et al. (1999) making it independent of the search procedure used to generate solutions. The general idea is to populate and maintain a pool with multiple plans that is used to select a *distinguished plan* upon which decisions are based on. The pool is updated periodically, ensuring that all routing plans are coherent with the current state of information and past decisions. The selection of a distinguished plan is accomplished using a consensus function ranking, at each time t, all plans based on their similarity to other plans in the pool. As noted in Bent and Van Hentenryck (2004), better results can be achieved when using a function ranking the plans based on their similarity than selecting the one with the smallest cost.

At each time $t \in [0, T]$, let the pool be represented by $\Omega_t = \{\sigma_1, \sigma_2, ..., \sigma_K\}$, where σ_i is a routing plan consistent with the current information at time t (i.e., a solution to the VRPRDL considering locations $\mathscr{L}^t$). Moreover, let plan $\sigma_i \in \Omega_t$ be specified by $\sigma_i = \{\sigma_{i,1}, \sigma_{i,2}, ..., \sigma_{i,m}\}$, where $\sigma_{i,j} = (0, \ell_{j_1}, ..., \ell_{j_p}, 0)$ is a route servicing customers at locations $\ell_{j_1}, ..., \ell_{j_p} \in \mathscr{L}^t$. Routes do not necessarily prescribe departure times from a location, but rather imposes constraints on them. In particular, for a route $\sigma_{i,j} = (0, \ell_{j_1}, \ell_{j_2}, ... \ell_{j_p}, 0)$, we keep *effective* time windows $[\alpha_0^{ij}, \beta_0^{ij}]$ and $[\alpha_{j_k}^{i}, \beta_{j_k}^{i}]$, for $k = 1, ..., p$, specifying the earliest and latest times a vehicle assigned to $\sigma_{i,j}$ would have to depart from the depot and from locations ℓ_{j_k}, respectively, to ensure that all customers in the route can be visited within their indicated time windows. A route σ_{ij} in the pool is consistent at time t if it has effective windows for departure at the depot, $[\alpha_0^{ij}, \beta_0^{ij}]$, such that $\alpha_0^{ij} = t$ and $\beta_0^{ij} > t$.

The MPA proposed in this work considers decisions regarding the set of customers to assign to a route and when to dispatch a route with assigned customers. Customers requiring delivery service can only be assigned to routes not yet dispatched from the depot. Locations from customers requiring pickups, however, can be included into routes assigned to already dispatched routes. At the start of the operating day, the pool consists of routing plans visiting all customers only at home. Over the course of the day, whenever a new location update from a customer arrives, we consider servicing that customer at this new location. To this purpose, we try inserting the location in a route (not visiting the associated customer) on each routing plan in the pool. The pool, thus, maintains a number of routing plans, with routes using different locations to service a given customer – for example, a customer might be serviced at his/her current roaming location in one route, but at home in another route. Pool updates are performed periodically, ensuring that all routing plans are coherent with the current state of vehicles and customers (locations).

5.3.2. INITIAL PLAN GENERATION

Some dynamic vehicle routing problems assume that part of the input data (e.g., customers) is known at planning time whereas some other part is revealed dynamically, defining a *partially* dynamic problem. This is the case for the problem considered in this work: we assume that part of the customer's itinerary is known at planning time, namely, the home location and the associated time windows, whereas other locations are announced throughout the time horizon, during execution time. Thus, the initial pool Ω_0 is populated by a number of routing plans, all visiting customers only at home. As customers' itineraries are revealed, these plans get modified, and some customers might be serviced at an announced location instead of at home. Within a plan, new routes may be created, but new plans are not created i.e., $|\Omega_0| = |\Omega_t|\ \forall\, t \in [0, T]$. In particular, at each time during the time horizon, all plans in the pool contain a route visiting not-yet serviced customers, $c \in \mathscr{C}^t$, at home.

5.3.3. EVENT HANDLING

Our MPA framework takes into consideration three types of events, namely, (1) announcement of a new location by a customer, (2) vehicle dispatching and (3) route timeouts. The MPA handles (1) by evaluating the insertion of any newly arrived location at the routing plans in the pool as well as performing the required updates. Vehicle dispatching decisions (2) are taken using a consensus function specifying which routes in the pool to dispatch at a given time. We will discuss the function later in this section. Events (3) concern routes in the plan (i.e., not yet dispatched) that have become infeasible at a given time. The MPA handles such events by re-evaluating assignment and routing decisions in any plan which triggered a timeout, making it consistent with the current state.

For each of the events considered, we specify how the current status of the pool transitions from Ω_t to Ω_{t+1} and how the consensus function is defined and used when taking dispatching decisions. Different from our MPA, we note that in Bent and Van Hentenryck (2004) the consensus function is used to select a distinguished plan $\sigma_i^* \in \Omega_t$ at each time t. The distinguish routing plan defines the next action to take (i.e., which customer to visit next), and this plan can change throughout the time horizon. In our proposed MPA, we use the consensus function to decide on which routes to dispatch, selected among all routes in all routing plans in the pool. Once a route is selected, a vehicle is assigned to perform that route – loaded with parcels and dispatched. During the execution of the route, additional parcel pickup requests at customer locations can be assigned to the vehicle as long as the route remains feasible in terms of its time-windows and capacity. Thus, our consensus function selects a distinguished route, $\sigma_{i,j}^*$, to dispatch and that will define a set of actions to take. Moreover, observe that at time $t \in [0, T]$, even if no event occurs, the natural transition to time $t+1$ consists basically in updating the effective time windows of the routes in the pool.

TIMEOUT

A timeout event is triggered at time $t = \beta_0^{ij}$ if the route $\sigma_{ij} \in \sigma_i$ (in the pool Ω_t) is not assigned to a vehicle. In this case, the route has become inconsistent as it can no longer serve all its customers within their respective time windows if dispatched after t. For this event, we re-evaluate the route (assignment and sequencing decisions) to make it consistent again, such that, after the re-evaluation, $\beta_0^{ij} > t$ holds. In particular, given a route $\sigma_{ij} = (0, \ell_{j_1}, \ldots, \ell_{j_p}, 0)$ triggering a time-out event, let $I \subseteq \{\ell_{j_1}, \ldots, \ell_{j_p}\}$ be the ordered set of (customers') locations for which $\alpha_k^{ij} = \beta_k^{ij}$ i.e., there is no slack in the departure time from location ℓ_k if the route departs from the depot at time t. Each location in I is removed from the route (in order) and earliest times of subsequent locations recomputed, until all earliest times from locations remaining in $k \in I$ are changed such that $\alpha_k^{ij} < \beta_k^{ij}$ and $\beta_0^{ij} > t$. Let $I' \subseteq I$ be the set of removed locations. We try to re-insert those locations in the plan σ_i, but considering earliest dispatching time from the depot as $\alpha_0^{ij} = t+1$ for all routes j in σ_i. In case re-inserting the location in the plan is not possible (even by creating a dedicated route), the location is not considered for servicing the associated customer in the given routing plan. The operations are summarized as:

$\forall \sigma_i \in \Omega_t :$
$\quad T \leftarrow \{\sigma_{ij} \in \sigma_i \mid \text{TIMEOUT}(\sigma_{ij}, t)\}$
$\quad I' \leftarrow \varnothing$
$\quad \forall \sigma_{ij} \in T :$
$\quad\quad \sigma_{ij} \leftarrow \text{COMPATIBLE}(\sigma_{ij}, t+1, I')$
$\quad \sigma_i \leftarrow \text{INSERT}(\sigma_i, t+1, I')$
$\Omega_{t+1} \leftarrow \Omega_t$

where TIMEOUT(σ_{ij}, t) returns whether route σ_{ij} became inconsistent at time t, COMPATIBLE$(\sigma_{ij}, t+1, I')$ returns a route in which the incompatible route σ_{ij} has been made compatible at time $t+1$, possibly removing customers and adding them on the set I'. INSERT$(\sigma_i, t+1, I')$, returns a routing plan in which locations in I' have been inserted in one of the routes in σ_i, minimizing insertion costs, and assuming departure time from the depot at $t+1$, in case feasible insertions exist.

LOCATION ANNOUNCEMENT

Whenever a not yet serviced customer, $c \in \mathscr{C}^t$, announces availability at his/her current location, ℓ_k^c, at time t, we try to accommodate such information in each routing plan $\sigma_i \in \Omega_t$, inserting the new location in a route $\sigma_{ij} \in \sigma_i$ at minimal cost. The plan σ_i might already contain a route servicing customer c at ℓ_h^c (home). In that case, we also evaluate removing the home visit and inserting the new announced location in that route. If the latter is the minimum insertion among all feasible insertions, it is performed and the home visit (ℓ_h^c) is removed and inserted into another route, at minimum cost. In this way, every plan in the pool contains a route servicing customers at home. This is done in order to assure that, in case a route visiting customers at dynamically announced locations is never dispatched, a routing plan resorting to home visit is always available. Since home locations and associated time windows are known from the beginning of the time horizon, the latest time a route has to be dispatched to visit a customer at home is also known. Thus, we can define:

$\Omega_{t+1} \leftarrow \Omega_t$
$\forall \sigma_i \in \Omega_{t+1}$
$\quad \sigma_i \leftarrow \text{INSERT}(\sigma_i, \ell_k^c)$

where INSERT(σ_i, ℓ_k^c) returns a plan in which location ℓ_k^c has been inserted in one of the routes $\sigma_{ij} \in \sigma_i$ (if a feasible insertion exists), minimizing insertion cost. The insert operation might add a new route in the plan σ_i in case location ℓ^c_k can only be reached in time by a dedicated route.

When a customer requiring pickup service announces a location, observe that routes assigned to a dispatched vehicle might be able to serve that request. Our proposed MPA evaluates the insertion of the request as described above, but also considers (feasible) insertions on routes already dispatched as follows. After evaluating the best insertion on a route of a vehicle already dispatched (if one exists), we also compute the average insertion cost on routes in the pool (as described). In case the cost of the (best) insertion in a dispatched route is lower than the average insertion cost in the pool, the insertion is performed in the dispatched route. This may be viewed as a way of capturing situations when, even though a cheap insertion might exist in the pool, dispatching a route servicing that customer has limited consensus value and detouring a dispatched vehicle seems more promising.

CONSENSUS-BASED VEHICLE DEPARTURE

Given the windows for departure at the depot, $[\alpha_0^{ij}, \beta_0^{ij}]$, a route $\sigma_{i,j}$ in the pool has to be assigned to a vehicle, and the vehicle dispatched, at latest at time β_0^{ij}. We use a similar idea as in Bent and Van Hentenryck (2004), and define a consensus function $f : S \mapsto \mathbf{Z}_{\geq 0}$ to rank a route in a set S accordingly to its similarity to all other routes in S. More precisely, given two routes $\sigma_{i_1 j}$ and $\sigma_{i_2 k}$, let $C_{i_1,j}, C_{i_2,k} \subseteq \mathscr{C}$ be the set of customers visited by routes $\sigma_{i_1 j}$ and $\sigma_{i_2 k}$, respectively. Then, the similarity between the two routes is defined as:

$$s(\sigma_{i_1,j}, \sigma_{i_2,k}) = \frac{|C_{i_1,j} \cap C_{i_2,k}|}{\max(|C_{i_1,j}|, |C_{i_2,k}|)} \tag{5.1}$$

it follows that routes servicing the same set of customers, have maximum similarity. Let the maximum similarity of a route $\sigma_{i_1,j}$ when compared to all routes in some plan $\sigma_{i_2} \in \Omega_t$, be given by:

$$s(\sigma_{i_1 j}, \sigma_{i_2}) = \max(s(\sigma_{i_1 j}, \sigma_{i_2 1}), s(\sigma_{i_1 j}, \sigma_{i_2 2}), \ldots, s(\sigma_{i_1 j}, \sigma_{i_2 m})) \tag{5.2}$$

The consensus value for a route $\sigma_{i,j} \in \sigma_i$ is then defined as:

$$f(\sigma_{ij}) = \sum_{p=1:p\neq i}^{K} s(\sigma_{ij}, \sigma_p) \tag{5.3}$$

where the $\Omega_t = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$ are the plans in the pool at a given time t. Observe that, by definition of equations 5.2 and 5.3, it holds that $s(\sigma_{i_1 j}, \sigma_{i_2 k}) \leq 1$ and $f(\sigma_{ij}) \leq K - 1$. Dispatching decisions are taken as follows. Routes are never dispatched if $\beta_0^{ij} - \alpha_0^{ij} > W$, where W specifies the maximum available waiting time at the depot before dispatching. Let $S = \{\sigma_{i,j} : \beta_0^{ij} - \alpha_0^{ij} \leq W\}$ the set of all routes in the pool available for dispatching at time t. Let σ^*_{ij} be the route with the largest amount of consensus, i.e.,

$\sigma^*_{ij} = \text{argmax}_{\sigma_{ij} \in S} f(\sigma_{ij})$. We only dispatch a route if a minimum amount of consensus has been reached, that is, if $f(\sigma^*_{ij}) \geq \mu K$, where $\mu \in (0,1]$ is a pre-defined scalar. If more than one route attains the largest consensus observed, the route with the minimum (travel) cost is selected. Ties are broken arbitrarily.

When route σ^*_{ij} is selected for dispatching at time t, the pool needs to be updated to account for this action. In particular, the customers $C_{i,j}$ serviced by route $\sigma^*_{i,j}$ must be removed from Ω_{t+1}:

$\sigma_i \leftarrow \sigma_i \setminus \sigma^*_{ij}$
$\forall \sigma_k \in \Omega_t :$
  $\sigma_k \leftarrow \text{REMOVE}(\sigma_k, C_{i,j})$
$\Omega_{t+1} \leftarrow \text{UPDATE}(\Omega_t)$

where $\text{REMOVE}(\sigma_k, C)$ returns a routing plan in which all customers in C have been removed from the routes in plan σ_k. $\text{UPDATE}(\Omega_t)$ updates all earliest and latest departure times for all routes in all plans in the pool, considering earliest departure from the depot at time $t+1$.

5

5.4. COMPUTATIONAL EXPERIMENTS

We carried out our computational analysis on data-sets used in previous works evaluating roaming locations for last-mile services. First, we elaborate on the characteristics of the VRPRDL instances used for conducting the experiments in section 5.4.1. Using those instances, we show that routing plans making use of locations announced by customers, in a dynamic fashion, can improve over routing plans optimized using only customer home locations. The rolling horizon framework used to simulate the dynamic setting is presented in section 5.4.2. In section 5.4.3, we show how we obtain the routing plans using only home locations. The only-home solution is compared against solutions obtained in the dynamic context, using our proposed MPA. All strategies used to initialize the pool are presented in 5.4.4. We report the numerical results in section 5.4.5, where we also evaluate the impact of a customer announcing his/her coming location in advance and the inclusion of pickup requests. Our proposed algorithms are implemented in C++ and experiments executed on an Intel Xeon E5-2666v3 CPU @3.5GHz machine, 15GiB, running Ubuntu Server 18.04. Problems formulated as Mixed Integer Programming (MIP) models are solved using Gurobi 8.0.1 as the MIP solver.

5.4.1. INSTANCE DESCRIPTION

We use the set of *realistic instances* proposed by Reyes et al. (2017). These instances have customers divided in: only at home, at home and work, and at home, work and somewhere else after work (gym, shopping mall, etc.). Work locations are divided into eight pre-defined work clusters, inspired by the geography of Atlanta, US. The eight clusters represent major regions and are centered accordingly to their location in the map of the city. The operation period, $[0, T]$, comprises 14 hours and a customer itinerary is such that the first and last locations are the customer's home, with first and second home time

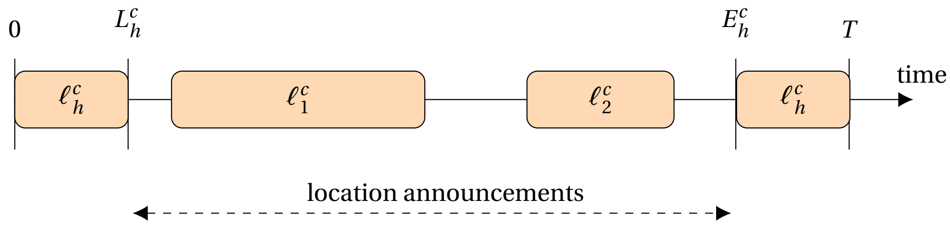


Figure 5.2: Example of a customer itinerary in a *realistic instance* proposed in Reyes et al. (2017). The customer is available at home for a few hours in the morning, leaves to work, where he/she spends most of the day. After work, pays a visit to the gym, before going back home.

windows $[a^c_{h_1}, b^c_{h_1}] = [0, L^c_h]$ and $[a^c_{h_2}, b^c_{h_2}] = [E^c_h, T]$, respectively, for a customer c (L^c_h is the time at which customer c leaves home and E^c_h the time at which he/she is back). Figure 5.2, illustrates an example for a customer visiting two locations (excluding for home) during the day. In these instances, customers spend only a fraction of the total time horizon travelling i.e., most of the time the customer's car is parked. Table 5.1 shows the characteristics of the instance set. We consider a total of 40 instances, divided in 4 groups with 60, 90, 120 and 150 customers each. On all instances, exactly 10% of customers are available at home during the whole period and, for the remaining customers, the average itinerary size (excluding home) is 1.6. Columns Average TW width, Time avail. for service at home and Time avail. at an announced location report the corresponding values, as a percentage of the full time horizon, considering only customers that are also available at locations other than home (90% of customers). The geographic area, R, where customers' locations are to be serviced is a squared region, 120 x 120 minutes^2, and the central depot is located in the center of R i.e., any point can be reached from the depot within, at most, $60\sqrt{2}$ minutes.

5.4.2. ROLLING HORIZON FRAMEWORK

The proposed MPA is applied over a rolling horizon framework in which each of the events described in Section 5.3.3 triggers a local update of the pool. Updates are based on insertion heuristics, providing reaction to incoming information in short computational time. To improve the final solution quality, a route is (re)-optimized for a short period (10 seconds), by solving an associated Travelling Salesman Problem, when selected for dispatching. This is in line with most dynamic vehicle routing methods based on re-optimization, where fast heuristics are used to (quickly) react to information updates and more complex re-optimization methods are applied periodically for further improvement (Bektas et al., 2014a).

The implementation of the rolling horizon framework is based on a solver-simulator-controller feedback loop, similar to the one proposed in Larsen and Pranzo (2018). Figure 5.3 illustrates the overall architecture of the framework. The simulator module is responsible for advancing the time and communicating customer events. In particu-

				% Time avail. for service at home			% Time avail. at an announced location		
Inst.	$\|C\|$	$\sum_c N_c$	Average TW width(%)	Avg	Min	Max	Avg	Min	Max
0	60	198	25.2	30.0	0.7	55.7	57.7	0.7	57.1
1	60	198	25.1	31.3	1.0	51.8	55.9	0.2	56.5
2	60	198	25.1	30.3	1.0	53.6	57.0	0.4	57.1
3	60	198	25.2	32.3	0.1	45.1	55.3	0.7	56.9
4	60	198	25.0	30.3	0.4	50.5	56.4	0.4	57.1
5	60	198	24.8	29.3	0.7	39.8	56.8	2.0	57.1
6	60	198	24.9	33.2	2.1	54.5	53.4	0.8	57.1
7	60	198	25.2	31.2	0.1	50.4	56.3	0.2	57.0
8	60	198	24.9	31.2	0.5	39.4	55.2	1.0	57.1
9	60	198	25.1	31.6	0.2	50.5	55.5	0.2	57.1
10	90	297	24.9	31.9	0.8	45.4	54.6	0.1	56.8
11	90	297	25.1	32.4	0.5	52.5	54.7	0.6	57.1
12	90	297	24.7	29.8	0.4	49.2	56.0	0.5	56.8
13	90	297	24.9	30.5	0.6	52.6	56.0	0.0	57.1
14	90	297	25.0	30.7	0.0	52.0	56.2	0.2	57.1
15	90	297	25.0	31.7	0.7	52.1	55.1	0.7	57.1
16	90	297	25.2	32.3	0.7	51.5	55.3	0.1	57.0
17	90	297	24.8	31.2	1.4	53.7	55.1	0.5	57.1
18	90	297	24.9	30.0	0.8	54.0	56.5	0.4	57.0
19	90	297	25.0	31.6	0.1	54.0	55.0	0.1	57.1
20	120	396	24.9	31.4	0.0	52.6	55.1	0.1	57.1
21	120	396	25.0	30.9	0.5	52.7	55.8	0.4	57.0
22	120	396	25.4	31.9	1.2	43.3	56.2	0.2	57.1
23	120	396	24.9	31.7	0.1	52.7	54.8	0.2	57.0
24	120	396	25.0	31.9	0.4	53.7	55.0	0.4	57.0
25	120	396	24.9	30.8	0.0	54.4	55.8	0.0	57.1
26	120	396	25.0	30.8	0.8	54.3	55.9	0.2	57.1
27	120	396	25.0	31.1	0.1	51.5	55.9	0.0	57.0
28	120	396	25.1	30.6	0.4	55.5	56.6	0.8	57.1
29	120	396	24.8	31.1	0.4	52.7	55.1	0.4	57.1
30	150	495	25.0	31.6	0.7	51.9	55.4	0.2	57.1
31	150	495	24.9	30.3	0.1	53.7	56.2	0.5	57.0
32	150	495	25.1	31.5	0.1	54.5	55.8	0.0	57.1
33	150	495	25.0	31.6	0.1	51.5	55.3	0.1	57.1
34	150	495	24.9	31.3	0.0	54.0	55.3	0.0	56.9
35	150	495	25.0	30.6	0.2	54.5	56.2	0.0	57.1
36	150	495	25.0	30.9	0.2	53.7	55.9	0.1	57.1
37	150	495	25.1	31.1	0.0	54.4	56.3	0.4	57.1
38	150	495	24.9	31.0	0.1	54.4	55.5	0.2	57.1
39	150	495	25.1	31.8	0.2	56.5	55.4	0.1	57.1

Table 5.1: Characteristics of the *realistic instances* proposed by Reyes et al. (2017).

lar, the simulator module reads the (full) VRPRDL instance – containing all customer's itineraries and time windows – and simulates the announcements of the locations to the controller module as time progresses. Moreover, the module keeps track of the progression of vehicles movements through time, such as when customers are visited and when vehicles return to the depot (potentially becoming available for dispatching again). The controller module implements the MPA by maintaining and updating the pool through the solver module, deciding when to call it. Note that the controller module is limited by the information provided by the simulator – only information up to time t in the VRPRDL instance is known to the module. Additionally, the controller implements the consensus function and informs the simulator when a route is selected for dispatching.

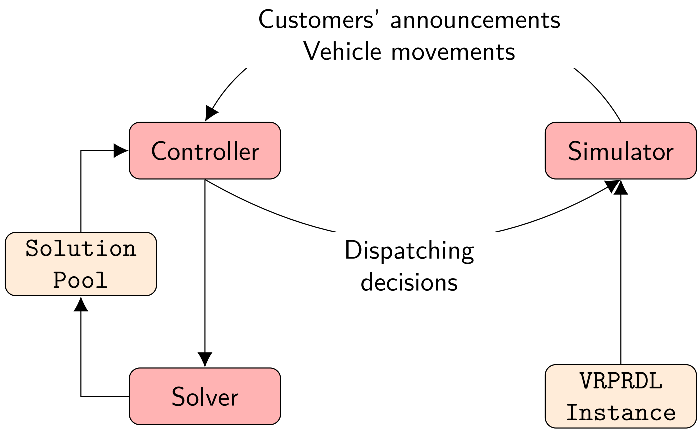


Figure 5.3: Dynamic framework architecture and relationships.

The dynamics of customers' locations announcements are considered in two ways. In the first, we assume customer $c \in \mathscr{C}$ announces a location ℓ_i^c in his/her itinerary as soon as the customer arrives at that location. In the second, we consider that the customer announces the next location in his/her itinerary at the time he/she leaves his/her present location. This effectively gives the system a lead-time announcement, γ, equal to the travel time from one location to the next in the itinerary. Figure 5.4 illustrates the two approaches.

5.4.3. ONLY-HOME SOLUTION

After all customer requests have been received – mostly likely, using a cut-off time for order arrival – the company has the time allotted for the planning phase to decide on a routing plan with the received information (home locations) before the order execution phase (see Figure 5.1b). The home-only solution is obtained by solving an MIP formulation for the VRPRDL (refer to constraints 4.3–4.11 in Chapter 4) considering only home locations and minimizing travelling costs, waiting times and vehicles used. Due to this

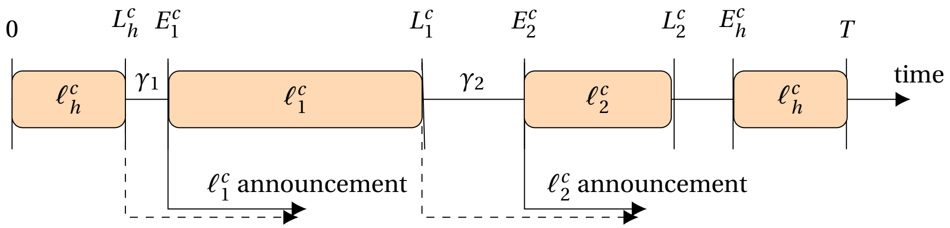


Figure 5.4: Example of a customer itinerary with two locations (except for home), ℓ_1^c and ℓ_2^c. In the first announcement scheme (solid lines), ℓ_1^c and ℓ_2^c are announced at times E_1^c and E_2^c, respectively, i.e., with a lead time $\gamma = 0$. Under the second scheme (dashed lines), ℓ_1^c is announced at L_h^c, with a lead-time $\gamma_1 = E_1^c - L_h^c$. Similarly, ℓ_2^c is announced at L_1^c, with a lead-time $\gamma_2 = E_2^c - L_1^c$.

	(Only) Home Solutions			
Inst.	Obj	Routes	Fleet	Range
C_{60}	1084.1	7.4	5.7	38.8
C_{90}	1620.6	9.9	7.5	39.0
C_{120}	1950.3	13.7	10.0	39.0
C_{150}	2121.6	15.5	11.0	38.6

Table 5.2: Solutions servicing customers only at home, customers are serviced only at home, either at the early or late windows.

objective function, any route visiting customers at early and late home time windows incurs in very large waiting times (and, consequently, high cost). Thus, a home-only solution comprises a set of routes servicing customers early (during the first home time window) and a set of routes servicing customers late (during the second home time window). We limit the solving of the model to eight hours, a reasonable time expend during planning phase for the operational problem considered.

In Table 5.2, we report the results obtained for the only-home solutions. Each line in the table is the average result over the 10 instances in a given class (column Inst.), detailed results for each particular instance in a class are presented in appendix C. We also report the routing plan cost (travel time and waiting, if any) in column Obj., the number of routes in column Routes, and the required fleet in column Fleet. Column Range reports the average distance from the depot to the locations used to service customers in the routing plan. The only-home solutions are static solutions, and provide the baseline for assessing the dynamic solutions obtained with our proposed MPA, when customers announce locations along their daily itineraries (other than home) where they can be serviced.

5.4.4. INITIAL SOLUTION POOL

At the beginning of the service day, only home locations and associated time windows (early and late) are known. Thus, the initial routing plans considered in the MPA contain routes only visiting customers at home. We consider three strategies for obtaining initial solutions: i) the only-home solution, as 5.4.3; ii) servicing customers only during their late home time windows, and iii) servicing the $\rho|C|$ closest customers to the depot in their early home time window and all others in their late home time window, with $0 < \rho < 1$. To solve ii) and iii), we resort to the same MIP formulation used to obtain the only home-solution but add appropriate constraints to impose the desired solution characteristics (i.e., imposing which time windows should be used or not for a given customer).

In order to populate the initial solution pool, Ω_0, is initialized with the 100 (or less, if not that many are found) best solutions found when solving the MIP formulation correspondent to the strategy used to obtain initial routing plans servicing customers at home. Solutions which are more than twice as costly as the best solution found are discarded. At time $t = 0$, following the best solution found, routes servicing customers at their early time windows are dispatched. Routes servicing customers at late time windows are kept in the pool.

The choice of an initialization strategy affects how the MPA is able to accommodate future location announcements from customers. By having initial routing plans servicing all customers at home during late time windows, we can postpone service and wait for customers announcing other locations along their daily itinerary that could potentially yield a more favorable routing plan. Some customers might have itineraries visiting locations far from the depot, thus by servicing customers nearby the depot during their early time windows we can avoid drivers visiting locations potentially far away from the depot.

5

5.4.5. RESULTS

To assess the benefits of customers dynamically providing locations for servicing, we compare only-home solutions with the solutions produced by the proposed MPA. The MPA solution consists of the final routing plan, at the end of the rolling horizon, containing all dispatched routes. In the experiments, we evaluate different values for W, the maximum available waiting time at the depot before dispatching of a route. Different strategies for starting the MPA and populating the pool are evaluated. Moreover, we also conduct experiments to show the impact of customers announcing in advance (lead time $\gamma > 0$) the next location at which they will be available for service.

EVALUATING THE MPA SOLUTIONS

We first evaluate the results obtained when customers only require delivery service (the original setting of the VRPRDL) and with no announcement lead-time (i.e., $\gamma = 0$). In the first set of experiments, we assess how the dynamic solutions behave given i) different waiting times, W, before dispatching.ii) the strategy used to provide initial routing plans servicing customers at home.

Strategy HM refers to using the solutions found when solving the only-home model to populate the initial solution pool, Ω_0 (i.e., the best solution found is the only-home

	Dynamic Solutions, $\gamma = 0$						
Inst.	St	W	Δ_o(%)	Roam.(%)	Routes(%)	Fleet(%)	Range
C_{60}	HM	15	-14.6	72.0	33.8	-22.8	26.7
	$0E$	30	-24.9	88.5	31.1	-22.8	23.9
	$10E$	15	-22.3	78.0	32.4	-14.0	22.7
	$25E$	60	-18.4	64.3	21.6	-24.6	23.8
C_{90}	HM	30	-23.0	73.4	45.5	-18.7	25.7
	$0E$	60	-29.6	87.8	35.4	-12.0	23.4
	$10E$	30	-22.9	78.0	63.6	-10.7	23.5
	$25E$	30	-18.9	63.7	55.6	-8.0	24.4
C_{120}	HM	30	-9.4	69.2	41.6	-15.0	27.9
	$0E$	30	-21.3	87.1	34.3	-1.0	24.4
	$10E$	30	-20.1	78.9	36.5	-15.0	24.5
	$25E$	15	-14.2	63.8	42.3	-20.0	25.1
C_{150}	HM	60	-10.2	68.8	41.9	-15.5	25.6
	$0E$	30	-18.3	86.9	43.9	13.6	24.0
	$10E$	15	-14.0	77.6	45.8	-8.2	23.6
	$25E$	60	-16.0	63.9	32.9	-25.5	23.7

Table 5.3: Best dynamic solutions obtained by the MPA starting with different initial routing plan strategies ($St = \{HM,\ 0E,\ 10E,\ 25E\}$). Each row shows the average results over the 10 instances in the correspondent class.

solution in Table 5.2). In strategies xE $x \in \{0, 10, 25\}$, for populating Ω_0 we impose that x% of the customers, those closest to the depot, are serviced in their early time windows at home. Thus, $0E$ means that no customer will be serviced in his/her early home time windows and are initially considered for service at the late home time windows.

Table 5.3 reports results for instances in each class (C_{60}, C_{90}, C_{120} and C_{150}). Each row in the table is the average over the 10 instances in a class, and shows the waiting time before dispatching (W, in the same unit as travel and wait times, e.g., minutes) which provided the best (average) solutions for a given start solution strategy (St). The MPA solution is compared to the only-home solution in terms of the relative difference in cost, number of dispatched routes (Routes) and number of vehicles used (Fleet), computed as $\Delta_o(\%) = \frac{c_d - c_h}{c_h}$, where c_d and c_h are the costs of the MPA and only-home solutions, respectively. The relative difference in routes and fleet size are computed similarly. Columns Roam. and Range show the percentage of visits at roaming locations and the average range of the routing plans (Range), respectively.

Overall, the MPA approach is able to successfully integrate the dynamically announced (roaming) customers locations and achieve final routing plans improving over the only home (static) solutions. We observe that solutions with larger improvements on total route cost (Δ_o) tend to include more roaming locations (Roam.) in the final routing plan and improvements of up to 30% could be obtained. The characteristics of the initial routing plans also has a significant impact on the quality of the solutions achieved by the MPA. Starting with not servicing customers in the early home time windows ($St = 0E$)

tend to provide final solutions improving the most on total route cost, but usually requires a slightly higher vehicle fleet. Moreover, with this strategy, the MPA is able to service more customers at roaming locations. As the routing plans initially consider only visits at home during late time windows, the routes are able to better accommodate dynamically announced locations. The drawback of such approach is that, as the routing plans in the pool change over time, customers whose announced locations are not inserted in a dispatched route are left in short routes in the pool, consolidating only a few customers. Considering the delivery to customers during their early home time windows ($10E$, $25E$) provide solutions which tend to contain early routes consolidating many customers and avoid having to dispatch vehicles consolidating only a few customers late in the service day.

AVAILABLE TIME UNTIL DISPATCH

Regarding the available waiting time at the depot, W, before dispatch, larger values of W result in quicker dispatching, if consensus is achieved. Observe that if $\beta_0^{ij} - \alpha_0^{ij} \le W = 0$ for a route σ_{ij}, but consensus is not achieved for dispatching, then σ_{ij} times out and the route is modified as discussed in Section 5.3. For $W > 0$, route σ_{ij} does not necessarily time-out when $\beta_0^{ij} - \alpha_0^{ij} \le W$ and will remain unmodified in the pool for longer. Thus, consensus is achieved on improving solutions more often for $W > 0$ than for $W = 0$. In Table 5.3 we show the values of W for which we obtained the best solutions but, for a given initial routing strategy, St, the solutions obtained with $W > 0$ are of similar quality (around no more than 5% higher than the best).

For a giving start strategy, St, dispatching the routes at the latest time possible (i.e., $W = 0$), resulted in the worse final solutions. As consensus is reached less often than for larger values of W, less roaming locations are included in the routing plans. Moreover, as route time-outs occur more often, the routes reaching consensus tend to consolidate less customers and, consequently, more vehicles are required. Dispatching vehicles sooner (larger W values), have a less significant impact but still also can be detrimental to the dynamic approach. Whereas, on average, more roaming locations can be visited, when the solutions obtained $W = 60$ have a higher improvement compared to $W = 15$ or $W = 30$, a slightly higher fleet is required. This is to be expected, as early dispatching can lead to the MPA missing opportunities to better consolidating customers.

DRIVERS DISPERSION OVER TIME

One aspect of servicing customers at the dynamic (roaming) locations is that it may allow company drivers servicing customers (knowing that the customer is available) at times covering a larger part of the time horizon. In contrast, only-home solutions visits customers at early or late time windows (assuming that, knowing the customer is not home, a delivery attempt will not be made), and drivers are idle for long periods. In Figure 5.6, we show how drivers are being used throughout the planning horizon considering two solutions, using two different strategies for the initial routing plans. Customers announce locations in between the home visits (early and late), and the MPA integrates some of this information on routes servicing customers at late time windows. Routes (drivers) are then active during the middle of the planning horizon and able to finish servicing all customers earlier than the only-home solutions and in total fewer drivers are needed.

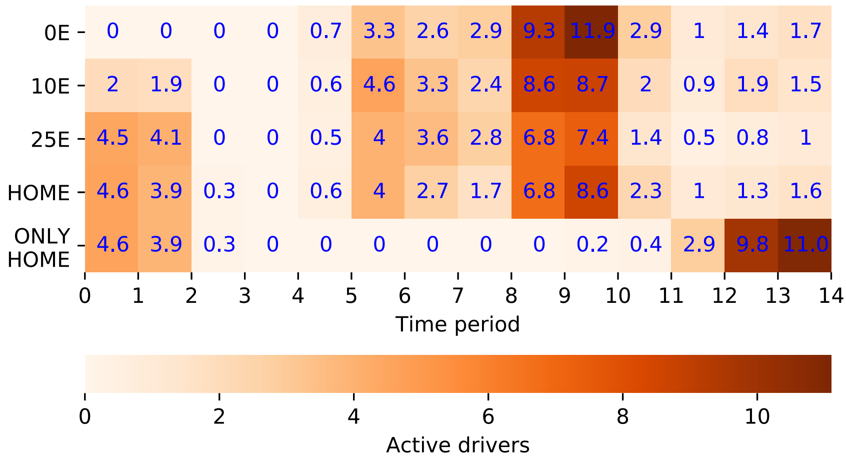


Figure 5.5: Time periods showing when routes are active in the dynamic solution, for each start solution, throughout the time horizon. A routes is active when it has been assigned to a vehicle, the vehicle has been dispatched and not yet finished servicing all customers in the route. Each time period comprises one hour. Results are the average for all instances in C_{150}, $St = 25E$ and $W = 30$.

THE VALUE OF INFORMATION

We also evaluate the routing plans obtained with the MPA regarding the *value of information*. In that case, the VRPRDL instance is solved as a static and deterministic problem, considering all information regarding customers' itineraries is known in advance and does not change during the planing horizon. The solution represents the best solution that could potentially be implemented in a dynamic setting, and is compared to the MPA solution. To that end, we solve the (full) instances using a MIP formulation for the VRPRDL and Gurobi 8.0.1 as the MIP solver, with a time limit of 12 hours. Due to the size of the instances, we only perform theses experiments for instances in classes C_{60} and C_{90}. We report the results in Table 5.4. Column Dynamic reports the best (average) value obtained with the settings providing the best results in the MPA ($W = 30$ and $St = 0E$), where Gap= $\frac{c_d - Obj}{Obj}$ and c_d is the cost of the solution obtained by the MPA.

ANNOUNCEMENT LEAD-TIME

In this part we consider scenarios with a positive announcement lead-time, $\gamma > 0$, i.e., the customer announces the next location he/she will visit when leaving his/her current

	All Information				Dynamic			
Inst.	Obj.	Roam.	Routes	Range	Gap	Roam.	Routes	Range
C_{60}	779.1	49	6	21.9	0.05	53	9	23.9
C_{90}	1063.5	73	11	22.8	0.08	78	13	23.7

Table 5.4: Value of information compared to the MPA (dynamic) solutions.

	Dynamic Solutions, $\gamma > 0$						
Inst.	St	W	Δ_o(%)	Roam.(%)	Routes(%)	Fleet(%)	Range
C_{60}	HM	15	-19.5	63.7	13.5	-31.6	24.1
	$0E$	30	-24.8	88.3	27.0	-22.8	23.9
	$10E$	60	-25.8	79.2	25.7	-29.8	21.9
	$25E$	60	-19.9	64.2	14.9	-31.6	24.1
C_{90}	HM	60	-21.9	74.1	49.5	-17.3	25.4
	$0E$	60	-31.8	88.2	29.3	-8.0	23.4
	$10E$	30	-23.5	78.6	59.6	-13.3	23.1
	$25E$	15	-19.9	63.4	54.5	-14.7	24.3
C_{120}	HM	15	-14.9	64.4	40.1	-20.0	24.9
	$0E$	30	-22.7	86.5	29.2	-3.0	24.7
	$10E$	15	-19.4	78.0	35.0	-13.0	24.5
	$25E$	30	-15.2	64.3	38.0	-21.0	24.7
C_{150}	HM	15	-18.2	63.6	29.0	-23.6	23.3
	$0E$	60	-20.4	88.3	37.4	-0.9	23.3
	$10E$	60	-15.3	78.8	43.2	-4.5	23.0
	$25E$	15	-18.2	63.6	29.0	-23.6	23.3

Table 5.5: Best dynamic solutions obtained by the MPA starting with different initial routing plan strategies ($St = \{HM,\ 0E,\ 10E,\ 25E\}$). Each row shows the average results over the 10 instances in the correspondent class.

location (see Figure 5.4). Table 5.5 shows the (best) results obtained, for each of the initial routing plan strategies.

We observe that under lead-time announcement best solutions are achieved more often with $W = 60$ than in case of no lead-time announcement (Table 5.3). With customers announcing locations in advance, that information is available for longer in the solution pool with consensus being achieved more frequently on routes with larger available waiting times before dispatched at the depot, leading to faster dispatching. Starting with routing plans servicing customers at home during late time windows provided the largest reduction in costs but, similarly to the case for $\gamma = 0$, at the expense of a relative larger fleet than the other strategies. On average, a reduction of 3% on cost (Δ_o(%)) and 4% on the required fleet could be achieved by having customers announcing locations along their itinerary in advance.

When a customer has to announce a location in advance, although the system might benefit from having the information earlier (in the case of the proposed MPA, consen-

5

sus is achieved faster), a downside in a real last-mile system implementation is that it can introduce a source of uncertainty, namely, if the customer will indeed reach the announced location within the informed time windows.

Figure 5.6 illustrates when drivers are active for solutions with and without lead-time announcement, considering starting the MPA with routing plans servicing customers late at home ($St = 0E$). Note that start of service (routes being dispatched) is similar in both cases, around the fourth hour, but more drivers are dispatched initially for $\gamma > 0$. Moreover, the peak on the number of simultaneous active drivers occur at similar times, but a slight lower number of drivers is required for $\gamma = 0$.

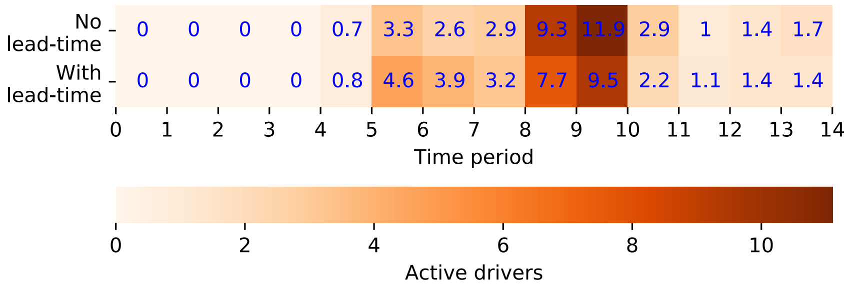


Figure 5.6: Time periods showing when vehicles are active in the dynamic solution, for each start solution, throughout the time horizon. Results are the average for all instances in C_{150} using 0E as the start method.

INCLUDING PICKUP REQUESTS

Next, we perform experiments considering that some (randomly selected) customers in a given instance require pickup service (e.g., returning of merchandising). The additional aspect introduced by considering pickups during execution is the fact that such services can also be fulfilled by vehicles already dispatched, as returning to the depot is not required (as opposed to delivery services).

Table 5.6 reports results for scenarios in which 25% of the customers require pickup operations. Column Picks shows the average number of pickups serviced by en-route vehicles (i.e., pickups inserted on an already dispatched route). Similarly to results considering only delivery requests, best solutions were achieved with $W > 0$ and we observe that only a few pickups could be included on already dispatched routes. However, for solutions with $W = 0$ e.g., dispatch routes at the latest possible moment, a significant greater number of announced pickup requests is inserted on such routes (on average, three times more). This is explained by the following. With $W = 0$, the urgency for dispatching is usually due to short (effective) time windows for locations visited at the beginning of the route, and effective time windows for the last locations tend to be wide enough to allow for a deviation due the insertion of a new location (a pickup request). For $W > 0$, insertions of newly announced (pickup) locations tend to be cheaper on routes in the pool than for routes already dispatched (when feasible), as the slack to

	Dynamic Solutions, $\gamma = 0$ and 25% pickup requests							
Inst.	St	W	Δ_o(%)	Roam.(%)	Routes(%)	Fleet(%)	Range	Picks
	HM	15	-17.1	71.7	28.4	-29.8	26.5	0.9
C_{60}	$0E$	15	-26.1	88.3	31.1	-19.3	22.7	0.2
	$10E$	15	-24.6	78.7	31.1	-14.0	22.8	0.5
	$25E$	30	-18.6	64.2	21.6	-24.6	23.7	0.5
	HM	30	-23.9	73.0	42.4	-14.7	25.9	1.7
C_{90}	$0E$	30	-30.7	87.4	38.4	-8.0	23.6	1.4
	$10E$	15	-24.4	77.8	59.6	-10.7	23.1	1.2
	$25E$	30	-20.3	63.8	54.5	-12.0	24.2	0.6
	HM	15	-11.9	67.2	38.7	-12.0	27.6	3.1
C_{120}	$0E$	60	-22.5	87.3	31.4	-3.0	24.4	1.4
	$10E$	30	-19.8	78.2	35.0	-14.0	24.4	1.0
	$25E$	60	-14.7	64.2	40.1	-23.0	25	0.2
	HM	60	-11.5	69.0	40.6	-8.2	25.7	2.2
C_{150}	$0E$	30	-18.0	86.3	39.4	9.1	24.1	2.4
	$10E$	30	-16.2	78.6	40.6	-9.1	24.0	0.7
	$25E$	60	-16.7	63.9	31.0	-26.4	23.7	0.4

Table 5.6: Best dynamic solutions obtained by the MPA starting with different initial routing plan strategies ($St = \{HM,\ 0E,\ 10E,\ 25E\}$). Each row shows the average results over the 10 instances in the correspondent class.

accommodate detours is only available at a few locations visited by the route. Moreover, announced (pickup) locations are inserted more frequently at routes in the pool for $W > 0$, whereas with $W = 0$ consensus is reached less often for routes in the pool, allowing for more opportunities for insertions on already dispatched routes.

For a company servicing a larger number of customers, thus experiencing a higher rate of information arrival (e.g., pickup announcements), and dispatching more vehicles, these cases can be better explored. In this case, pickup announcements and active dispatched routes will overlap more often. Moreover, we do not consider diversion. In other words, after a vehicle has finished servicing all assigned customers and is returning to the depot, any pickup location announced during this time, even if feasible to be visited by the vehicle, will not be considered. An approach could be to allow for diversion only once a vehicle is on its way to the depot, that is, once the driver is informed to return (after servicing all assigned customers) he/she could receive, later, information regarding new pickup visits that can be satisfied by the vehicle.

In Table 5.7, we report results showing the effects of lead-time announcement for customer locations for the situation with 25% of pickup requests. Similarly for results without pickup requests, having customers announcing in advance their next location provides a slight improvement on the costs and required fleet. We can also observe an increase in the number of pickups serviced by en-route vehicles. Consensus is reached on routes with relative more available time before dispatch (W), and consequently more slack for the detour is required to service announced locations (pickup) after the dis-

	Dynamic Solutions, $\gamma > 0$ and 25% pickup requests							
Inst.	St	W	Δ_o(%)	Roam.(%)	Routes(%)	Fleet(%)	Range	Picks
C_{60}	HM	60	-19.4	72.3	27.0	-28.1	25.4	1.3
	$0E$	15	-27.3	87.7	21.6	-26.3	23.9	0.4
	$10E$	60	-26.2	79.5	25.7	-28.1	21.7	0.5
	$25E$	15	-20.4	63.5	12.2	-36.8	24.0	1.6
C_{90}	HM	60	-25.6	73.8	39.4	-21.3	25.7	2.6
	$0E$	15	-31.8	87.1	35.4	-6.7	23.4	1.4
	$10E$	60	-23.9	79.1	58.6	-6.7	23.2	0.6
	$25E$	30	-20.1	63.6	55.6	-14.7	24.5	1.1
C_{120}	HM	15	-14.3	67.3	35.8	-16.0	27.4	4.8
	$0E$	60	-24.9	88.3	24.1	-7.0	23.9	1.1
	$10E$	15	-19.6	78.3	32.8	-11.0	24.3	1.6
	$25E$	15	-16.0	64.4	37.2	-23.0	24.8	1.1
C_{150}	HM	60	-12.0	69.8	40.0	-17.3	24.8	2.7
	$0E$	30	-21.0	87.9	38.7	-0.9	23.5	2.9
	$10E$	15	-16.1	78.2	45.2	-4.5	23.2	3.1
	$25E$	15	-19.6	63.6	28.4	-20.9	23.2	1.3

Table 5.7: Best dynamic solutions obtained by the MPA starting with different initial routing plan strategies ($St = \{HM,\ 0E,\ 10E,\ 25E\}$). Each row shows the average results over the 10 instances in the correspondent class.

patch.

5.5. CONCLUSIONS

In this work, we introduced a new variant of the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL). In this new variant, customers' itineraries are not known to the service provider during the planning phase but rather announced by the customers dynamically, throughout the planning horizon (day). Contrary to other roaming models proposed, in the setting considered in our work customers are not required to announce their full itineraries in advance (e.g., during posing the request). This might provide customers with a more convenient and flexible experience, as they do not need to know or share their full itinerary in advance.

We proposed a Multiple Plan Approach (MPA) in which the possibility of visiting a customer in more than one location is accounted for in the solution pool. Dispatching decisions are taken by a consensus function selecting the most promising routes in the pool, trying to identify good opportunities for consolidating customers when their current locations (e.g., closer to the depot) may improve over visiting them at home. In our experiments, we observed only-home solutions with increased costs up to 30% compared to servicing customers at dynamically announced locations. The plans obtained with the MPA tend to use more routes, but requiring a similar, or fewer, number of vehicles than the only-home plans. Routes visiting the dynamically announced locations tend to be shorter (lower average range) than the only-home routes.

The MPA solutions combine servicing customers at home, using known information at the beginning of the service day, and at dynamically announced locations. Thus, an important aspect is how to devise the initial routing plans (with only home locations), such that integrating newly announced locations also brings benefits to the logistics company e.g., reducing routing costs. We have tested different strategies for obtaining initial routing plans. In particular, starting with plans servicing customers during their late windows at home provided the best improvements over the only home solutions regarding routing costs. On one hand, by postponing service at home to the latest time, it is possible to integrate a larger number of locations announced dynamically by customers. On the other hand, if customers do not announce other locations, opportunities for servicing customers during their early home windows could be missed. While the improvements obtained with starting with some customers serviced at home in their early time windows are smaller than starting with late service at home, the former tend to require a smaller vehicle fleet. Ideally, that decision should be supported by information on customers behaviour and willingness to share his/her locations. Moreover, similarly to the initial plans servicing customers only late at home, plans starting with servicing customer early at home can potentially be improved by integrating dynamically announced locations if stochastic information on customers is available. For example, when dispatching a route with early customers at home, the company might also include packages of customers it expects to announce a location while the vehicle is en-route.

We also observed benefits from considering the integration of returning services (pickup). Since customers requiring that type of service can be visited by en-route vehicles, the system has more flexibility on (re)assigning past decisions. However, there is a trade-off between that flexibility and waiting policies e.g., the longer the wait for gathering more information, the shorter the flexibility on dispatched routes to include pickup requests. Another possibility is customers informing their locations earlier (e.g., before being actually there) to service providers as a means to leverage decision making. In our experiments, the settings with positive lead-time announcements provided, on average, improvements of up to 3%. One possible disadvantage of such setting is that this could lead to uncertainty in the informed data (e.g., due to travel time changes) and possibly decrease the value of anticipating the location to the service provider.

Additionally, in this work we did not assume any stochastic information regarding customers' itineraries. By making use of advances in information technologies, predicting the daily itinerary of customers could be a possibility. This would allow, for example, the use of a sampling mechanism for possible customer locations. The MPA proposed in this work could then be extended to the Multiple Scenario Approach (MSA) (Bent and Van Hentenryck, 2004). It is a generalization of the MPA considering potential future information (e.g., a when and where a customer will be available in the future) and can significantly improve over the MPA.

6

CONCLUSIONS

We've all been up through the night time baby
Now let's read the rays of reality

Jimi Hendrix, Power of Soul

We conclude the thesis by summarizing the main results and findings presented in each previous chapter. We also highlight some important limitations that will need to be resolved before a full realization of the concepts discussed in the thesis can be implemented in a real-life system. Finally, we present points of interest for further research and possible directions to extend the work in this thesis. We hope to stimulate the community to further consider the problems presented and their applicability.

6.1. GENERAL DISCUSSION

Transportation activities play an important role in modern societies and are in continuous change. Not only because transport of freight and people is a driving factor for promoting economic development, but also due to the challenges to satisfy increasing service levels and customers expectations.

One of the main drivers of changes in the transport sector has been on-line shopping. As e-commerce develops and grows, e-shoppers and, consequently, e-retailers are demanding more efficient, reliable and cheap delivery services from logistics service providers. To compete against the immediate gratification provided by brick-and-mortar stores and to attract more customers, e-retailers now have a wide variety of options to choose from in deciding how to get the parcel to its buyer such as multiple carriers, home delivery at preferred times, and the use of alternative collection points like automatic parcel stations. However, taking the parcel from the warehouse's shelf to the customer's hands, the last-mile conundrum, is not without its challenges. While carrying out e-commerce allows for significant cost reductions, specially due to reduced need for physical establishment, order placement, staffing and customer support, shipping costs can increase the final cost of products purchased on-line (Lee, 2002). Moreover, most of the world's population lives in urban areas today and, as a consequence, cities with a high population density are posing new challenges to transportation services within urban areas. Congestion, for example, is a major issue in medium and big cities, hindering the movements of both freight and people. Transport of merchandise and passenger commuters are major contributors to greenhouse gas emissions. Logistic service providers now have to, on one hand, satisfy customer demands for fast, reliable and cheap services and, on the other hand, face the hurdles of operating in such highly populated regions while complying with strict environmental policies.

Moreover, facing the pandemic crisis during 2020 accelerated the expansion of e-commerce – and its importance for the continued access of products to consumers during strict confinement measures – but has also heightened many challenges that already existed before COVID-19. On top of that, persistent inequalities among the population have been brought forward during the the crisis. As not everyone has the same level of digital accessibility, buying online might not be an option to everyone, especially the most vulnerable. During the pandemic, e-commerce proved crucial for the continued access to not only high tech goods, but also to everyday necessities, like groceries and medicines, relevant to most of the population. Regulations aimed at enabling innovative environments that foster innovative e-commerce solutions have to ensure that it can reach and deliver to everyone.

This thesis consists of two parts and investigates innovative models aimed at providing more cost-efficient solutions to last-mile transportation within urban contexts. In the first part, the research focuses on crowd-based logistic services. We presented examples of crowd-based logistic applications and introduced a system in which transportation requests are performed by crowd-sourced drivers. We consider the use of transfer locations in the transportation network through which drivers are able to exchange requests as a means to better manage the available pool of drivers. In the second part, the aim of research was to evaluate roaming delivery systems in which direct-to-consumer orders can be fulfilled using the trunk of a customer's car. In order to capture more realistic aspects of the operational scenarios faced by the operators, we extend the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL), recently introduced to model roaming systems, in two ways. First, we solve the problem considering stochastic travel times and, second, we introduce a new dynamic variant of the VRPRDL in which customers announce their itinerary locations as time goes by.

6.2. Research Questions Revisited and Results

The main objective of the thesis is to quantify possible costs savings in last-mile systems using the aforementioned models. These include reductions in total fuel usage, less overtime for drivers and required capacity (number of vehicles/drivers) to satisfy demand. Thus, when assessing the proposed methodologies and solutions, we conduct the experimental discussion mainly in terms of decrease in travel time or kilometers driven, and fleet size. For urban freight transportation, reduced total kilometers driven have far-reaching consequences, as it implies not only reduced travelled distance and costs for operators, but also less negative social and environmental impacts (Browne et al., 2012).

What are the characterizing features of crowd-logistics, specially crowd-shipping?

In Chapter 2, we provided an overview of crowd-based solutions for logistics and transportation activities. *Crowd-Logistics* have the potential to transform the industry but, until now, it is not entirely clear how to proper define the participants constituting the crowd and how to integrate the services offered in crowd platforms into existing logistics systems. In particular, for the crowd-sourcing of (last-mile) transportation activities, *crowd-shipping*, two realizations have been considered. In the first, crowd-drivers are employed for transportation activities (e.g., pickup/delivery of parcels) while already performing another duty (e.g., driving from work to home). In the second realization, individuals offer their vehicle and time to perform the activities, agreeing to work for a certain period of time and for a minimum payment assurance e.g., based on number of worked hours or deliveries performed. In the first model, logistic activities might be performed in a more efficient manner, by using pre-existent movements and consolidating existing flows. However, service providers have less flexibility in using crowd-drivers during the trips they would perform anyway, as their availability and willingness to perform certain requests are limited (e.g., requiring a long detour or with a low compensation). Crowd-sourced drivers that agree to work for a certain period of time can be managed in a more flexible way, but fulfillment of requests is realized by creating new service flows rather than exploiting existing ones. In any case, an important issue that companies need to assess when integrating crowd-sourced services into their operational chain is to which extent rely on the crowd. Since drivers join the crowd on a voluntarily basis, service providers will most likely always have to rely on professional capacity to ensure customers fulfillment at desired service levels.

HOW CAN AVAILABLE CROWD-SOURCED CAPACITY (DRIVERS) BE MANAGED AND USED IN ORDER TO EFFECTIVELY MEET TRANSPORTATION DEMAND?

When willing to allocate crowd-drivers to perform transportation activities, Chapter 2 emphasized the need for operators to employ appropriate compensation mechanisms in order to attract enough individuals into the crowd platform. As important is how to employ the available capacity to satisfy demand as effectively as possible i.e., fulfilling as many requests as possible without the need of adjusting compensation (e.g., hourly wages). In Chapter 3, we evaluate a model in which individuals commit to work for a specific period of time, a block, in return for a minimum pay assurance during that time. Large e-tailers (e.g., Amazon Flex) and meal delivery platforms (e.g., Grubhub) already employ such models. In order to provide more flexibility to operators and to crowd drivers as well, blocks comprise short periods of time. We assess the benefits of employing the possibility of transferring requests between drivers as a means to make better use of the available crowd-based drivers to handle requests with different characteristics (e.g., short and long distance tasks). Our experiments showed that a decrease of up to 47% on total travelled distance and using near half the number of required drivers to service all requests can be achieved on settings where transfers can be used compared to settings without transfers. The benefits of using transfers reduces as driver shift length increases since routes can cover larger distances and serve more requests in the same route, which tends to be less costly than when using transfers. The experiments also showed that transfers allow for more requests to be served with a given capacity when drivers operate in a block-based system. On average, 22% more requests could be satisfied by crowd-sourced drivers when transfers are utilized.

CAN TRUNK DELIVERY BE AN OPTION TO MITIGATE FAILED DELIVERIES DUE TO UNCERTAINTY IN SERVICING CUSTOMERS AT PLANNED LOCATIONS IN A PRIORI DESIGNED DELIVERY ROUTES?

In the second part of the thesis, we focus on roaming delivery systems, in particular, when a customer's car is used to facilitate the delivery process of direct-to-consumer orders. Such systems have been modeled as a Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL), but only considering fully deterministic scenarios. In order to account for more real aspects, in Chapter 4 we consider the VRPRDL under stochastic travel times. Due to uncertain disruptions during the day e.g., accidents, vehicle breakdowns, weather, travel times might hinder the service provider of fulfilling requests within promised time windows at a given location, and service at another location and time windows must be planned. Thus, we model the problem as a two-stage stochastic problem and propose different recourse actions, using the possibility of servicing customers at different locations at different times, reducing the number of failed deliveries caused by travel time uncertainty. The benefits of the stochastic solutions over only deterministic ones, considering expected travel time values, are twofold. First, evaluating the a priori plan over different travel time scenarios allows for solutions hedging better against travel time uncertainty, leading to savings of, on average, 30%. Second, recourse policies exploiting the customer itinerary structure leads to more robust plans, on average, with a cost reduction of 25%.

IS TRUNK DELIVERY STILL EFFECTIVE WHEN INFORMATION REGARDING CUSTOMERS' ITINERARIES ARE UNKNOWN BEFOREHAND?

In Chapter 4, we assumed that customers inform their full itinerary for the delivery day to the service provider. In Chapter 5, we relax this assumption and rather consider that customers might announce alternative locations to home where their car is or will be parked, throughout the day, in real-time. The service provider might then decide to dispatch vehicles for servicing customers either at home or at a dynamically announced location. In the experiments conducted, comparing a solution visiting customers only at home and a solution obtained by integrating the dynamically announced customer locations to define the routing plans, improvements of up to 11% could be observed on some instances compared to only-home solutions. While the number of dispatched routes is slightly higher in the dynamic environment, the required vehicle fleet is not significantly different. Within the dynamic environment, vehicles tend to perform shorter routes, servicing customers at locations closer to the depot, on average, than home locations in the only-home solutions. Vehicles (and consequently drivers), are better utilized throughout the day and are able to finish servicing all customers earlier than compared to the only-home solution (since routes servicing customers at home next to the end of the time horizon finish later). Integrating pickup flows (e.g., returning of merchandising) might lead to a reduction of 5% on the total time required to service all customers' requests.

6

6.3. CHALLENGES AND LIMITATIONS

Notwithstanding the promising findings of this thesis, a number of simplifying assumptions were made when modelling the systems discussed, so that appropriate methodologies to achieve insightful solutions with reasonable computational effort could be applied. In Chapter 3, we assumed that the (strategic) decisions on the transfers infrastructure are already in place when solving the (operational) routing plans for the crowd-drivers. Such decisions might not necessarily require the establishment of new facilities, for example, by using existent available spaces such as the parking lots available in shopping areas, gas stations or public transportation hubs. However, some additional measures and considerations will be required in order to make transfer operations as efficient and secure as possible. The solutions using transfers in Chapter 3 require the synchronization of drivers at these locations, either by having drivers waiting (shortly) for other drivers or by allowing parcels to be (shortly) stored at the location. Clearly, such issues have to be accounted when assessing the cost-benefits of the proposed transportation systems.

Additionally, as highlighted in Chapter 2, the use of crowd-sourced drivers raises concerns regarding security, reliability and accountability. Crowd platforms have to gain the confidence of both e-tailers and e-shoppers, creating willingness to rely on services provided by the crowd e.g., the parcel will be delivered timely and with privacy. An issue that was not addressed in this study was whether crowd-drivers always fulfill the directions given by the platform e.g., decline a delivery service after having accepted it. Whereas such cases can be mitigated by the platform via trust generation mechanisms, it could require additional measures by the platform, potentially increasing the final cost of transportation and decreasing the value of crowd-services. However, hidden beneath promises of higher levels of flexibility and autonomy, individuals working for crowd platforms might often experience low pay, social isolation, working at irregular hours, overwork, sleep deprivation and exhaustion (Wood et al., 2018). While the commercialization of spare time is one of the central ideas behind the gig-economy, it has become, however, a way for many to avoid unemployment, resulting in effectively full-time employees no longer protected by a legal system. Greater efforts are needed to ensure basic rights and conditions for workers in the gig-economy, lest the benefits of more flexible, cost-efficient business models are reaped only by companies in detriment of workers, who are also consumers.

Regarding the roaming systems considered in Chapters 4 and 5, some questions still remain to be answered. Perhaps one of the most important ones concerns the scalability of the methodologies proposed in the thesis to tackle the problems. As mentioned, the volume of direct-to-consumer orders are increasing at a staggering pace, following the growth of the e-commerce market. Operators have to deal with thousands of orders per day, managing and coordinating a fleet of thousands of vehicles. In the thesis, we propose heuristics methods able to cope with problem instances having a few hundred customers, beyond what is usually handled, within reasonable computational times, by exact methods. However, it would be worthwhile, from a practical point of view, assessing such limitations on real-life scenarios, not only taking into account more customers but also other aspects that increase the complexity of solving the problems.

Furthermore, security and privacy are also issues that need to be accounted for in such systems. The automotive industry has partnered with e-tailers and delivery operators to provide secure technologies that allows remote control of the vehicle's trunk. Regardless, data leaks and breaches are still a concern for some, which could allow for unauthorized access. Moreover, the convenience of having the parcels going to the consumer rather than the other way around, comes with a trade-off in privacy. Knowing when and where to deliver the parcel to the costumer implies that information regarding the geo-position of the car is constantly monitored as well as the daily routines of the costumer. These are, for instance, features of a pilot trunk delivery service in Belgium (`cardrops.com`).

6.4. FURTHER RESEARCH

The work presented in this thesis can be extended in different ways. One of the main limitations of the approach presented in Chapter 3 is that it assumes all information regarding customer orders and, more importantly, crowd-drivers are available at the start of the planning horizon. In reality, crowd drivers join the platform in a dynamic way, and crowd capacity becomes available over time during the planning horizon.

Given the uncertain nature of crowd drivers, an interesting direction for further investigation could be to consider an *a priori* approach for the problem presented in Chapter 3. If stochastic information regarding how drivers join the platform is available, one can design an initial plan servicing customers based on (expected) crowd capacity. Similarly to the approach presented in Chapter 4, recourse policies using professional drivers could be defined for meeting demand when crowd capacity realizes but is not enough following the *a priori* plan i.e., consider a *static and stochastic* problem.

From a methodological point of view, the Sample Average Approximation (SAA) method proposed in Chapter 4 can be improved by adapting the scenario sampling mechanism, such that in each SAA replication a better lower bound (estimation) can be computed by selecting appropriate scenarios. This could be achieved, for example, using advances in Machine Learning technology used to identify sample elements that yield a good representation of the full sampling set. Moreover, solving each SAA replication can be executed, to a certain extend, using parallel strategies to speed up the method.

Future research could involve the role of new technology to monitor and predict daily itineraries of receivers and drivers. Some works have attempted to understand the mechanisms of a traveler's behaviour through space an time. For example, Hamed and Mannering (1993) proposed a methodology for modeling traveler's post-work activities, but lacked the use of more elaborate and detailed data, specifically, regarding home-stay duration, network and congestion information. Nowadays, the ubiquity of smart phones and the internet, coupled with predictive analytics, can certainly fulfill that gap.

BIBLIOGRAPHY

Accenture Interactive (2015). The next generation of commerce. Accenture. `https://tinyurl.com/y7wnk87e`. Accessed August-2017.

Adulyasak, Y. and Jaillet, P. (2016). Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science*, 50(2):608–626.

Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.

Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295 – 303.

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

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

Arvidsson, N., Givoni, M., and Woxenius, J. (2016). Exploring last mile synergies in passenger and freight transport. *Built Environment*, 42(4):523–538.

Authority for Consumers & Markets (2019). Like last year, the growth on the parcel delivery market offsets decrease on the postal market. Authority for Consumers & Markets. `https://tinyurl.com/tjlbk37`. Accessed December-2019.

Baird, N. and Rosenblum, P. (2015). Home delivery: Retailers' brave new world. Retail Systems Research. `https://tinyurl.com/t8gjwgm`. Accessed September-2019.

Bandler, J., Callahan, P., Burke, D., Bensinger, K., and O'Donovan, C. (2019). Inside documents show how Amazon chose speed over safety in building its delivery network. ProPublica. `https://tinyurl.com/slnk2uq`. Accessed December-2019.

Barr, A. and Wohl, J. (2013). Exclusive: Wal-mart may get customers to deliver packages to online buyers. Reuters. `https://tinyurl.com/yccje985`. Accessed October-2017.

Bates, J. and Leibling, D. (2012). Spaced out: Perspectives on parking policy. RAC Foundation. `https://tinyurl.com/7vn8gcw`. Accessed 20-January-2019.

Behrend, M. and Meisel, F. (2018). The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? *Transportation Research Part B: Methodological*, 111:227 – 243.

Behroozi, M. and Ma, D. (2020). Crowdsourced Delivery with Drones in Last Mile Logistics. In Huisman, D. and Zaroliagis, C. D., editors, *20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020)*, volume 85 of *OpenAccess Series in Informatics (OASIcs)*, pages 17:1–17:12, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Bektaş, T., Erdoğan, G., and Røpke, S. (2011). Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, 45(3):299–316.

Bektas, T., Repoussis, P. P., and Tarantilis, C. D. (2014a). Chapter 11: Dynamic vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications*, chapter 11, pages 299–347.

Bektas, T., Repoussis, P. P., and Tarantilis, C. D. (2014b). Dynamic vehicle routing problems. In *Vehicle Routing*, pages 299–347.

Bektaș, T., Crainic, T. G., and Van Woensel, T. (2015). From Managing Urban Freight to Smart City Logistics Networks. (Research Papers 2015, CIRRELT-2015-17), Montreal, CA CIRRELT.

Bent, R. W. and Van Hentenryck, P. (2004). Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31.

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

Biesinger, B., Hu, B., and Raidl, G. (2016). An integer L-shaped method for the generalized vehicle routing problem with stochastic demands. *Electronic Notes in Discrete Mathematics*, 52:245 – 252. INOC 2015 – 7th International Network Optimization Conference.

Binart, S., Dejax, P., Gendreau, M., and Semet, F. (2016). A 2-stage method for a field service routing problem with stochastic travel and service times. *Computers & Operations Research*, 65:64 – 75.

Branda, M. (2012). Sample approximation technique for mixed-integer stochastic programming problems with several chance constraints. *Operations Research Letters*, 40(3):207 – 211.

Browne, M., Allen, J., Nemoto, T., Patier, D., and Visser, J. (2012). Reducing social and environmental impacts of urban freight transport: A review of some major cities. *Procedia - Social and Behavioral Sciences*, 39:19 – 33.

Buldeo Rai, H., Verlinde, S., Merckx, J., and Macharis, C. (2017). Crowd logistics: an opportunity for more sustainable urban freight transport? *European Transport Research Review*, 9(3):39.

Campbell, A. M. and Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science*, 38(3):369–378.

Cattaruzza, D., Absi, N., Feillet, D., and González-Feliu, J. (2015). Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, pages 1–29.

Chen, C. and Pan, S. (2016). *Using the Crowd of Taxis to Last Mile Delivery in E-Commerce: a methodological research*, pages 61–70. Springer International Publishing, Cham.

Chen, C., Pan, S., Wang, Z., and Zhong, R. Y. (2017a). Using taxis to collect citywide e-commerce reverse flows: a crowdsourcing solution. *International Journal of Production Research*, 55(7):1833–1844.

Chen, W., Mes, M., and Schutten, M. (2017b). Multi-hop driver-parcel matching problem with time windows. *Flexible Services and Manufacturing Journal*.

Cohen, R. (2019). Pakketbezorging komt straks tot in de kofferbak. Het Financieele Dagblad. `https://tinyurl.com/yx3dr2xw`. Accessed September-2019.

Cordeau, J.-F. and Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46.

Cortés, C. E., Matamala, M., and Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711 – 724.

Crainic, T. G. and Montreuil, B. (2016). Physical internet enabled hyperconnected city logistics. *Transportation Research Procedia*, 12(Supplement C):383 – 398. Tenth International Conference on City Logistics 17-19 June 2015, Tenerife, Spain.

Dablanc, L., Morganti, E., Arvidsson, N., Woxenius, J., Browne, M., and Saidi, N. (2017). The rise of on-demand 'instant deliveries' in european cities. *Supply Chain Forum: An International Journal*, 18(4):203–217.

Dahle, L., Andersson, H., Christiansen, M., and Speranza, M. G. (2019). The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122 – 133.

Dayarian, I. and Savelsbergh, M. (2017). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Optimization Online 2017-07-6142*.

de Brito, M. P. and Dekker, R. (2004). *A Framework for Reverse Logistics*, pages 3–27. Springer Berlin Heidelberg, Berlin, Heidelberg.

DePillis, L. (2019). America's addiction to absurdly fast shipping has a hidden cost. CNN Business. `https://amp.cnn.com/cnn/2019/07/15/business/fast-shipping-environmental-impact/index.html`. Accessed September-2019.

DHL (2015). Postbus to be a parcel courier. DHL Press Release. `https://tinyurl.com/y9w2aury`. Accessed October-2017.

DHL Trend Research (2017). Sharing economy logistics: Rethinking logistics with access over ownership. DHL Trend Research, `https://tinyurl.com/y7z6w628`. Accessed July-2017.

Ehmke, J. F., Campbell, A. M., and Urban, T. L. (2015). Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, 240(2):539 – 550.

eMarketer Editors (2017). A brief overview of the global e-commerce market. `https://tinyurl.com/y7txkfkk`. Accessed October-2017.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. (2016). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research*, 249(1):55 – 66.

Estellés-Arolas, E. and González-Ladrón-de Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Article Journal of Information Science*, 38(2):189–200.

Fatnassi, E., Chaouachi, J., and Klibi, W. (2015). Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics. *Transportation Research Part B: Methodological*, 81(Part 2):440 – 460.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57(Supplement C):28 – 46.

Gambella, C., Naoum-Sawaya, J., and Ghaddar, B. (2018). The vehicle routing problem with floating targets: Formulation and solution approaches. *INFORMS Journal on Computing*, 30(3):554–569.

Gendreau, M., Ghiani, G., and Guerriero, E. (2015). Time-dependent routing problems: A review. *Computers & Operations Research*, 64:189 – 197.

Gendreau, M., Guertin, F., Potvin, J.-Y., and Taillard, E. (1999). Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390.

Gendreau, M., Jabali, O., and Rei, W. (2016). 50th anniversary invited article—future research directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173.

Geron, T. (2013). Airbnb and the unstoppable rise of the share economy. Forbes. `https://tinyurl.com/y9k73xlj`. Accessed October-2017.

Ghiani, G. and Improta, G. (2000). An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research*, 122(1):11 – 17.

Ghilas, V., Demir, E., and Van Woensel, T. (2016). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12 – 30.

Goetting, E. and Handover, W. N. (2016). Crowd-shipping: Is crowd-sourced the secret recipe for delivery in the future? Germany Industry and Commerce Ltd.`https://tinyurl.com/ycfuynty`. Accessed August-2017.

Guastaroba, G., Speranza, M. G., and Vigo, D. (2016). Intermediate facilities in freight transportation planning: A survey. *Transportation Science*, 50(3):763–789.

Hamari, J., Sjöklint, M., and Ukkonen, A. (2016). The sharing economy: Why people participate in collaborative consumption. *Journal of the Association for Information Science and Technology*, 67(9):2047–2059.

Hamed, M. M. and Mannering, F. L. (1993). Modeling travelers' postwork activity involvement: Toward a new methodology. *Transportation Science*, 27(4):381–394.

Hawkins, A. (2018). Amazon will now deliver packages to the trunk of your car. Verge. `https://tinyurl.com/yaeektvd`. Accessed 20-January-2019.

Howe, J. (2006). The rise of crowdsourcing. Wired. `https://www.wired.com/2006/06/crowds/`. Accessed September-2017.

Hu, M., editor (2019). *Sharing Economy: Making Supply Meet Demand*. Springer International Publishing, 1 edition.

Hvattum, L. M., Løkketangen, A., and Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40(4):421–438.

Jabali, O., Leus, R., Van Woensel, T., and de Kok, T. (2015). Self-imposed time windows in vehicle routing problems. *OR Spectrum*, 37(2):331–352.

Jesus Gonzalez-Feliu, F. S. and Routhier, J.-L., editors (2014). *Sustainable Urban Logistics: Concepts, Methods and Information Systems*. Springer, Berlin, Heidelberg.

Kafle, N., Zou, B., and Lin, J. (2017). Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. *Transportation Research Part B: Methodological*, 99(Supplement C):62 – 82.

Kamar, E. and Horvitz, E. (2009). Collaboration and shared plans in the open world: Studies of ridesharing. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, IJCAI'09, pages 187–194, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Klapp, M. A., Erera, A. L., and Toriello, A. (2018). The one-dimensional dynamic dispatch waves problem. *Transportation Science*, 52(2):402–415.

Kleywegt, A., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

Klumpp, M. (2017). *Crowdsourcing in Logistics: An Evaluation Scheme*, pages 401–411. Springer International Publishing, Cham.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2015). The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816.

Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133 – 142.

Laporte, G., Louveaux, F. V., and van Hamme, L. (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423.

Larsen, R. and Pranzo, M. (2018). A framework for dynamic rescheduling problems. *International Journal of Production Research*, 57(1):16–33.

Lee, K. R. (2002). Impacts of information technology on society in the new century. *Business and management*, 5(6):46–55.

Li, B., Krushinsky, D., Reijers, H. A., and Woensel, T. V. (2014). The share-a-ride problem: People and parcels sharing taxis. *European Journal of Operational Research*, 238(1):31 – 40.

Li, X., Tian, P., and Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, 125(1):137 – 145.

Lipsman, A. (2019). eMarketer. `https://www.emarketer.com/content/global-ecommerce-2019`. Accessed September-2019.

Lombard, A., Tamayo-Giraldo, S., and Fontane, F. (2018). Vehicle routing problem with roaming delivery locations and stochastic travel times (VRPRDL-S). *Transportation Research Procedia*, 30:167 – 177. EURO Mini Conference on Advances in Freight Transportation and Logistics.

Macharis, C. and Kin, B. (2017). The 4 a's of sustainable city distribution: Innovative solutions and challenges ahead. *International Journal of Sustainable Transportation*, 11(2):59–71.

Maknoon, Y. and Laporte, G. (2017). Vehicle routing with cross-dock selection. *Computers & Operations Research*, 77:254 – 266.

Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.

Masson, R., Lehuédé, F., and Péton, O. (2014). The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12 – 23.

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., and Tlahig, H. (2017). Optimization of a city logistics transportation system with mixed passengers and goods. *EURO Journal on Transportation and Logistics*, 6(1):81–109.

McCrea, B. (2016). From dc to final destination: Last mile dilemma. Logistics Management. `https://www.logisticsmgmt.com/article/from_dc_to_final_destination_last_mile_dilemma`. Accessed September-2019.

Miranda, D. M. and Conceição, S. V. (2016). The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, 64:104 – 116.

Mitrović, S. and Laporte, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR: Information Systems and Operational Research*, 44(3):217–227.

Mladenow, A., Bauer, C., and Strauss, C. (2015). Crowdsourcing in logistics: Concepts and applications using the social crowd. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '15, pages 30:1–30:8, New York, NY, USA. ACM.

Moccia, L., Cordeau, J.-F., and Laporte, G. (2012). An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. *Journal of the Operational Research Society*, 63(2):232–244.

Montreuil, B. (2011). Toward a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2):71–87.

Nakao, Y. and Nagamochi, H. (2008). Worst case analysis for pickup and delivery problems with transfer. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E91-A(9):2328–2334.

OECD (2020). E-commerce in the time of COVID-19. `https://tinyurl.com/yynvx59j`. Accessed December-2020.

Oxford Economics (2017). Future trends and market opportunities in the world's largest 750 cities, Executive Summary. `https://tinyurl.com/y8u5rcra`. Accessed October-2017.

Ozbaygin, G., Karasan, O. E., Savelsbergh, M., and Yaman, H. (2017). A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 100:115 – 137.

Ozbaygin, G. and Savelsbergh, M. (2019). An iterative re-optimization framework for the dynamic vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 128:207 – 235.

Paloheimo, H., Lettenmeier, M., and Waris, H. (2016). Transport reduction by crowdsourced deliveries – a library case in Finland. *Journal of Cleaner Production*, 132:240 – 251.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013a). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1 – 11.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013b). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1 – 11.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403 – 2435.

Punakivi, M., Yrjölä, H., and Holmström, J. (2001). Solving the last mile issue: reception box or delivery box?

Punel, A. and Stathopoulos, A. (2017). Modeling the acceptability of crowdsourced goods deliveries: Role of context and experience effects. *Transportation Research Part E: Logistics and Transportation Review*, 105:18–38.

Qu, Y. and Bard, J. F. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439 – 2456.

Rai, H. B. (2019). *Environmental sustainability of the last mile in omnichannel retail*. PhD thesis, Vrije Universiteit Brussel.

Rais, A., Alvelos, F., and Carvalho, M. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530 – 539.

Reagan, C. (2016). A $260 billion 'ticking time bomb': The costly business of retail returns. CNBC. `https://tinyurl.com/ybltp95b`. Accessed October-2017.

Reyes, D., Savelsbergh, M., and Toriello, A. (2017). Vehicle routing with roaming delivery locations. *Transportation Research Part C: Emerging Technologies*, 80:71 – 91.

Rodrigue, J.-P. (2017). Residential parcel deliveries: Evidence from a large apartment complex. MetroFreight Volvo Center of Excellence. `https://tinyurl.com/ybc929qa`. Accessed October-2017.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Russell, R. A. and Urban, T. L. (2008). Vehicle routing with soft time windows and erlang travel times. *Journal of the Operational Research Society*, 59(9):1220–1228.

Sampaio, A., Savelsbergh, M., Veelenturf, L., and Woensel, T. V. (2018). Crowd-based city logistics. In Faulin, J., Grasman, S. E., Juan, A. A., and Hirisch, P., editors, *Sustainable Transportation and Smart Logistics: Decision-Making Models and Solutions*, chapter 15, pages 382–398. Joe Hayton.

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. *Networks*, 76(2):232–255.

Savelsbergh, M. and Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590.

Savelsbergh, M. W. P. and Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29(1):17–29.

Schultz, R., Stougie, L., and van der Vlerk, M. H. (1998). Solving stochastic programs with integer recourse by enumeration: A framework using gröbner basis. *Mathematical Programming*, 83(1):229–252.

Song, J. and Regan, A. C. (2003). An auction based collaborative carrier network. Technical report, University of California Transportation Center.

Stenger, A., Vigo, D., Enz, S., and Schwind, M. (2013). An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47(1):64–80.

Taillard, E. D., Gambardella, L. M., Gendreau, M., and Potvin, J.-Y. (2001). Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135(1):1 – 16.

Tang, C. S., Bai, J., So, K. C., Chen, X. M., and Wang, H. (2016). Coordinating Supply and Demand on an On-Demand Platform: Price, Wage, and Payout Ratio. Available at SSRN: `https://ssrn.com/abstract=2831794`. Accessed September-2017.

Taylor, T. (2017). On-demand service platforms. Available at SSRN: `http://dx.doi.org/10.2139/ssrn.2722308`. Accessed September-2017.

Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1):214 – 224.

Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. (2014a). The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies*, 48:66 – 83.

Taş, D., Gendreau, M., Dellaert, N., van Woensel, T., and de Kok, A. (2014b). Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, 236(3):789 – 799.

Tom Tom (2016). Tomtom traffic index: Measuring congestion worldwide. Tom Tom Traffic Index. `https://tinyurl.com/y6wgf8e6`. Accessed 20-January-2019.

Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Number 18 in MOS-SIAM Series on Optimization. SIAM.

Trentini, A., Masson, R., Lehuédé, F., Malhéné, N., Péton, O., and Tlahig, H. (2012). A shared " passengers & goods " city logistics system. In *4th International Conference on Information Systems, Logistics and Supply Chain*, page 10p, Quebec, Canada.

Turkeš, R., Sörensen, K., and Hvattum, L. M. (2020). Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search. *European Journal of Operational Research*.

United Nations (2015). World urbanization prospects: The 2014 revision. Department of Economic and Social Affairs, Population Division. `https://tinyurl.com/yd7mfv5z`. Accessed September-2017.

United Parcel Service of America (2015). UPS pulse of the online shopper: A customer experience study. comScore. `https://tinyurl.com/h3kp8hr`. Accessed October-2017.

Van Duin, R., Wiegmans, B., Tavasszy, L., Hendriks, B., and He, Y. (2019). Evaluating new participative city logistics concepts: The case of cargo hitching. *Transportation Research Procedia*, 39:565 – 575. 3rd International Conference Green Cities – Green Logistics for Greener Cities, Szczecin, 13-14 September 2018.

Vareias, A. D., Repoussis, P. P., and Tarantilis, C. D. (2017). Assessing customer service reliability in route planning with self-imposed time windows and stochastic travel times. *Transportation Science*, Articles in Advance(on-line):1–26.

Verdonck, L., Caris, A., Ramaekers, K., and Janssens, G. K. (2013). Collaborative logistics from the perspective of road transportation companies. *Transport Reviews*, 33(6):700–719.

Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A. (2003). The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2):289–333.

Voccia, S. A., Campbell, A. M., and Thomas, B. W. (2019). The same-day delivery problem for online purchases. *Transportation Science*, 53(1):167–184.

Wall, T. (2019). Driven to the edge: life on the christmas parcel delivery run. The Guardian. `https://tinyurl.com/qrodgvdc`. Accessed December-2019.

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. *Transportation Research Part E: Logistics and Transportation Review*, 93(Supplement C):279 – 293.

Wood, A., Graham, M., Lehdonvirta, V., and Hjorth, I. (2018). Good gig, bad gig: Autonomy and algorithmic control in the global gig economy. *Employment and Society*, 1(33):56–75.

A

PDPTW-T instances from literature

We provide a comparison of our proposed methodology and the local search approach proposed by Mitrović and Laporte (2006), highlighting some different aspects of solutions obtained with each method. To the best of our knowledge, Mitrović and Laporte (2006) were the first to propose instances for the pickup and delivery problem with transfers (PDP-T) in the context of freight transportation. The authors were motivated by a courier company which allows transfers of loads between vehicles as a way to keep drivers within their home area. The authors propose a generalization of that practice, where vehicles can visit customers over the entire service region. Masson et al. (2013) also consider PDP-T instances, but motivated by the transportation of disabled people. Moreover, travel times in those instances did not satisfy triangle inequality, which poses a difficulty for our ALNS. Thus, we did not consider such instances.

Table A.1 presents a comparison of the results obtained on a subset of the instances proposed by Mitrović and Laporte (2006) with 50 requests. Each line on Table A.1 is the average result over 30 instances. In particular, we consider instances where pickup and delivery locations are randomly (uniform) generated inside a $60km \times 60km$ square. Time windows for customers are generated with two schemes: in TW=10, the time for servicing each request (the time between earliest pickup and latest delivery times) is 10 hours and, in TW=2-4-8, $30\%, 50\%$ and 20% of the requests have $2, 4$ and 8 hours to be serviced, respectively. For each scheme, an instance in which no transfers are available and another (with the same customers) wherein four transfer locations are available in the network are considered. A service time of 5 minutes is assumed at each customer, whereas no service time is incurred at transfer locations. Moreover, capacity is not a binding constraint. It is assumed that vehicles start at dummy locations at distance/time zero from any other location and have a shift length of 10 hours (all starting at time 0). We observe that our approach is designed with focus on crowd-shipping applications, and such assumptions on the vehicles would hardly hold in these contexts, where drivers have limited shift lengths and are not always willing to drive over all the service region.

		Local Search by Mitrović and Laporte (2006)			Proposed ALNS				
TW	TR	c	Trs(%)	Veh	c	Trs(%)	Veh	Δ_D(%)	Δ_V(%)
10	0	845.86	0	4.63	836.35	0	3.36	−1.12	−27.42
	4	791.53	51	7.03	824.24	5.14	3.43	4.14	−51.21
2-4-8	0	1324.63	0	11.03	1291.88	0	10.47	−2.47	−5.08
	4	1297.84	28	11.90	1279.66	0	10.63	−1.40	−10.67

Table A.1: Comparison of results obtained on instances proposed by Mitrović and Laporte (2006) using the Local Search proposed by the same authors and our proposed ALNS. For each scheme of customer's time windows (TW=10, TW=2-4-8), we consider a scenario in which transfers are not available and a scenario with four transfer locations in the network. Column c reports the total (distance) cost of a solution, Trs presents the percentage of requests that are transferred between vehicles, and column Veh reports the total number of vehicles required. Columns Δ_D(%) and Δ_V(%) illustrate the relative difference in total distance and number of vehicles, respectively, between the solutions achieved by the local search of Mitrović and Laporte (2006) (s^1) and our proposed ALNS (s^2) as $\Delta = 100 \times (s^1 - s^2)/s^1$.

When solving an instance without transfers available, our ALNS uses an initial solution consisting of one dedicated vehicle per customer (servicing only the pickup and delivery operations of one customer). When the same instance is solved considering the use of transfers, we provide the best solution found without transfer as initial solution to our ALNS. Results in Table A.1 are the average values over 30 instances in each TW scheme.

We can observe that, in general, solutions achieved with our ALNS require less vehicles to serve all customers and tend to transfer requests less frequently than solutions achieved in Mitrović and Laporte (2006). As we focus on applications of crowd-sourced drivers, it is paramount that demand can be met using capacity efficiently, without the need of eventually having to attract more drivers to the system (e.g., transferring might reduce total distance, but requires relatively more drivers available). For instances in which customers have a 10 hours time windows (TW=10), our ALNS provide solutions with comparable total distance as the local search by Mitrović and Laporte (2006) when no transfers are available, but requiring, on average, 27% less vehicles. When transfers are available, our ALNS achieves solutions with a slightly higher total distance (4.14%), but requiring half the number of vehicles.

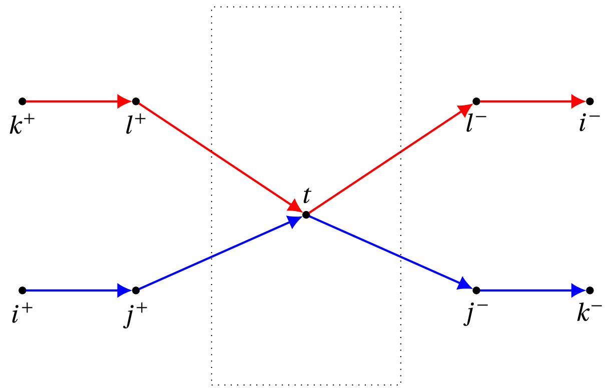
For instances with customer time windows generated in the scheme $TW = 2-4-8$, the best solutions obtained with our ALNS do not make use of transfers. However, compared to solutions in Mitrović and Laporte (2006) with and without transfers, our methodology achieves solutions with slightly lower total distance and less required vehicles. In such instances, two factors contribute for reducing the attractiveness of transferring requests: (i) customers with tight time windows, reducing the time available for consolidation of requests at the transfers locations and (ii) the shift length allows vehicles to move through the entire service area (no constraints on the initial and end locations of a vehicle). As our computational experiments show, transfers are more beneficial when drivers have a limited shift length and are not able to move over the entire service region.

B

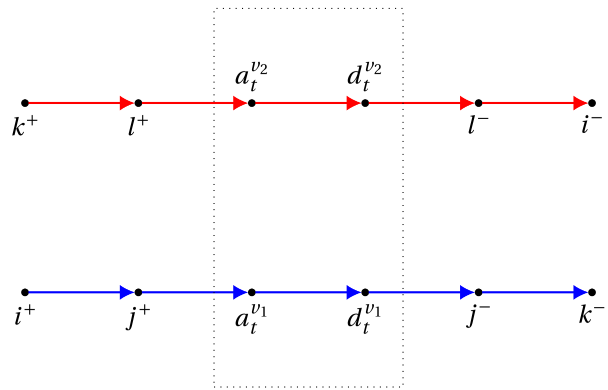
Modeling Transfer Operations

In oder to model transferred requests along the routes of a solution, most works in the literature (Mitrović and Laporte (2006), Masson et al. (2013), Masson et al. (2014)) consider the duplication of requests at the transfer (i.e. request r_i is duplicated in (i^+, t_i^-) and (t_i^+, i^-)). In order to avoid those duplications while modeling interactions between vehicles, we propose a model in which transfer locations are duplicated in inbound and outbound nodes for a given route at a transfer, regardless of how many requests are being transferred (dropped-off or picked-up) by a given route at a particular transfer location. Given a (partial) solution of the PDPTW-T, s, let $R_s \subset R$ be the set of requests served by this solution. The set of pickups (deliveries) serviced in s is denoted $P_s(D_s)$. If a vehicle v_1 visits a transfer location $t \in \Gamma$, nodes $a_t^{v_1}$ and $d_t^{v_1}$ representing the arrival (inbound) and departure (outbound) operations, respectively, of v_1 at t are created. Denote the set of transfers arrival's and departure's nodes in solution s by I_s and O_s, respectively. Figures B.1a and B.1b illustrate the correspondence between routing implementation and the model. Requests r_i and r_k are transferred between vehicles v_1 and v_2 at transfer location t.

The *support graph* of solution s is the directed graph $G_s(W_s, A_s)$ where $W_s = M \cup P_s \cup D_s \cup I_s \cup O_s$ and A_s contains arcs (i, j) such that i and j are visited by the same vehicle and j is visited immediately after i, or $i \in I_s,\ j \in O_s$ such that i is an arrival vertex at a transfer for one vehicle and j the departure vertex at the same transfer location for another vehicle. Those last arcs (between arrival and departure nodes) capture relations between two vehicles at a transfer: if a vehicle v_1 transfers one or more requests to another vehicle, v_2, at transfer location t, then arc $(a_t^{v_1}, d_t^{v_2}) \in A_s$ i.e. the departure of v_2 at t depends on the arrival of vehicle v_1 at t. Figure B.2 illustrates a solution with three vehicles and two transfers, t_1, t_2. At transfer location t_2, vehicle v_1 receives request r_i



(a) Routing implementation. Vehicle v_1 (blue) and v_2 (red) meet at transfer location t.



(b) Transfer representation in routes. $I_s = \{a_t^{v_1}, a_t^{v_2}\}, O_s = \{d_t^{v_1}, d_t^{v_2}\}$.

Figure B.1: Modeling Transfer Operations

B

from vehicle v_2, and vehicle v_2 receives requests r_j and r_l from v_1. Vehicle v_3 only visits location t_1 to drop-off request r_k, which is picked-up at t_1 by vehicle v_2.

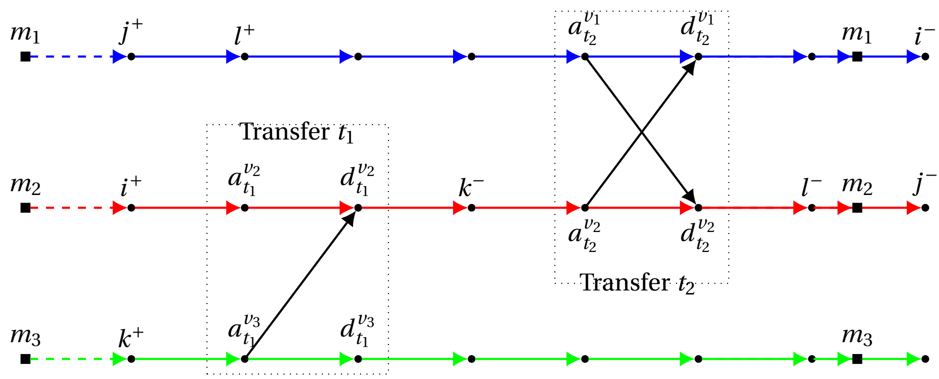


Figure B.2: Vehicles v_1 and v_2 are synchronized at t_2 by $(a_{t_2}^{v_1}, d_{t_2}^{v_2})$ and $(a_{t_2}^{v_2}, d_{t_2}^{v_1})$, and vehicles v_2 and v_3 by arc $(a_{t_1}^{v_3}, d_{t_1}^{v_2})$. $I_s = \{a_{t_2}^{v_1}, a_{t_1}^{v_2}, a_{t_2}^{v_2}, a_{t_1}^{v_3}\}, O_s = \{d_{t_2}^{v_1}, d_{t_1}^{v_2} d_{t_2}^{v_2}, d_{t_1}^{v_3}\}$

B.1. EVALUATING AND UPDATING SOLUTIONS

Given a solution s and its associated support (precedence) graph G_s, a route for vehicle $v \in V$ is a trip starting at its correspondent depot, m_v, visiting a sequence of vertices $i \in P \cup D \cup I_s \cup O_s$ and back to m_v. Let $k_v = \{m_v, i_1, i_2, ..., i_n, m_v\}$ represent the sequence visited by vehicle $v \in V$ in solution s and denote by ρ_i and σ_i the predecessor and successor sets, respectively, of vertex i on its route. Note that vertices $i \in P \cup D$ have exactly one direct predecessor and one direct successor (i.e. $|\rho_i| = |\sigma_i| = 1$), whereas vertices $i \in I_s \cup O_s$ (inbound and outbound operations at transfers) might have more than one successor ($i \in I_s$) or more than one predecessor ($i \in O_s$). With a slight abuse of notation, we refer to the (unique) direct predecessor (successor) for vertices $i \in P \cup D$ as ρ_i (σ_i).

INFORMATION MAINTAINED ON EACH NODE

For every location $i \in P \cup D$ already assigned to a route, let e_i and l_i be the earliest time and the latest time, respectively, that service can start at location i (note that E_i and L_i are the time windows for request r_i, but e_i and l_i are variables changing accordingly to other elements in the route). For the vertices representing transfer locations in solution s, let e_i, l_i for $i = a_t^v \in I_s$, denote the earliest and latest possible arrival times for vehicle $v \in V$ at transfer $t \in \Gamma$, respectively. For $i = d_t^v \in O_s$, let e_i, l_i denote the earliest and latest possible departure times for vehicle v at transfer t. For each vehicle $v \in V$ in the solution, let $e_v = E_v$, the earliest time it can leave from its depot, and $l_v = L_v$, the latest time it can be back at its correspondent depot.

Given a solution s, earliest and latest times for visits in a route $\{i_0 = m_v, i_1, i_2, ..., i_n, i_{n+1} = m_v\}$ can be computed as follows:

- For $k = 1, ..., n$, let $e_{i_k} = \max\{E_{i_k}, e_{i_{k-1}} + \tau_{i_{k-1}, i_k}\}$ if $i_k \in P \cup D \cup I_s$, and $e_{i_k} = \max_{j \in \rho(i_k)} e_j$ if $i_k \in O_s$.

- For $k = n, ..., 1$, let $l_{i_k} = \min\{L_{i_k}, l_{i_{k+1}} - \tau_{i_k, i_{k+1}}\}$ if $i_k \in P \cup D \cup O_s$, and $l_{i_k} = \min_{j \in \sigma(i_k)} l_j$ if $i_k \in I_s$.

B

CHECKING THE FEASIBILITY OF INSERTIONS

Let $r_i = (i^+, i^-)$ a request to be inserted in a (partial) solution s, for which earliest and latest values are already computed for nodes visited by the routes in s. The feasibility of inserting i^+ between i_k and i_{k+1} is checked by computing:

$$e_{i^+} = \max\{E_{i^+}, e_{i_k} + \tau_{i_k, i^+}\}$$

and

$$l_{i^+} = \min\{L_{i^+}, l_{i_{k+1}} - \tau_{i^+, i_{k+1}}\}$$

if $e_{i^+} \leq l_{i^+}$ the insertion is feasible. Given that the insertion of i^+ is feasible, a feasible insertion for i^- can be searched on the same route wherein the insertion of i^+ was checked, or on a different route, using a transfer location. Checking the feasibility of inserting i^- consists in the same procedure described before, but considering updated values e_i, l_i for vertices i in all routes affected by the (feasible) insertion of i^+. Even if pickup and delivery are inserted on the same route, more than one route might need to be updated after an insertion due to the presence of transfers in the solution.

UPDATING ROUTES AFTER AN INSERTION

The insertion of a node (either a pickup or delivery) affects nodes before and after it on route $k_v = \{m_v, i_1, i_2, ..., i_n, m_v\}$ for vehicle v. When the vehicle does not visit any transfer location, inserting i between nodes i_k and i_{k+1} might modify earliest times for nodes $i_{k+1}, ..., i_n$ and latest times for nodes $i_1, ..., i_k$: the detour time to visit the inserted location might require that subsequent visits have to start later (increased earliest time) and service at prior visits might not be postponed as much later as before (decreased latest time) Campbell and Savelsbergh (2004).

When route $k_v = \{m_v, i_1, i_2, ..., i_n, m_v\}$ visits one or more transfer locations, an insertion can alter earliest and latest times of nodes in the routes of vehicles transshipping requests with v at any of those transfer locations. Figure B.3 illustrates a simple example with two vehicle routes, v_1 and v_2 (blue and red routes, respectively). If node i is to be inserted in route v_1, between nodes i_k and i_{k+1}, then, due to the fact that v_1 transfers one or more requests to v_2 at transfer location $t \in \Gamma$ (the arc (a_t^1, d_t^2) indicates this fact), vehicle v_2 can only leave transfer t after the arrival of vehicle v_1 i.e. $e_{d_t^2} \geq e_{a_t^1}$. Thus, earliest times for nodes visited by v_2 in the sub-path starting at d_t^2 onwards, until the depot, might have to be updated. If v_2 transfer any request to other vehicles at any other transfer in that sub-path (i.e. an arc $(a_t'^2, d_t^w)$ for a transfer $t' \in \Gamma$ and a vehicle $w \in V$), then earliest times for nodes visited by those vehicles might also have to be updated.

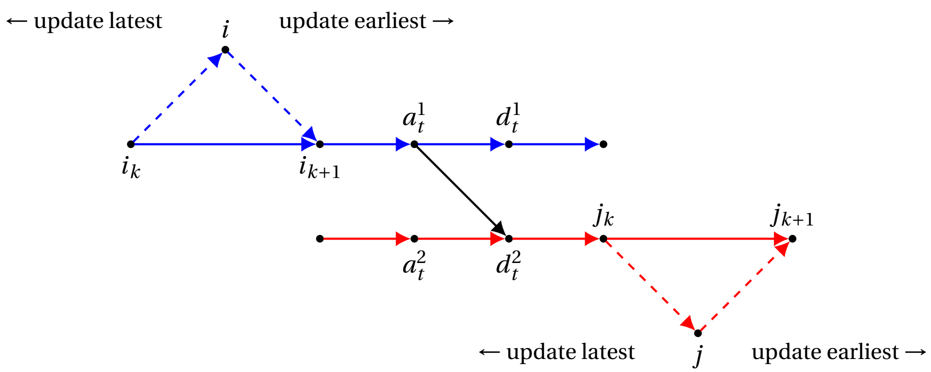


Figure B.3: Insertion and update procedures

Similarly, inserting j in the route $k_{v_2} = \{m_{v_2}, j_1, j_2, ..., j_n, m_{v_2}\}$ for vehicle v_2, between nodes j_k and j_{k+1} can alter the latest service times for nodes visited by vehicle v_1 since the the latest time vehicle v_1 can arrive at t ($l_{a_t^1}$) is bounded by the latest departure time at transfer t for vehicle v_2 ($l_{d_t^2}$). Latest times for nodes visited by other routes connected with v_1 at transfer locations in the sub-path starting at a_t^1 backwards might also need to be updated.

Observe that if node i was inserted in the route for vehicle v_1 after the visit to transfer t, then this insertion would not require that latest times for nodes in the route for vehicle v_2 to be updated. Similarly, if j was inserted in the route for vehicle v_2 before the visit to transfer t, then earliest times for nodes visited by vehicle v_1 would not need to be updated.

Updating is performed similarly to the process for computing all earliest and latest times for a solution, but taking into account that the solution already contains previously computed values e_i and l_i, so only locations that might be affected by the insertion are considered. After the insertion of node i between nodes i_k and i_{k+1} in the route $\{m_v, i_1, i_2, ..., i_n, m_v\}$, the update procedure is as follows:

- For $k' = k+1, ..., n$, let $e_{i_{k'}} = \max\{E_{i_{k'}}, e_{i_{k'-1}} + \tau_{i_{k'-1}, i_{k'}}\}$ if $i_{k'} \in P \cup D \cup I_s$, and $e_{i_{k'}} = \max_{j \in \rho(i_{k'})} e_j$ if $i_{k'} \in O_s$.
- For $k' = k, ..., 1$, let $l_{i_{k'}} = \min\{L_{i_{k'}}, l_{i_{k'+1}} - \tau_{i_{k'}, i_{k'+1}}\}$ if $i_{k'} \in P \cup D \cup O_s$, and $l_{i_{k'}} = \min_{j \in \sigma(i_{k'})} l_j$ if $i_{k'} \in I_s$.

To speed up the procedure, observe that whenever an earliest or latest update does not change the original value, updating can stop. This is especially useful since that might avoid unnecessary updates for routes connected at transfers. For example, after the insertion of j between j_k and j_{k+1}, if the latest time for node d_t^2 does not change, it is not necessary to continue with the update for v_2 and to update v_1 backwards from a_t^1.

C

Detailed Results for the Dynamic VRPRDL

C.1. Only delivery requests, no lead-time announcement

Dynamic Solutions - C_{60} Instances, $\gamma = 0$						
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range
HM	0	987.2	41.5	10.8	5.1	26.5
	15	925.9	43.2	9.9	4.4	26.7
	30	936.0	43.6	10.0	4.5	26.3
	60	957.2	43.2	10.0	4.4	25.8
$0E$	0	980.5	48.5	11.8	6.4	25.2
	15	837.7	52.3	9.4	4.5	24.0
	30	814.4	53.1	9.7	4.4	23.9
	60	815.0	53.3	9.7	4.4	24.3
$10E$	0	974.9	44.4	10.8	5.4	24.6
	15	842.2	46.8	9.8	4.9	22.7
	30	857.5	47.0	9.9	4.9	23.7
	60	873.6	46.9	9.6	4.9	23.1
$25E$	0	973.9	36.8	10.5	5.4	24.7
	15	887.2	38.4	9.2	4.4	23.6
	30	887.1	38.5	8.9	4.3	23.8
	60	884.3	38.6	9.0	4.3	23.8

Table C.1: Dynamic solutions obtained with the proposed MPA on C_{60} instances. Each line is the average over the 10 instances in the class.

C

Dynamic Solutions - C_{90} Instances, $\gamma = 0$						
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range
HM	0	1698.4	55.5	19.2	10.1	28.9
	15	1279.4	65.9	14.9	6.9	25.7
	30	1248.4	66.1	14.4	6.1	25.7
	60	1262.5	66.0	14.0	6.1	26.1
$0E$	0	1351.0	73.8	17.1	8.1	25.4
	15	1217.4	77.9	14.5	7.4	23.9
	30	1145.3	78.8	13.8	6.8	23.7
	60	1141.2	79.0	13.4	6.6	23.4
$10E$	0	1457.8	65.4	18.4	8.4	25.0
	15	1252.8	69.7	16.0	6.9	23.4
	30	1248.8	70.2	16.2	6.7	23.5
	60	1265.0	70.6	16.1	7.2	23.2
$25E$	0	1494.6	53.7	17.0	7.9	26.2
	15	1322.7	56.7	15.5	6.9	24.4
	30	1313.7	57.3	15.4	6.9	24.4
	60	1317.9	57.2	15.5	7.0	23.9

Table C.2: Dynamic solutions obtained with the proposed MPA on C_{90} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{120} Instances, $\gamma = 0$						
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range
HM	0	2705.2	51.9	29.5	18.2	32.9
	15	1780.5	81.1	19.7	9.2	27.9
	30	1767.8	83.0	19.4	8.5	27.9
	60	1816.4	83.1	19.2	9	28.1
$0E$	0	2549.5	66.2	28.9	19.8	32.8
	15	1552.5	102.0	18.8	10.8	25.1
	30	1534.5	104.5	18.4	9.9	24.4
	60	1540.1	104.9	18.2	10.2	25.1
$10E$	0	1777.5	88.4	21.3	9.5	25.9
	15	1574.5	93.8	18.6	9.3	24.6
	30	1558.7	94.7	18.7	8.5	24.5
	60	1561.7	94.6	18.6	8.8	24.4
$25E$	0	1799.6	73.9	20.9	8.6	25.8
	15	1672.9	76.6	19.5	8.0	25.1
	30	1676.0	76.7	19.1	7.7	25.1
	60	1706.2	76.5	19.5	7.9	25.5

Table C.3: Dynamic solutions obtained with the proposed MPA on C_{120} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{150} Instances, $\gamma = 0$						
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range
HM	0	2748.0	75.6	31.4	17.3	30.3
	15	1932.0	100.7	23.0	9.3	25.8
	30	1930.9	103.0	22.7	9.8	26.2
	60	1904.5	103.2	22.0	9.3	25.6
$0E$	0	2668.6	97.0	32.0	20.4	30.4
	15	1750.0	129.3	22.3	11.8	24.5
	30	1734.2	130.4	22.3	12.5	24.0
	60	1782.5	131.3	22.5	13.1	24.0
$10E$	0	2040.7	108.5	24.8	11.1	25.0
	15	1825.1	116.4	22.6	10.1	23.6
	30	1834.3	118.1	22.4	10.0	24.0
	60	1858.5	118.1	22.7	10.3	23.9
$25E$	0	2041.9	91.3	23.6	9.2	25.5
	15	1801.2	95.0	21.0	8.1	24.1
	30	1791.1	95.3	20.6	8.0	23.7
	60	1781.9	95.9	20.6	8.2	23.7

Table C.4: Dynamic solutions obtained with the proposed MPA on C_{150} instances. Each line is the average over the 10 instances in the class.

C.2. ONLY DELIVERY REQUESTS, LEAD-TIME ANNOUNCEMENT

Dynamic Solutions - C_{60} Instances, $\gamma > 0$						
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range
HM	0	1049.6	41.2	11.7	4.7	26.4
	15	873.1	38.2	8.4	3.9	24.1
	30	907.6	43.4	9.9	4.2	26.2
	60	904.3	43.7	9.6	4.1	25.7
$0E$	0	1124.7	43.7	12.6	7.5	26.9
	15	827.7	52.6	9.3	4.5	24.5
	30	815.5	53	9.4	4.4	23.9
	60	832.8	52.8	9.3	4.2	24.1
$10E$	0	888.4	46.4	10.8	5.1	23.2
	15	844.3	46.9	9.6	4.6	22.4
	30	817.0	47.6	9.4	4	22.3
	60	804.8	47.5	9.3	4	21.9
$25E$	0	907.1	38.1	9.8	4.6	24.2
	15	873.1	38.2	8.4	3.9	24.1
	30	886.9	38.3	8.5	3.9	24.1
	60	868.4	38.5	8.5	3.9	24.1

Table C.5: Dynamic solutions obtained with the proposed MPA on C_{60} instances. Only delivery orders and positive lead-time announcement ($\gamma > 0$). Each line is the average over the 10 instances in the class.

C

C

Dynamic Solutions - C_{90} Instances, $\gamma > 0$						
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range
HM	0	1533.6	58.1	17.3	8.2	28.1
	15	1298.7	57.1	15.3	6.4	24.3
	30	1292.6	66.1	14.6	6.2	26.0
	60	1266.3	66.7	14.8	6.2	25.4
$0E$	0	2261.0	69.75	27.0	16.75	30.2
	15	1138.3	78.0	13.2	7.1	23.4
	30	1122.2	79.2	12.9	7.0	23.0
	60	1104.5	79.4	12.8	6.9	23.4
$10E$	0	1424.0	64.5	17.7	8.8	25.2
	15	1263.9	70.1	15.8	7	23.6
	30	1239.0	70.7	15.8	6.5	23.1
	60	1243.8	70.9	15.9	6.8	23.4
$25E$	0	1472.2	53.2	17.3	7.8	25.8
	15	1298.7	57.1	15.3	6.4	24.3
	30	1302.2	57.4	15.5	6.4	24.1
	60	1320.3	56.9	15.3	6.5	24.3

Table C.6: Dynamic solutions obtained with the proposed MPA on C_{90} instances. Only delivery orders and positive lead-time announcement ($\gamma > 0$). Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{120} Instances, $\gamma > 0$						
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range
HM	0	2815.1	45.4	30.1	19.2	34.7
	15	1659.5	77.3	19.2	8.0	24.9
	30	1711.0	83.0	19.7	8.1	27.0
	60	1729.7	84.2	19.6	8.3	27.5
$0E$	0	2261.0	69.75	27.0	16.75	30.2
	15	1522.1	102.4	18.4	9.5	24.3
	30	1508.1	103.8	17.7	9.7	24.7
	60	1519.5	105.4	17.9	9.5	23.9
$10E$	0	1747.1	87.7	21.1	10.5	25.7
	15	1572.4	93.6	18.5	8.7	24.5
	30	1596.9	94.0	18.5	8.9	24.2
	60	1581.1	94.7	18.3	9.0	23.6
$25E$	0	1782.4	73.2	20.9	9.0	24.9
	15	1659.5	77.3	19.2	8.0	24.9
	30	1654.3	77.1	18.9	7.9	24.7
	60	1689.6	77.5	19.1	8.2	24.9

Table C.7: Dynamic solutions obtained with the proposed MPA on C_{120} instances. Only delivery orders and positive lead-time announcement ($\gamma > 0$). Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{150} Instances, $\gamma > 0$						
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range
HM	0	3128.4	60.3	35.3	22.7	32.6
	15	1735.5	95.4	20.0	8.4	23.3
	30	1913.2	103.1	23.1	9.8	25.4
	60	1895.5	103.5	22.3	9.3	25.1
$0E$	0	2883.0	88.9	34.8	23.2	30.9
	15	1752.2	130.1	22.7	11.2	23.7
	30	1697.7	132.3	21.1	10.4	23.6
	60	1689.8	132.5	21.3	10.9	23.3
$10E$	0	2214.4	106.0	28.1	12.7	25.6
	15	1802.3	117.2	22.9	10.8	23.3
	30	1834.7	117.5	22.9	10.5	23.6
	60	1796.7	118.2	22.2	10.5	23.0
$25E$	0	1880.7	92.2	22.2	10.0	24.0
	15	1735.5	95.4	20.0	8.4	23.3
	30	1779.2	95.7	20.8	8.6	23.2
	60	1746.8	95.7	20.2	8.3	23.1

Table C.8: Dynamic solutions obtained with the proposed MPA on C_{150} instances. Only delivery orders and positive lead-time announcement ($\gamma > 0$). Each line is the average over the 10 instances in the class.

C.3. DELIVERY AND 25% PICKUP REQUESTS, NO LEAD-TIME ANNOUNCEMENT

C

Dynamic Solutions - C_{60} Instances, $\gamma = 0$, 25% pickup requests							
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range	Picks
HM	0	982.6	40.5	10.9	5.3	26.9	1.5
	15	898.6	43.0	9.5	4.0	26.5	0.9
	30	905.9	43.6	9.7	4.3	26.0	0.7
	60	916.6	43.3	9.5	4.3	26.0	0.7
$0E$	0	950.5	48.8	11.7	6.6	24.8	1.6
	15	801.0	53.0	9.7	4.6	22.7	0.2
	30	807.9	53.2	9.7	4.8	23.6	0.3
	60	827.2	53.2	9.7	4.4	24.2	0.2
$10E$	0	970.5	44.7	10.9	5.4	24.9	1.5
	15	817.2	47.2	9.7	4.9	22.8	0.5
	30	841.1	47.1	9.8	5.1	23.0	0.5
	60	855.5	47.0	9.2	5.0	22.5	0.6
$25E$	0	985.2	36.5	10.8	5.5	25.2	0.4
	15	892.4	38.3	9.2	4.4	24.0	0.4
	30	882.2	38.5	9.0	4.3	23.7	0.5
	60	897.3	38.4	9.1	4.3	23.9	0.4

Table C.9: Dynamic solutions obtained with the proposed MPA on C_{60} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{90} Instances, $\gamma = 0$, 25% pickup requests							
St	W	$\Delta_o(\%)$	Roam.	Routes	Fleet	Range	Picks
HM	0	1479.6	59.8	16.4	7.8	28.1	5.4
	15	1279.8	65.6	15.0	6.6	25.8	1.6
	30	1232.5	65.7	14.1	6.4	25.9	1.7
	60	1265.1	65.9	14.3	6.1	26.0	1.7
$0E$	0	1406.0	72.2	16.7	8.7	26.6	5.2
	15	1142.8	77.6	13.7	7.1	23.7	2.2
	30	1123.5	78.7	13.7	6.9	23.6	1.4
	60	1129.6	79.7	13.4	7.0	23.6	1.1
$10E$	0	1410.8	65.3	17.7	8.3	25.3	2.7
	15	1225.0	70.0	15.8	6.7	23.1	1.2
	30	1270.1	70.4	16.4	7.0	23.4	0.6
	60	1262.3	70.7	16.2	7.2	23.5	0.4
$25E$	0	1433.4	54.4	16.6	7.4	25.7	3.0
	15	1334.8	56.6	15.4	6.8	25.0	0.6
	30	1291.0	57.4	15.3	6.6	24.2	0.6
	60	1307.3	57.2	15.6	6.7	24.0	0.3

Table C.10: Dynamic solutions obtained with the proposed MPA on C_{90} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{120} Instances, $\gamma = 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	Picks
HM	0	1993.5	65.9	22.4	11.5	30.1	6.9
	15	1718.4	80.6	19.0	8.8	27.6	3.1
	30	1737.1	82.0	19.0	9.3	27.4	2.3
	60	1758.8	82.6	19.3	9.0	27.9	1.7
0*E*	0	1762.8	88.5	21.4	12.1	26.6	6.0
	15	1521.2	103.7	19.1	11.0	24.3	1.9
	30	1513.0	104.3	18.2	9.9	24.3	2.2
	60	1510.9	104.8	18.0	9.7	24.4	1.4
10*E*	0	1700.9	89.1	20.4	10.3	25.6	4.1
	15	1580.8	93.2	18.8	9.0	25.2	1.1
	30	1563.7	93.8	18.5	8.6	24.4	1.0
	60	1595.1	94.3	18.5	8.7	24.6	0.4
25*E*	0	1756.3	75.4	20.8	9.5	24.6	2.5
	15	1697.3	77.1	20.1	8.0	25.0	0.3
	30	1665.7	77.1	19.5	7.5	25.1	0.5
	60	1663.6	77.0	19.2	7.7	25.0	0.2

Table C.11: Dynamic solutions obtained with the proposed MPA on C_{120} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{150} Instances, $\gamma = 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	Picks
HM	0	2619.0	76.3	29.4	16.7	30.6	9.5
	15	1908.1	99.5	22.1	9.1	26.1	5.6
	30	1884.5	102.4	22.1	9.3	25.8	2.8
	60	1877.9	103.5	21.8	10.1	25.7	2.2
0*E*	0	2640.0	95.7	31.5	20.7	30.2	10.4
	15	1792.2	125.3	22.4	12.4	24.5	4.0
	30	1740.7	129.4	21.6	12.0	24.1	2.4
	60	1778.7	131.6	21.7	13.1	24.3	1.5
10*E*	0	1920.2	113.4	24.2	10.8	24.6	4.5
	15	1790.4	116.9	22.2	10.5	23.6	1.7
	30	1777.9	117.9	21.8	10.0	24.0	0.7
	60	1827.0	118.1	22.1	10.1	23.8	0.5
25*E*	0	1862.7	93.5	21.9	8.1	24.2	3.3
	15	1804.2	95.1	20.8	8.2	24.4	1.2
	30	1797.3	95.3	20.3	7.6	23.9	0.6
	60	1766.8	95.9	20.3	8.1	23.7	0.4

Table C.12: Dynamic solutions obtained with the proposed MPA on C_{150} instances. Each line is the average over the 10 instances in the class.

C.4. DELIVERY AND 25% PICKUP REQUESTS, LEAD-TIME ANNOUNCEMENT

Dynamic Solutions - C_{60} Instances, $\gamma > 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	Picks
HM	0	1002.6	41.0	11.1	5.2	26.8	2.1
	15	898.9	43.0	9.9	4.4	26.2	1.0
	30	889.1	43.2	9.9	4.4	25.9	1.3
	60	873.9	43.4	9.4	4.1	25.4	1.3
$0E$	0	1134.7	41.5	12.4	7.6	29.0	2.9
	15	788.1	52.6	9.0	4.2	23.9	0.4
	30	825.1	53.0	9.4	4.3	24.6	0.3
	60	812.1	53.0	9.2	4.2	24.0	0.3
$10E$	0	904.1	46.4	11.0	5.4	23.5	1.8
	15	830.7	47.0	9.5	4.5	22.4	0.6
	30	813.9	47.4	9.5	4.3	22.1	0.7
	60	800.5	47.7	9.3	4.1	21.7	0.5
$25E$	0	950.0	37.6	9.9	5.0	25.0	2.2
	15	862.7	38.1	8.3	3.6	24.0	1.6
	30	882.7	38.2	8.8	3.8	24.3	1.2
	60	876.8	38.4	8.6	3.8	24.7	1.3

Table C.13: Dynamic solutions obtained with the proposed MPA on C_{60} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{90} Instances, $\gamma > 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	
HM	0	1420.0	60.2	16.0	7.2	27.3	6.5
	15	1279.7	65.5	14.4	6.0	26.5	3.0
	30	1214.0	65.7	13.9	6.4	26.0	2.9
	60	1206.1	66.4	13.8	5.9	25.7	2.6
$0E$	0	1346.2	71.4	16.4	9.5	25.7	4.4
	15	1105.2	78.4	13.4	7.0	23.4	1.4
	30	1115.4	79.3	12.7	6.9	23.2	0.7
	60	1106.4	79.9	12.5	6.7	23.6	0.6
$10E$	0	1349.5	65.8	17.1	8.4	25.2	3.3
	15	1240.3	70.5	15.5	6.8	23.6	0.8
	30	1242.5	70.9	15.6	6.7	23.3	0.7
	60	1232.7	71.2	15.7	7.0	23.2	0.6
$25E$	0	1348.3	55.2	15.9	6.3	25.3	2.7
	15	1296.0	57.3	15.5	6.6	24.4	1.0
	30	1294.4	57.2	15.4	6.4	24.5	1.1
	60	1298.5	57.0	14.9	6.4	24.1	0.9

Table C.14: Dynamic solutions obtained with the proposed MPA on C_{90} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{120} Instances, $\gamma > 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	
HM	0	2016.3	68.8	22.0	10.0	29.5	6.4
	15	1671.3	80.8	18.6	8.4	27.4	4.8
	30	1689.4	81.2	18.5	8.0	27.4	3.7
	60	1677.2	82.4	18.7	8.2	27.5	2.4
$0E$	0	2301.1	69.4	26.8	17.9	30.8	7.8
	15	1484.4	102.9	17.8	9.3	25.2	3.1
	30	1480.6	104.7	17.1	9.2	24.4	2.4
	60	1464.4	106.0	17.0	9.3	23.9	1.1
$10E$	0	1675.1	88.8	20.2	9.6	25.8	4.9
	15	1567.9	94.0	18.2	8.9	24.3	1.6
	30	1576.1	94.2	18.3	8.7	24.3	0.5
	60	1579.8	94.7	18.2	9.1	23.9	0.1
$25E$	0	1693.8	74.1	19.9	8.8	24.8	2.6
	15	1639.1	77.3	18.8	7.7	24.8	1.1
	30	1668.4	77.2	19.0	8.0	24.7	1.2
	60	1707.4	77.6	19.4	8.2	25.2	1.1

Table C.15: Dynamic solutions obtained with the proposed MPA on C_{120} instances. Each line is the average over the 10 instances in the class.

Dynamic Solutions - C_{150} Instances, $\gamma > 0$, 25% pickup requests							
St	W	Δ_o(%)	Roam.	Routes	Fleet	Range	Picks
HM	0	2555.7	76.3	28.5	15.9	30.3	11.8
	15	1869.4	102.5	21.7	8.5	25.8	4.0
	30	1873.5	103.9	21.4	8.8	25.4	3.0
	60	1867.6	104.7	21.7	9.1	24.8	2.7
$0E$	0	2660.8	96.4	31.9	20.8	29.6	8.6
	15	1725.3	130.2	22.4	11.4	23.7	3.1
	30	1676.8	131.9	21.5	10.9	23.5	2.9
	60	1705.2	132.6	21.7	11.5	23.8	2.1
$10E$	0	1955.4	110.0	24.8	10.7	24.9	5.4
	15	1781.0	117.3	22.5	10.5	23.2	3.1
	30	1808.5	117.8	23.0	11.3	23.2	1.7
	60	1782.5	118.5	22.3	10.7	22.8	0.7
$25E$	0	1801.1	92.3	21.6	9.2	24.1	3.9
	15	1705.2	95.4	19.9	8.7	23.2	1.3
	30	1746.7	95.8	20.3	8.6	23.4	1.0
	60	1757.3	95.4	20.3	8.1	23.3	0.3

Table C.16: Dynamic solutions obtained with the proposed MPA on C_{150} instances. Each line is the average over the 10 instances in the class.

SUMMARY

INNOVATIVE BUSINESS-TO-CONSUMER LAST-MILE SOLUTIONS: MODELS AND ALGORITHMS

Delivering goods in urban areas is one of the most challenging logistics activities. Given both the increasing urbanization levels and the share of e-commerce in retail sales, last-mile operations will only become more challenging in the future. The last-mile refers to the last link in the transport chain followed by a parcel to fulfill consumers' requests for goods, from the shelf of the last distribution center to the hands of the buyer. In fact, given the explosion of e-commerce, the last-mile supply chains of even the largest e-tailers are strained by the sheer volume increase of direct-to-consumer orders. This challenge is even amplified given the service levels offered by e-tailers to compete against the instant gratification of brick-and-mortar stores. Thus, companies are evaluating new and innovative business models that could help improving last-mile operations.

This thesis focuses on novel approaches for dealing with last-mile operations faced by logistics service providers in urban contexts. We investigate two recent innovative models and the potential cost-benefits of introducing such models into transportation logistics for last-mile operations. More specifically, we first consider crowd-sourced logistic solutions – e.g., where drivers are not employed by a carrier but occasionally offer their services through on-line platforms and are contracted as required by the carrier – on the fulfilment of logistic activities. The second novel model we consider is what has been defined as roaming delivery systems, in which the service provider has access to private cars' storage compartments and can service customers using the trunk of their cars. Supported by automotive and communication technologies, the model has the potential to make e-commerce operations more convenient, mitigating failed deliveries at home.

City Logistics advocates a holistic view of the transport and logistic activities within a city, considering the negative (e.g., congestion and pollution) as well as the positive (e.g., economic, mobility, safety) impacts on the city's population. It seeks cost-efficient, but sustainable, solutions that minimize required flows of people and goods. In Chapter 2, we focus on how the crowd-based models may be part of such cost-efficient, but sustainable solutions, especially related to the flows of goods. We define the characterizing features of crowd logistics, review applications of crowd-based services within urban environments and discuss potential research opportunities.

Chapter 3 focuses on the assessment of a transportation system considering the use of crowdshipping, where transportation capacity is provided by individuals willing to provide their time and their vehicle for a (short) period and evaluate the benefits of in-

troducing transfers to support driver activities. At transfer locations, drivers can drop off packages for pick up by other drivers at a later time. We frame the problem as an extension of a pickup and delivery problem with transfers and propose an adaptive large neighborhood search (ALNS) algorithm to solve it. Computational experiments indicate that introducing transfer options can significantly reduce system-wide travel distance as well as the number of drivers required to serve a given set of requests, especially when drivers have short availability, as in a crowd-based system, and requests have high service requirements.

Trunk delivery is a novel approach to last-mile delivery being tested by even the largest e-tailers companies as a means to increase service levels to consumers. In this approach, the company has access to the trunk of the customer's vehicle where delivery couriers can leave packages for that customer. In Chapter 4, we introduce a stochastic version of the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) and propose a two-stage stochastic model using the possibility of servicing customers at different locations as a recourse action. Our contribution to this routing literature stems from considering stochastic travel times in solving the problem. We believe such contexts are important given the increasing level of urbanization and its consequences on the ability of retail companies to fulfill promised service levels (in particular, in e-commerce) for last-mile delivery within urban environments e.g., due to increased levels of congestion. We exploit the flexibility introduced by trunk-delivery to define recourse policies able to reduce the number of failed deliveries caused by stochastic travel times

Finally, in Chapter 5, we introduce a dynamic variant of the VRPRDL in which customers are not required to inform their full planned routes at the day of service but announce, in real-time, the locations where their cars are or will be parked. The only information required by the service provider from its customers are their home locations and the corresponding customer's availability at home. A Multiple Plan Approach (MPA) is used in which multiple routing plans are maintained to provide alternative ways for reacting to dynamic information. The service provider then decides whether to visit customers at home or at their (announced) roaming locations. We consider a dynamic strategy to decide on the actions to take during the operational day to support vehicle dispatching decisions. While having customer itineraries revealed dynamically provides more flexibility to customers, compared to solutions considering only home visits, it can, however, potentially improve the operations of service providers.

About the author

Que outrem possa louvar esforço alheio,
Cousa é que se costuma e se deseja;
Mas louvar os meus próprios, arreceio
Que louvor tão suspeito mal me esteja;
E para dizer tudo, temo e creio,
Que qualquer longo tempo curto seja;
Mas, pois o mandas, tudo se te deve,
Irei contra o que devo, e serei breve. [1]

Luís Vaz de Camões, Os Lusíadas, Canto III

Afonso Sampaio was born in Belo Horizonte, Brazil, on June 10, 1985, together with his twin sister. He received both his bachelor's degree in Computational Mathematics (2011) and master's degree in Computer Science (2014) from the Federal University of Minas Gerais, Brazil. After receiving his M.Sc. degree, he worked as an assistant researcher in the project Port-Ship Coordinated Planning (PoShCoP) at Molde University College, Norway.

From September 2015, Afonso started as a PhD candidate at the Eindhoven University of Technology under the supervision of Tom Van Woensel, Luuk Veelenturf and Joris Kinable. His work was mainly focused on investigating innovative business models for last-mile services, and aimed at overcome the challenges posed by fulfilling those services within urban contexts. During his PhD, he presented his work on several academic conferences like INFORMS, IFORS and the ALIO/EURO on Combinatorial Optimization. He was also the teaching assistant in the master course Decision-making in Transport and Logistics. In 2018, Afonso was awarded a NWO grant to support a research visit to the Georgia Institute of Technologty (US), where he spent three months working in collaboration with Prof. Martin Savelsbergh. Parts of the work in Chapters 2 and 3 stem from this collaboration.

[1] That one praise other's efforts; is honour'd custom which we all desire; yet fear I it's unfit to praise mine own; least praise, like this suspect, no trust inspire; nor may I hope to make all matters known; for time however long were short: yet, as thou commandest, all is owed to thee; against my will I speak and brief will be.