PhD dissertation:
Optimizing operational costs and service quality in dial-a-ride systems

Yves Molenbruch

Hasselt University
Agoralaan Gebouw D
BE-3590 Diepenbeek

Research Foundation Flanders
Egmontstraat 5
BE-1000 Brussels

May 9, 2017

Acknowledgements

Writing this dissertation felt like removing barriers and expanding people's worlds. It was a pleasure to use my analytical skills to enable efficient, quality-oriented transportation of people with reduced mobility. This subject allowed me to live my passion for mobility while conducting research with an important social dimension, which has always been a great incentive. Furthermore, this dissertation has also expanded my own world. I had the privilege of making a challenging, but inspiring journey through academia. Though the roads in this world were not straight and smooth, they were littered with numerous opportunities to develop myself into a mature researcher. Looking back at this enriching experience, I am grateful for all opportunities that I received.

However, it would have been so much harder to reach my destination without the guidance, support and confidence of many people along the way. First and foremost, I would like to express my sincere gratitude to prof. dr. An Caris and to prof. dr. Kris Braekers for supervising this dissertation. An, your advice has always assisted me to choose the right direction at every crossroad that I encountered. Your valuable feedback ensured that I never lost sight of the big picture, while your trust in my capabilities fueled my motivation as a researcher. Kris, I've lost count of all the time you spent reading my work and providing me with suggestions. Yet, I will always remember how the insights that I gained during our meetings have considerably contributed to the quality of this dissertation. Furthermore, I would like to thank all jury members, whose interesting questions and valuable remarks have broadened the horizon of this dissertation. Special thanks go to prof. dr. Greet Vanden Berghe for accurately reading and commenting my draft papers throughout the process.

Performing research often feels like paving your path beyond existing roads. Fortunately, exploring this path is easier and more enjoyable whenever it crosses motivated people with similar ambitions. In this respect, I was lucky to be part of the discipline group Quantitative Methods, a network of colleagues connected by solidarity and friendship. We enjoyed our daily chats, laughs and bickering, as well as many coffee breaks, noontime walks and conference trips. We even found sufficient time to

work hard and to organize many pleasant activities after the office hours. The company and support of all QM'ers was undoubtedly a key to success.

Last but not least, I would like to thank my friends and relatives for showing their encouragements and sincere interest during the past years. Among all of them, one person deserves special recognition for all efforts that she made and for the unceasing support that she gave me. Mom, even after four years in the research group Logistics, I have never seen a better logistic service provider than you, as evidenced by - among many other things - the hundreds of shirts you ironed and meals you prepared while I was working on this dissertation. But most importantly, you have always been the rock I could lean on, my biggest supporter as well as my greatest believer. You taught me never to give up. I didn't.

In short, this dissertation is more than the destination of my trip through the fascinating world of science. It is also the merit of all people who traveled with me. I owe them many thanks.

Yves Molenbruch
Maasmechelen, May 2017

This research was funded by FWO (Research Foundation Flanders) and fits in the COMEX project (Combinatorial Optimization: Metaheuristics & Exact Methods) of the Belgian Science Policy Office.

Contents

Chapter 1

Problem statement

— *Short summary* —
Collective on-demand transportation systems are typically invoked to provide accessible transportation for elderly and disabled. The demand for these services will increase in the coming years, due to demographic evolutions and developments in the healthcare sector. Moreover, new policy visions on the organization of efficient public transportation include collective on-demand transportation as a basic component. Illustrating the role of such services in the rapidly changing Western societies, Sections 1.1 and 1.2 delineate the application context of the research conducted in this PhD thesis. Its contributions, defined in Section 1.3, arise from the operational challenges service providers face due to the growing scale of the associated vehicle routing problem. Specifically, this thesis proposes optimization algorithms and operational strategies to balance service quality and operational costs, taking into account real-life problem characteristics.

1.1 Mobility demand of elderly and disabled

Due to substantial societal changes, both healthcare and mobility have become challenging policy domains in most Western countries. Many of these challenges arise from a major demographic evolution, which is the ageing population. By 2030, an average Belgian citizen will be 42.67 years old, whereas this was only 39.65 in the year 2000 (FOD Economie, 2016). The share of people at retirement age (67+) will increase from 14.74% to 19.82% over the same period, given a fertility rate below the replacement level and an increasing life expectancy. Based on the evolution of mortality ratios, life expectancy at the age of 67 will rise from 17.26 years in 2000 to 20.82 years in 2030. Behind this seemingly promising statistic lies a major social challenge, since approximately half of the remaining lifetime is - unfortunately - spent in poor health conditions (European Commission,

2016). In the Belgian federal state structure, healthcare provided to elderly and disabled falls within the regional competences. This section explains the vital importance of those people's mobility in the healthcare policy of the Flemish Government.

The actual share of people with reduced mobility in the Flemish population is hard to estimate. A study on accessible transportation ordered by the Flemish Government (Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2013) reports that 169,510 citizens (or 3%), more than half of whom aged 67 or older, received a financial compensation on grounds of disability in 2010. This is a lower bound on the actual number of disabled, as people may not apply for compensation due to ignorance, shame or income restrictions. Most disabilities affect mental (55.95%) or physical (33.73%) health conditions, thereby often reducing the individual's independence in terms of mobility. Given the current demographic evolutions, the number of individuals receiving compensation will increase to 222,355 by 2030.

Elderly and - as far as possible - disabled often prefer to be cared for at home, assisted by home care services or informal caregivers (e.g. family members) as needed. However, individuals with reduced mobility are in need of accessible transportation for trips to medical consultations, rehabilitation centers, work or leisure activities. This demand for accessible transportation is enhanced by the policy of the Flemish Government, encouraging types of support that facilitate living at home (Vlaams Parlement, 2009) in order to shorten the waiting lists for permanent residential care[1]. For example, elderly or disabled may spend one or several days a week in a daycare center to relieve the pressure on their caregivers. In 2016, the government provides funds to subsidize 350 strategically spread daycare centers for elderly in Flanders, compared with 54 in 2001. Similar initiatives are taken for disabled. The right to accessible transportation is a prerequisite for making these daycare services available to their target group. At the same time, it contributes to a qualitative and more affordable healthcare policy that is better tailored to individual needs.

Given these policy objectives, the Flemish Government - also responsible for regional mobility - prefers to coordinate the organization of accessible transportation on its territory. In any case, it must comply with several regulations concerning equal opportunities. Both the European Union and Belgium ratified the UN convention on the rights of persons with disabilities (United Nations, 2006), which requires that reasonable adjustments are made to ensure the accessibility of public places

[1]For example, 84,952 permanent places in nursing homes will be available by the end of 2018 (Vandeurzen, 2015) owing to the Flemish Government's funding policy. However, already in 2012, at least 109,150 elderly were registered on waiting lists (van Buggenhout et al., 2012). Similar problems arise in specialized institutions for disabled. They incite policy makers to support a broader range of care provision besides permanent residential care.

through, amongst others, accessible transportation. The Flemish decree on equal opportunities and equal treatment (Vlaams Parlement, 2008) applies across all the policy domains, including mobility. The resulting mobility policy of the Flemish Government will be discussed in the next section.

1.2 A new and integrated policy vision on mobility

Many on-demand transportation systems were originally developed to provide a customized service for people with reduced mobility, supplementary to the more traditional public transportation. Policy makers in most Western countries recognized the inefficiency of regulating (and subsidizing) multiple transportation systems operating in the same area. This triggered new integrated policy visions, typically devoting more attention to demand-responsive approaches in order to decrease the overall operational cost. As an example, this section discusses current developments in the Flemish mobility policy, but similar trends can be observed in other Western countries.

1.2.1 Traditional policy

In Flanders, an integrated mobility policy will be implemented in the coming years. Currently, people who are unable to drive a vehicle should, depending on the severity of their mobility restriction, still choose among the following three separate systems.

First, a dense network of regular *public transportation* between predefined stops is subsidized by the federal and regional governments. Urban and regional bus services are exclusively organized by the autonomous public company *De Lijn*, whose majority stakeholder is the Region of Flanders (Vlaams Parlement, 2004). In 2001, the Flemish Government adopted the decree on *basic mobility*, imposing stringent service requirements (Vlaams Parlement, 2001). For example, inhabitants in rural areas should be able to reach a bus stop within 750 meters of their homes. This stop is served at least once an hour between 6 am and 9 pm on weekdays. Bus lines in sparsely populated areas may be exploited according to a semi-demand-responsive system. Such lines have fixed timetables, but rides are only executed after reservation up to one hour in advance. The annual subscription fee amounts to €299, reduced to €51 for elderly and people entitled to compensation (De Lijn, 2016). Although accessibility is a point of attention in the purchase of new vehicles and the refurbishment of bus stops, people with reduced mobility still experience major difficulties using public transportation in Flanders (Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2013). Besides, they may be unable to reach the closest bus stop, even if it is located only a few hundreds of meters from their home.

Second, most local authorities exploit a *Minder Mobielen Centrale* operated by *voluntary drivers*. This on-demand service provides transportation on prior request between any locations of choice. It targets people with limited income who are unable to use public transportation. Requests should be submitted 48 hours in advance, after which they are assigned to a volunteer using his own car. In addition to a €10 yearly subscription fee, users compensate the driver for the actual transportation cost through an average contribution of €0.30 per km (Minder Mobielen Centrale, 2016). Unfortunately, regular cars are generally not equipped to transport (electric) wheelchair users, which makes the system unusable for part of the target group. Moreover, finding sufficient volunteers to meet all demand at a particular moment in time may be a challenging task.

Third, *Diensten voor Aangepast Vervoer* are companies using adapted vehicles and professional drivers to provide accessible on-demand transportation. Users may be served collectively, meaning that different requests are combined if this is beneficial from an operational point of view. In other words, the vehicles may serve a second user while a first user is still aboard, provided that certain service criteria (e.g. time preferences, maximum travel time, ...) are respected. Within this category, the Flemish Government created a network of *Openbaar Aangepast Vervoer* providers. Such a subsidized provider complies with the *compensation decree* of the Flemish Government (Vlaams Parlement, 2013), imposing maximum rates and accessibility requirements. For example, the overall price of a wheelchair user's ride over a 50 km distance may consist of a fixed €2 fee supplemented with a contribution of €1.75 per km (Meer Mobiel, 2016). The Flemish Government subdivided the regions of Flanders and Brussels into 27 transportation areas, admitting one subsidized provider per area. Users submit their request 48 hours in advance to the *Mobiliteitscentrale Aangepast Vervoer* located in their province. Users may also invoke non-subsidized private service providers, who can charge higher rates. Such providers often specialize in the transportation of particular user profiles or offer specific quality guarantees. In Chapter 2, *Diensten voor Aangepast Vervoer* will be formally introduced as *dial-a-ride* systems.

This overview illustrates that the current transportation offer is fragmented and lacks coordination, which is financially unsustainable in the long term (Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2009, 2013). The decree on *basic mobility* caused the yearly passenger rides by *De Lijn* to double from almost 250 million in 1998 to 500 million in 2008, while the government expenses covering the exploitation costs tripled from €40 to €120 per inhabitant. Combined with a low-fare policy, the cost coverage percentage of *De Lijn* decreased from 32% in 1998 to 16% in 2008. The subsidies provided for *Openbaar Aangepast Vervoer* are equally alarming. Based on the current demand patterns, modal split and demographic evolution, the number of users will increase from 25,232 in 2010 to 33,126 in 2030. Under an unchanged policy, the yearly government compensation

will rise from €437 million to €876 million over the same period.

1.2.2 Integrated policy

The only viable future scenario consists of an integrated system combining the benefits of regular public transportation and on-demand services. In a recently adopted conceptual note (Vlaamse Regering, 2015), the Flemish Government recognizes that the current model is highly inefficient: "Basic mobility hampers an efficient, demand-driven public transportation. The stringent requirements cause high costs for public transportation, whereas they do not correspond to a real demand, do not stimulate an efficient combination of mobility modes and make it financially impossible to reinforce important axes where public transportation may generate large returns. At the same time, there is a lack of flexibility to respond to limited demand in rural areas." (translated from Dutch)

In the same note, the Flemish Government introduces the concept of *basic reachability* as an integrated vision on mobility to elaborate a consistent subsidy strategy based on an overall efficient use of resources. Their vision is driven by two fundamental innovations. First, the mobility policy should be demand-based, rather than imposing requirements on the transportation offer. Second, the policy should focus on combined mobility through a hierarchical node network in which different transportation modes connect to each other. On the one hand, *De Lijn* will organize regular public transportation on important axes (the core network, e.g. urban and interurban connections) and functional lines (the supporting network, e.g. rides for commuters or students during peak hours). On the other hand, tailored transportation will be provided where/when demand is more limited, among which collective on-demand transportation by private companies or volunteers. The territory will be divided into 13 transportation zones based on the geographical spread of the demand. Within these zones, all stakeholders (including local authorities) will combine their expertise to ensure a coherence between different transportation modes, facilitated through real-time information and an integration of tickets and fares. This integrated vision on mobility still has to be elaborated in further detail and a number of pilot projects will be developed. A comprehensive legal framework will be provided by 2018.

Based on this description of *basic reachability*, the role of collective on-demand transportation systems will be enhanced through their integration into the regular public transportation network. This creates major commercial opportunities for service providers, but also important operational challenges. One of these challenges relates to the underlying vehicle routing problem that has to be solved, i.e. the process of assigning the user requests to a particular vehicle route and determining appropriate time schedules. Nowadays, many service providers perform this task either manually or

with limited technological support. Even if any routing software is used, manual adaptations of the routing solutions are often required to take into account real-life constraints. Since it is increasingly difficult to construct feasible and efficient routing solutions as the scale of the problem expands, service providers will need tailored algorithms to support and optimize their operational activities. These algorithms may also be used to analyze potential changes in their service policy.

1.3 Research objectives and thesis overview

This thesis is situated in the field of operations research, which consists of developing analytical techniques to support decision making in business economics. Its focus will be on the vehicle routing algorithms for on-demand transportation. On the one hand, these algorithms should minimize the operational costs of a service provider subject to a number of technical constraints on the feasibility of routes, which will be discussed in Chapter 2. On the other hand, solution algorithms should also respond to the quality expectations of users, which are more difficult to grasp. While designing the decree on *basic reachability*, the Flemish Government ordered a market survey (Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2013) among 344 individuals with various disabilities representing the current target group of collective on-demand transportation. The respondents were questioned about their quality expectations and their openness to changes in the service level. The average respondent attending an activity with a fixed starting time would accept a deviation of 20 minutes from his preferred time of arrival/departure at the activity location. Over 80% of the respondents would accept a detour of 10 minutes while being aboard the vehicle, whereas less than half of them would appreciate a detour of 20 minutes. Most respondents are not prepared to pay more than €0.60 per km and would like to have the possibility of submitting requests up to two hours in advance. In summary, users attach particular importance to the service level they receive and many would even not accept a lower service quality in exchange for a price reduction. However, as will be shown throughout this thesis, decisions related to this service level policy strongly affect the operational costs incurred by the service provider.

Figure 1.1 visualizes this tradeoff for a small network involving two users. Each user i needs to be transported from a pickup node P_i to a delivery node D_i. In addition, user 1 should reach his destination within the time interval [16,21], whereas user 2 should be picked up within the interval [11,16]. The vehicle depot is located at the center of the network and the travel times on all arcs are indicated. It is assumed that the length of each arc is equal to the corresponding travel time. Figure 1.2 presents a solution in which the service provider offers the highest-quality service to both users, as they do not face any detour between their origin and destination. From an operational point of view, the service provider needs to activate two vehicles and the total distance traveled

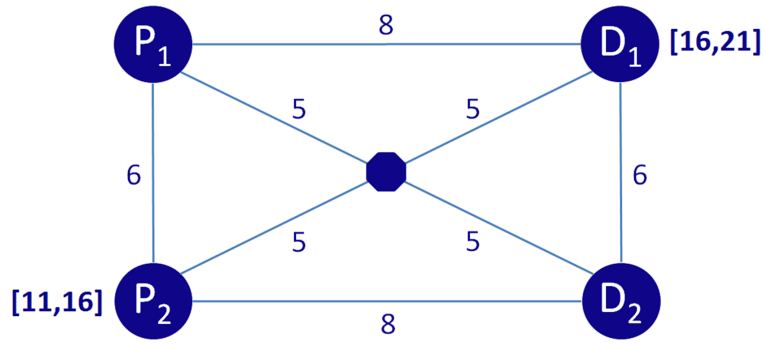

Figure 1.1: Example network illustrating the impact of service quality on the operational costs.

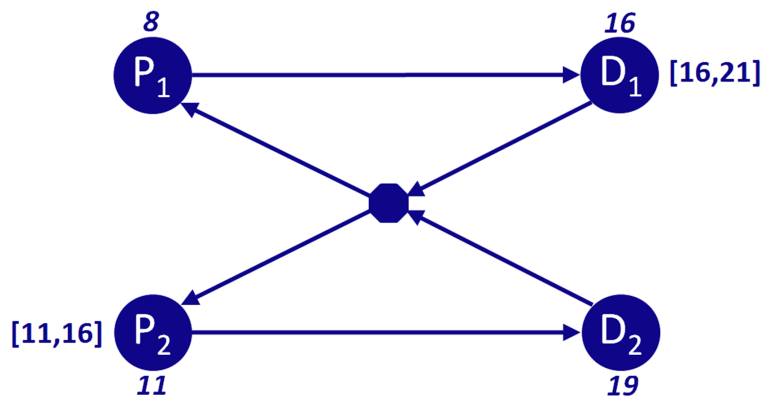

Figure 1.2: Example solution with excellent service quality and high operational costs.

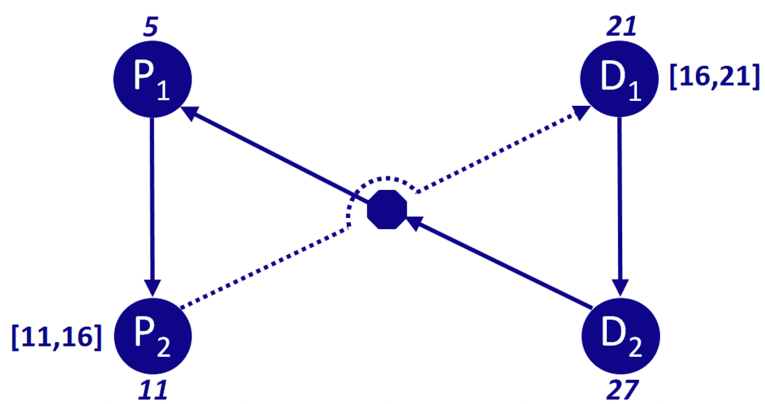

Figure 1.3: Example solution with poor service quality and low operational costs.

equals 36. A feasible start time of the service in each node is indicated. In contrast, Figure 1.3 shows a solution in which service quality is reduced to improve the operational efficiency. Both users are served by the same vehicle, traveling a total distance of 32. However, a feasible time schedule (respecting the users' time preferences) can only be found by combining them on the dotted arc. As a result, both users face a detour of 8 time units, since they spend 16 time units aboard the vehicle.

Although the importance of service quality is inherent to on-demand passenger transportation, Paquette et al. (2009) observe that most academic contributions in this domain are directed towards solution algorithms minimizing operational costs subject to minimum service level criteria, without optimizing service quality as such. They conclude that additional research is needed to understand the tradeoff between costs, operational policies and service quality. Therefore, this thesis provides innovations in optimization techniques and operational strategies to incorporate service quality into the vehicle routing process. Using the terminology of Schneider and White (2004), service quality in demand-responsive transportation systems can easily be approached from a technical perspective by measurable attributes, such as waiting or travel times. However, also a customer-based approach is required to know the expectations and practical needs of users (e.g. wheelchair accessible vehicles). This thesis combines both approaches by focusing on (1) providers' internal optimization processes and operating strategies and (2) problem features characterizing real-life systems. More details on the aims and structure of this thesis are presented below.

The objective of this thesis is to investigate how service providers may obtain a more efficient balance between the service level offered to their customers and the corresponding operational costs incurred. This involves a combination of two problem aspects. First, service providers should dispose of a vehicle routing algorithm that allows them to compute the tradeoff of operational costs and service quality in an efficient manner. Second, once this tradeoff is known, a solution that provides an acceptable balance between both objectives should be selected. Therefore, particular attention is paid to (1) the development of enhanced optimization techniques that support providers' vehicle routing processes and (2) strategic choices of service providers with respect to their quality policy. Moreover, this thesis aims to demonstrate how the tradeoff between service quality and operational costs is affected by the characteristics of the operational setting, which should encourage academics to consider sufficient real-life problem features in the development of solution methods. This results in the following central research question:

How can the service level and the operational costs of dial-a-ride systems be balanced in an efficient manner, taking into account the influence of real-life problem characteristics?

A comprehensive answer to this question can only be provided after addressing it from different perspectives. The central research question is therefore split into the following subquestions, related to the different components of the problem:

(1) What is the state of the art in solving dial-a-ride problems with real-life characteristics?
(2) How important is the effect of service level variations on the operational costs?
(3) How can optimization techniques reveal the tradeoff of the service level and operational costs?
(4) How can operating strategies of service providers balance the service level and operational costs?

As shown in Figure 1.4, this thesis consists of seven interrelated chapters, each addressing one or several subquestions. The remainder of this section discusses the contributions of these chapters and the relationships between them.

Chapter 1 explains the societal contribution of the research conducted in this thesis, delineating the application context of collective on-demand transportation systems. These systems are gaining importance in Western countries as a result of demographic changes, developments in the healthcare sector and new policy visions on efficient public transportation. Moreover, this chapter introduces the objectives and research questions of the thesis, focusing on the operational challenge of balancing the service level and operational costs while solving the associated vehicle routing problem.

In Chapter 2, this vehicle routing problem is formalized as a *dial-a-ride* problem and its standard characteristics are described. To enhance the practical applicability of solution approaches, authors increasingly include additional real-life features. This chapter introduces a classification of problem characteristics and solution techniques discussed in the academic literature, devoting specific attention to recent developments. Based on this typology and literature review, lacunae in the research conducted to date are identified as research opportunities.

Chapter 3 quantifies the sensitivity of the operational costs to variations in constraints related to the ride times and time preferences of the users, which determine the service level. Various operating circumstances are considered regarding the size of the service provider, traffic conditions and heterogeneity of users. The resulting tradeoff between the service level and operational costs substantiates the relevance of the research conducted in this thesis. It should encourage service providers to make well-informed decisions regarding the service level offered to users.

Most solution techniques minimize the operational costs subject to minimum service level requirements. In contrast, Chapter 4 emphasizes the fundamental problem nature by proposing an efficient

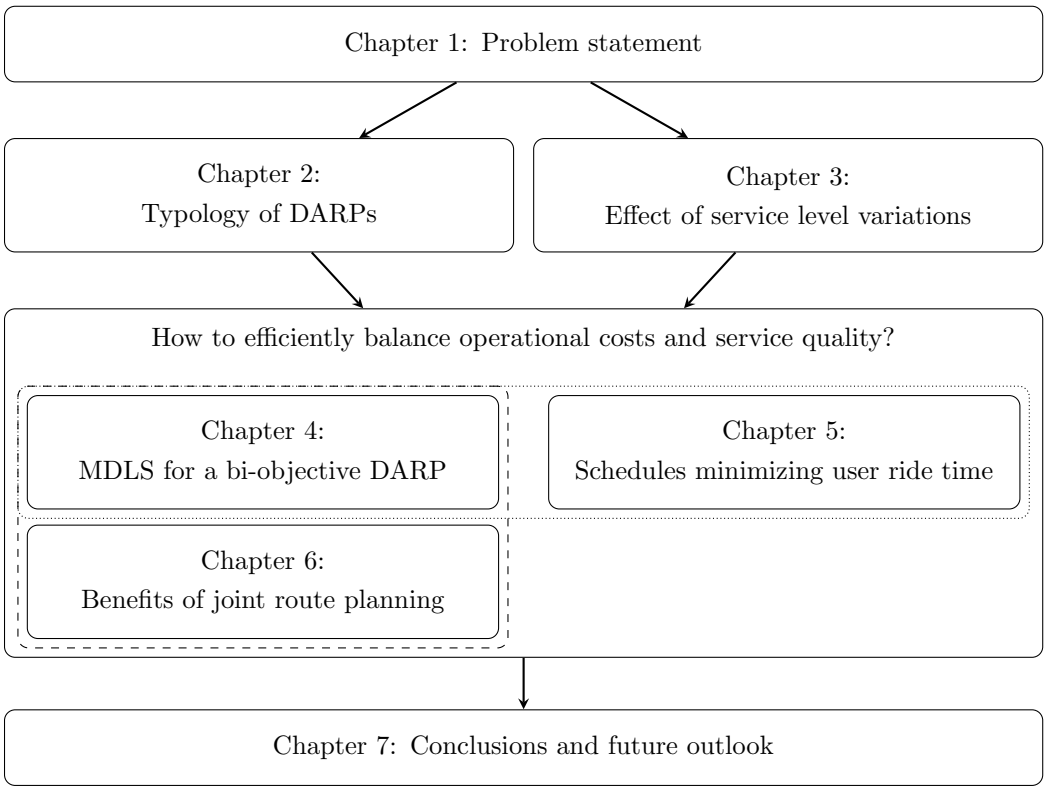

Figure 1.4: Thesis overview

multi-directional local search metaheuristic that solves a bi-objective problem variant. The total ride time over all users is incorporated as an additional objective. The resulting set of non-comparable solutions provides practitioners with useful insights to balance service quality and operational costs. This chapter also investigates the effect on operational costs caused by combination restrictions in patient transportation, such as users requiring service by medically qualified drivers.

Since minimizing the overall ride time is an objective in the aforementioned solution approach, the vehicle routes are ideally submitted to a scheduling procedure that minimizes the total ride time of the users and delivers a feasible schedule whenever one exists. Unfortunately, no such procedure has been developed to date. Yet, Chapter 5 presents a new scheduling heuristic which fails on fewer routes and is faster than existing work.

Chapter 6 studies joint route planning as an operational strategy for competing service providers to balance costs and quality. The overall efficiency may improve through exchanges of user requests, which is also relevant in light of the aforementioned evolution towards an integrated mobility policy. A large neighborhood search metaheuristic is developed to compare scenarios with and without such centralized decision making. Multiple operational characteristics are shown to influence the savings obtained. A real-life case study reveals the underlying mechanisms of request exchanges.

Finally, Chapter 7 draws general conclusions based on the findings in all preceding chapters and identifies opportunities for future research. Specific recommendations are formulated to three types of stakeholders influencing the usability of dial-a-ride systems: the academic community, the service providers and the government. Their decisions and actions determine the extent to which a system satisfies the needs of the users, representing the fourth main stakeholder in the context of this thesis.

Chapter 2

Typology and literature review of dial-a-ride problems

— *Short summary* —
In the academic literature, the routing problem arising in the collective on-demand transportation systems described in Chapter 1 is known as the *dial-a-ride* problem. The aim of this chapter is to formalize the characteristics of different problem variants and discuss the corresponding solution methods presented in the literature. Section 2.1 delineates the scope of this review and clarifies the strategy adopted while composing it. Section 2.2 starts by presenting a mathematical formulation of the standard problem. However, since authors increasingly focus on enhancing the applicability of their solution methods by incorporating real-life problem characteristics, an up-to-date classification of all problem variants is introduced. Mainly variants that involve heterogeneity, more complicated routing properties and stochastic or dynamic information have gained much interest in recent years. Section 2.3 structures the wide range of proposed solution methods, devoting particular attention to recent developments in the field of metaheuristics and matheuristics. This section also includes an analysis of different procedures to handle the scheduling subproblem, since this also determines the efficiency of a solution method. Section 2.4 identifies the lacunae in the research conducted to date, as well as opportunities for future research. For completeness, a comprehensive overview table containing details on all problem characteristics and solution methods is provided in Appendix A. Note that this chapter is based on Molenbruch et al. (2017b).

2.1 Introduction

A dial-a-ride system is an application of demand-dependent, collective passenger transportation (Cordeau and Laporte, 2007). Users request a trip between an origin and a destination of choice, to which a number of service level requirements are linked. The service provider attempts to develop efficient vehicle routes and time schedules, respecting these service level requirements and the technical constraints of a pickup and delivery problem (Parragh et al., 2008). A frequent objective is to minimize operational costs subject to full demand satisfaction and side constraints, but service level criteria may be optimized as well. Balancing human and economic perspectives involved in solving such a dial-a-ride problem (DARP) is essential for organizing a quality-oriented, yet efficient service. Due to the rising demand, service providers may no longer manage to compose vehicle routes and schedules manually. Routing algorithms are required to safeguard cost efficiency and service quality.

This chapter analyzes the existing literature on routing algorithms for the dial-a-ride problem, which delivers a fourfold contribution to the research field. First, a classification of existing problem variants is established. Originally, most authors adopted the standard DARP definition (Cordeau and Laporte, 2003), which explains why early literature reviews (Cordeau and Laporte, 2007; Parragh et al., 2008) only distinguish between static or dynamic variants, as well as single-vehicle or multi-vehicle systems. Nowadays, most authors tend to address additional real-life problem characteristics, enhancing the practical applicability of their solution methods. Although some extensions have been mentioned in recent book chapters (Parragh et al., 2010a; Doerner and Salazar-González, 2014), this work provides a more comprehensive classification to distinguish multiple categories of real-life problem characteristics. Within each category, similarities and differences between papers are analyzed. As a second contribution, this chapter reviews the wide range of solution approaches applied to various problem variants. Whereas the discussion emphasizes general evolutions in the literature, full details on the problem characteristics and solution methods in each single paper are listed in an exhaustive overview table in Appendix A, which is a third contribution. This table also reflects the focus of this chapter, being the literature on the definition of problem variants and the development of corresponding optimization algorithms. The increasing relevance of this subject is evidenced by the fact that half of all reviewed papers have been published during the last five years. As a final contribution, promising directions for future research are identified by identifying trends and lacunae in the current state of the art.

Two search strategies were adopted to collect all relevant literature in a structured way. First, electronic bibliographic databases, such as EBSCOhost and Web of Knowledge, were consulted. A broad search was performed for academic papers having the concepts ‘dial-a-ride’, ‘on-demand

transportation', 'demand-responsive transportation' or 'patient transportation' in their title, content or keywords. Examining these search results, papers focusing on problem variants and optimization algorithms have been retained for the actual literature review process. Second, additional literature was identified on the basis of references by other authors. These papers were obtained by searching other sources or by contacting the authors. Once again, the aforementioned criteria were used to decide which of them have been included. Papers dealing with related pickup and delivery problems have been included if they deliver a general contribution which is also relevant to the literature on the DARP.

2.2 Problem characteristics

This part identifies and classifies all problem variants that have been considered in the literature. Section 2.2.1 discusses the standard DARP (Cordeau and Laporte, 2003) and its mathematical formulations. Although this standard problem has been studied extensively, practical applications of dial-a-ride systems often exhibit numerous additional problem characteristics. Algorithms ignoring such extensions may deliver infeasible or unrealistic solutions. Therefore, the current trend of studying rich vehicle routing problems with a broader practical applicability (Caceres-Cruz et al., 2014) is also noticeable in research on the DARP. More specifically, the classification scheme presented here is based on the observation that real-life systems may require three types of extensions to the standard problem definition. The first type involves additional characteristics related to the design of the service. On this matter, Section 2.2.2 presents problem variants that consider heterogeneous users, vehicles or drivers, more complex routing properties and different service level specifications. The second type of extension relates to which objective function is optimized. While the standard problem definition includes a single operational objective, Section 2.2.3 discusses a wide range of operational and service-related objectives whose practical relevance has been described in the literature. The third type of extension concerns the nature of the information available to the service provider. The standard problem assumes that data is deterministic and known before vehicle routes and schedules are designed. In contrast, Section 2.2.4 focuses on problem variants that involve dynamic or stochastic information on travel times, requests or user behavior. Table 2.1 outlines the detailed classification scheme of additional real-life problem characteristics and indicates the references associated with each category. Papers delivering multiple contributions may appear in several categories. To conclude the analysis of problem characteristics, Section 2.2.5 summarizes past and current trends in the literature and identifies emerging topics for future work.

2.2.2 Advanced service design

2.2.2.1 Heterogeneity

Beaudry et al. (2010), Braekers et al. (2014), Detti et al. (2016), Hanne et al. (2009), Liu et al. (2015), Parragh (2011), Parragh et al. (2012), Qu and Bard (2013), Wong and Bell (2006), Xiang et al. (2006), Zhang et al. (2015)

2.2.2.2 Routing properties

Aldaihani and Dessouky (2003), Beaudry et al. (2010), Braekers et al. (2014), Braekers and Kovacs (2016), Detti et al. (2016), Häll et al. (2009), Hanne et al. (2009), Karabuk (2009), Liu et al. (2015), Masmoudi et al. (2016), Masson et al. (2014), Parragh et al. (2012), Parragh et al. (2014), Zhang et al. (2015)

2.2.2.3 Quality specifications

Diana and Dessouky (2004), Jaw et al. (1986), Jørgensen et al. (2007), Melachrinoudis et al. (2007), Paquette et al. (2009), Toth and Vigo (1997)

2.2.3 Alternative objectives

2.2.3.1 Single objective

Attanasio et al. (2004), Berbeglia et al. (2012), Diana et al. (2006), Feuerstein and Stougie (2001), Garaix et al. (2011), Krumke et al. (2006), Parragh et al. (2014), Rekiek et al. (2006), Ritzinger et al. (2016), Yi and Tian (2005)

2.2.3.2 Multiple objectives

Diana and Dessouky (2004), Guerriero et al. (2013), Jørgensen et al. (2007), Lehuédé et al. (2013), Mauri and Lorena (2006), Melachrinoudis et al. (2007), Paquette et al. (2013), Parragh et al. (2009), Schilde et al. (2011), Schilde et al. (2014), Wong and Bell (2006), Xiang et al. (2006)

2.2.4 Stochastic or dynamic

2.2.4.1 Travel times

Fu (2002), Schilde et al. (2014), Xiang et al. (2008)

2.2.4.2 Requests

Attanasio et al. (2004), Beaudry et al. (2010), Berbeglia et al. (2010), Berbeglia et al. (2012), Coslovich et al. (2006), Cremers et al. (2009), Ho and Haugland (2011), Hanne et al. (2009), Schilde et al. (2011), Hyytiä et al. (2010)

2.2.4.3 User behavior

Heilporn et al. (2011)

Table 2.1: Classification of papers extending the standard problem characteristics of Cordeau and Laporte (2003).

2.2.1 Standard DARP

2.2.1.1 Definition

A standard definition of the DARP has been established by Cordeau and Laporte (2003). The problem consists of designing a number of minimum-cost routes in a complete graph of nodes and arcs such that all user requests are satisfied. Nodes correspond to pickup and delivery locations of users, supplemented with the vehicle depot. Each directed link between two nodes is an arc, characterized by a travel time and an associated cost which is incurred if the arc is part of the solution. Each route starts and ends at the depot within fixed time intervals and respects a maximum route duration. The service at each user location starts within a time window. The maximum user ride time cannot be exceeded and a vehicle's load cannot violate its capacity. In order to ensure a correct physical route construction, precedence and pairing of a user's origin and destination should be respected by visiting them in the right order, using the same vehicle. A service duration indicates the required time to load and unload the users. As indicated in the classification scheme of Parragh et al. (2008), this definition distinguishes the DARP from any related problem in vehicle routing. Most similar is the pickup and delivery problem with time windows (PDPTW), which also involves demand-dependent transportation between paired pickup and delivery locations. However, the definition of the PDPTW is based on the transportation of goods, which explains why fewer quality constraints need to be included. Particularly the maximum user ride time constraint is specific to the DARP and complicates the scheduling subproblem.

2.2.1.2 Mathematical formulations

Cordeau (2006) introduces an arc-based mixed-binary linear program, shown by equations 2.1-2.14. A three-index binary decision variable x_{ij}^k indicates whether vehicle k traverses the arc between nodes i and j. Each vehicle route is assumed to start at an origin depot and end at a destination depot (eq. 2.4, 2.6). However, this does not necessarily imply that the entire fleet is activated. Both depot nodes may be directly connected to each other, which means that the corresponding vehicle is not used in practice. One and the same vehicle should reach and leave corresponding pickup and delivery locations i and $n+i$ (eq. 2.2, 2.3, 2.5), which ensures flow conservation and pairing. The decision variable L_i^k computes the ride time of user i (eq. 2.9) and cannot exceed the maximum user ride time L (eq. 2.12). Explicit precedence constraints become redundant if L_i^k is set at least equal to the associated direct ride time. The decision variables B_i^k and Q_i^k track the service start in node i (eq. 2.7) and the load upon leaving node i (eq. 2.8), respectively. They should respect the time window of node i (eq. 2.11) and the capacity Q_k of vehicle k (eq. 2.13), respectively. The time span between the moment a vehicle leaves the origin depot and the moment it returns to the destination depot cannot exceed the maximum route duration T_k (eq. 2.10). A minimum-cost selection of arcs

is made (eq. 2.1), subject to all constraints and full demand satisfaction. A glossary of all notation used in this model can be found in the *List of symbols* section of this thesis.

$$\text{Minimize} \qquad \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij}^k x_{ij}^k \tag{2.1}$$

Subject to

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \qquad \forall i \in P \tag{2.2}$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \qquad \forall i \in P,\ \forall k \in K \tag{2.3}$$

$$\sum_{j \in N} x_{0j}^k = 1 \qquad \forall k \in K \tag{2.4}$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0 \qquad \forall i \in P \cup D,\ \forall k \in K \tag{2.5}$$

$$\sum_{i \in N} x_{i,2n+1}^k = 1 \qquad \forall k \in K \tag{2.6}$$

$$B_j^k \geq (B_i^k + d_i + t_{ij}) x_{ij}^k \qquad \forall i \in N,\ \forall j \in N,\ \forall k \in K \tag{2.7}$$

$$Q_j^k \geq (Q_i^k + q_j) x_{ij}^k \qquad \forall i \in N,\ \forall j \in N,\ \forall k \in K \tag{2.8}$$

$$L_i^k = B_{i+n}^k - (B_i^k - d_i) \qquad \forall i \in P,\ \forall k \in K \tag{2.9}$$

$$B_{2n+1}^k - B_0^k \leq T_k \qquad \forall k \in K \tag{2.10}$$

$$e_i \leq B_i^k \leq l_i \qquad \forall i \in N,\ \forall k \in K \tag{2.11}$$

$$t_{i,i+n} \leq L_i^k \leq L \qquad \forall i \in P,\ \forall k \in K \tag{2.12}$$

$$max\{0, q_i\} \leq Q_i^k \leq min\{Q_k, Q_k + q_i\} \qquad \forall i \in N,\ \forall k \in K \tag{2.13}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i \in N,\ \forall j \in N,\ \forall k \in K \tag{2.14}$$

Since an arc cannot be traversed by multiple vehicles, the vehicle index k can be avoided whenever route duration is unbounded and travel times/costs are vehicle-independent. Different vehicle capacities can still be modeled using artificial pickup and delivery nodes. In this case, the vehicles with lower capacity are forced to perform artificial pickups and deliveries at the beginning and at the end of their routes, respectively. Røpke et al. (2007) present two two-index formulations. The first only adapts constraints on the physical route construction. Pairing is ensured by precedence constraints, defining subsets of arcs which cannot jointly be part of a feasible solution. The second two-index formulation defines a single decision variable x_{ij}. Time and load restrictions are imposed by infeasible path constraints and rounded capacity constraints. Experiments by Røpke et al. (2007)

show that a branch-and-cut approach (see Section 2.3.1.1) is able to solve larger instances using these two-index formulations instead of the three-index formulation of Cordeau (2006). The second two-index formulation, containing fewer decision variables, is most efficient. However, eliminating L_i, B_i and Q_i limits possibilities to include objectives related to user quality (see Section 2.2.3). In addition, the number of constraints grows exponentially with the number of users.

As will be explained in Section 2.3, arc-based formulations are particularly suitable for branch-and-cut approaches. However, still other types of formulations have been presented. Set-partitioning and set-covering formulations (Parragh and Schmid, 2013) are useful for column generation. These formulations require a set of feasible routes. A minimum-cost selection is made such that the fleet size is respected and every request appears in exactly one (set-partitioning) or in at least one (set-covering) route. Constraint programming approaches may invoke a formulation based on successor variables (Berbeglia et al., 2011). A decision variable indicates the immediate successor of each node. Routes are represented as a closed circuit, assuming that the origin depot conceptually succeeds the destination depot. Another decision variable registers the vehicle performing service in each node, given that it should belong to the same route as its successor and its corresponding pickup/delivery node. The start of service and load in a node are tracked by separate decision variables. A feasible set of routes should be constructed subject to all constraints and full demand satisfaction.

2.2.2 Advanced service design

2.2.2.1 Heterogeneity

Parragh (2011) observes that users exhibit heterogeneous *physical needs*. They may travel in a wheelchair or on a stretcher and require the presence of one (Parragh, 2011; Parragh et al., 2012) or several (Liu et al., 2015; Zhang et al., 2015) accompanying persons. This is reflected in the layout of the vehicles, which consists of multiple capacitated resource types, such as staff seats, normal seats, wheelchair spaces and stretchers. If upgrading conditions apply, users can be assigned to a 'lower' resource type than requested. Parragh (2011) provides the formulation of Cordeau (2006) with an additional index, such that vehicle capacity should be respected for each resource type. Any set of eligible resource types may be defined for each user category. Parragh (2011) also extends the first formulation of Røpke et al. (2007) with artificial origin and destination depots for each vehicle. Tournament inequalities are defined, based on all paths which start at a particular artificial origin depot and violate a load constraint. Consequently, loads in each route can be checked without using a third index. The second formulation of Røpke et al. (2007) is adapted by Braekers et al. (2014). Rounded capacity constraints are formulated for the resource having the smallest capacity relative to the corresponding demand.

Qu and Bard (2013) note that service providers may use *configurable vehicles* to become more demand-responsive. The distribution of the total vehicle capacity over resource types is assumed to be adjustable, given a set of possible layouts which are inherent to the vehicle type. Mathematically, an additional index is added to the heterogeneous problem formulation of Parragh (2011), indicating which configuration should be selected while traversing an arc. Liu et al. (2015), Hanne et al. (2009) and Beaudry et al. (2010) also include disjunctive multi-dimensional capacities, but only the former provide a formulation. Wong and Bell (2006) consider two-dimensional vehicle capacity with two-way substitution, meaning that resources of both types can readily be converted into each other.

In addition to physical needs, users may be heterogeneous with respect to their *medical condition*. Hanne et al. (2009) and Beaudry et al. (2010) discuss priority constraints for patient transportation. Emergency requests receive first priority in order to complete their service within a limited time frame. Users may require isolated transportation due to contamination risks, after which the vehicle needs to return to the depot to be cleaned. Consequently, the depot may be visited multiple times throughout the day (see Section 2.2.2.2). Xiang et al. (2006) assume that a degree of complexity is associated with users and with fixed medical equipment in the vehicle. A driver is only assigned to users and vehicles which do not exceed his qualification level. Similarly, Hanne et al. (2009) define tasks for staff employees and tasks that may be performed by volunteers. Detti et al. (2016) extend the aforementioned mathematical formulations with real-life constraints in patient transportation, imposing incompatibilities among different types of patients and restrictions with respect to the vehicles that are eligible to transport them. Analogous constraints are investigated in Chapter 4.

2.2.2.2 Routing properties

Masson et al. (2014) allow users to be *transferred* between vehicles, which increases the productivity of systems with random origins and widespread, clustered destinations (or vice versa). On the other hand, transfers may be impracticable for the disabled, spread delays through the network and cause waiting times or discomfort. A formulation can be obtained by adding user ride time and route duration constraints to the formulation of Cortés et al. (2010) for the PDPTW with transfers. Transfer points are fixed and modeled as consecutive arrival and departure nodes. Upper and lower bounds on transfer times may be included. Since corresponding origins and destinations are not necessarily in the same route, a three-index decision variable registers which vehicle traverses each arc. Also the integration of on-demand transportation and regular public transportation may cause additional operational benefits. Guidelines to determine which requests are eligible for hybrid execution (Hickman and Blume, 2001; Aldaihani and Dessouky, 2003) consider distance traveled,

handicap of the user and design of the fixed routes. Although the on-demand routes are often created subject to predetermined transfer points, an integrated approach is desirable to balance travel times and total distance. Häll et al. (2009) assume that pickup and delivery locations are visited by on-demand vehicles, but the middle section of the ride may involve regular public transportation. Buses are assumed to drive so frequently that the waiting times at transfers can be neglected. Aldaihani and Dessouky (2003) permanently fix the transfer locations of each user before solving the routing problem, which may harm the solution quality.

Groups of users formulating a single *multiple-load request* are normally transported together. Parragh et al. (2014) note that splitting group requests among multiple rides may be more efficient from an operational viewpoint, even if the size of the group does not exceed the vehicle capacity. However, some typical split-delivery properties do not hold in the presence of precedence constraints (Nowak et al., 2008). For example, there may not exist an optimal solution in which each arc is traversed at most once. This explains why Parragh et al. (2014) refrain from an arc-based formulation. They present an alternative mixed-integer program in which nodes are assigned to certain positions in the routes, as well as a set-partitioning formulation. Qu and Bard (2014) and Liu et al. (2015) reduce large-sized problems by handling requests with similar characteristics as a group request.

Various authors consider a *multi-trip* problem variant, motivated by restrictions on staff availability and working times. Parragh et al. (2012) assume that accompanying persons are limited in number, work mornings or afternoons and should be picked up or delivered at the depot. Drivers take lunch breaks starting within a prefixed time window. An arc-based mixed-binary linear program requires an artificial noon depot. Binary decision variables indicate whether a vehicle takes an accompanying person during the morning and/or afternoon. Parragh et al. (2012) also provide a set covering formulation for this problem variant. Apart from lunch breaks, Zhang et al. (2015) and Liu et al. (2015) impose a maximum tour duration that requires vehicles to visit the depot for cleaning activities. Both authors present an arc-based mixed-binary linear program which distinguishes the subsequent tours of a vehicle using an additional index. Liu et al. (2015) assure that users may require assistance of multiple accompanying persons, who are limited in number. This introduces an additional manpower planning problem. Masmoudi et al. (2016) combine lunch and coffee breaks with heterogeneous users (Parragh, 2011) and multiple depots (Braekers et al., 2014) and present a comprehensive mathematical formulation. Other multi-trip contributions include Beaudry et al. (2010), Hanne et al. (2009) and Karabuk (2009). The latter two papers only insert lunch breaks in the final scheme, which is a non-integrated approach that is frequently applied in practice. Finally, Braekers and Kovacs (2016) consider a multi-period variant involving multiple operating days in order to ensure driver consistency. They propose three-index and four-index formulations in which

the total number of drivers serving each user is limited.

2.2.2.3 Specifications of service quality

Most problem variants involve two constraints that determine the service level experienced by users. A *time window* restricts the waiting period a user may face before being picked up or after being delivered. As noted in Jaw et al. (1986), a user is typically asked to indicate either a preferred departure time or a preferred arrival time. For an outbound trip to an appointment, it is important not to be late for the appointment, but the user may arrive somewhat earlier than requested. This gives rise to a time window for the destination of an outbound trip. The opposite reasoning applies to an inbound trip, resulting in a time window for the origin node. In both cases, the single time window for each trip is combined with a *maximum user ride time*, which cannot be smaller than the corresponding direct ride time. Since this limits the time a user can spend aboard the vehicle, an additional time window is implicitly constructed for the origin of an outbound trip (resp. the destination of an inbound trip). This time window can be narrowed for reasons of transparentness to the user (Diana and Dessouky, 2004). Paquette et al. (2009) review several approaches to express the maximum user ride time. Although most authors impose a fixed value for all users, this entails the counterintuitive consequence of a detour being feasible for a short trip and not for a longer one. Alternatively, the maximum user ride time may be defined as a multiple of the corresponding direct ride time or a maximum exceedance of the direct ride time could be imposed. The maximum user ride time may also be enforced indirectly by imposing two time windows for each user according to several strategies (Wolfler Calvo and Touati-Moungla, 2011; Parragh et al., 2014; Atahran et al., 2014). In Chapter 3, the operational effect of variations in both aforementioned service level criteria will be analyzed.

Time windows and maximum user ride times may also be seen as *soft constraints* (Jørgensen et al., 2007; Melachrinoudis et al., 2007), which means that violations are allowed but discouraged through a penalty cost. This approach increases the probability of finding a solution, but may result in users being late for an appointment or unforeseen waiting time for vehicles which are due to leave before the user is available. Analogously, rather than constructing a time window, deviations from the user's preference time can be penalized proportionally (Toth and Vigo, 1997). Noticeably, also violations of essential feasibility requirements (e.g. load constraints) are sometimes minimized (Mauri and Lorena, 2006), rather than imposed as hard constraints. Less frequent aspects of service quality are reviewed in Paquette et al. (2009). For example, some service providers avoid *waiting time* being assigned to a vehicle while a user is aboard, except for loading or unloading other users (Diana and Dessouky, 2004; Luo and Schonfeld, 2007; Schilde et al., 2014).

2.2.3 Alternative objectives

2.2.3.1 Single objective

Most solution methods include an *operational objective*, whereas the service level is assumed to be sufficient if all quality-related constraints are satisfied. The most common objective is minimizing the total distance traveled. Assuming that the cost and travel time associated with an arc are proportional to its length, minimizing the total routing costs (Cordeau and Laporte, 2003) or total vehicle travel time (Ritzinger et al., 2016) are equivalent objectives. Less common single-objective optimizations consider fleet efficiency, such as minimizing fleet size (Diana et al., 2006) or maximizing vehicle usage efficiency (Rekiek et al., 2006). Maximizing the vehicle occupancy rate stimulates social contacts among users, rather than enhancing the operational efficiency of the system (Garaix et al., 2011). Lastly, for private service providers who are not obliged to meet all demand, profit may be maximized based on a revenue associated with each request (Parragh et al., 2014).

In a dynamic environment, which is discussed in Section 2.2.4.2, different objectives are common. Full demand satisfaction may no longer be attainable if requests are revealed in real time. Therefore, the number of requests served in time may be maximized (Attanasio et al., 2004; Yi and Tian, 2005; Berbeglia et al., 2012). With users requesting immediate service, the time needed to complete all available requests or the average completion time may be minimized (Feuerstein and Stougie, 2001). Another relevant objective is a minimization of the maximum flow time (Krumke et al., 2006), with flow time being the difference between the release of a request and the start of its service.

2.2.3.2 Multiple objectives

Jozefowiez et al. (2008) mention three reasons why multiple objectives may be defined to solve a vehicle routing problem. First, the practical applicability of models may be enhanced, recognizing that the problem is not exclusively driven by a single operational objective. It may be insufficient to impose a minimum service level by means of constraints, which explains why DARPs are often optimized to a combination of operational and *quality-related objectives*. Common quality-related objectives are a minimization of the total user ride time and the total user waiting time, caused by deviations from the users' preference times. The maximum user ride time may be minimized as well (Lehuédé et al., 2013). Additional operational objectives may relate to a minimization of the vehicle idle time (Diana and Dessouky, 2004), total driver wage (Xiang et al., 2006), total route duration (Jørgensen et al., 2007) and taxi costs to cover a fleet shortage (Wong and Bell, 2006). The variable routing costs may be supplemented with a fixed vehicle activation cost, contributing to a minimization of the fleet size (Xiang et al., 2006; Guerriero et al., 2013). Second, multiple objectives may be defined to generalize a problem, considering some constraints as additional objectives. Whenever

the feasibility of a solution is enforced using soft constraints, additional *penalty terms* need to be included in the objective function (Mauri and Lorena, 2006; Jørgensen et al., 2007; Melachrinoudis et al., 2007). They penalize violations of a soft constraint and guide a solution procedure towards the feasible solution area. Profits for serving a user may be defined as an objective if request satisfaction is no hard constraint. Third, a decision maker may explicitly identify multiple objectives in real-life cases. For example, in models which consider an advanced service design, the objective function may minimize user inconvenience caused by the design characteristics (e.g. the number of transfers).

Jozefowiez et al. (2008) distinguish between two particular multi-objective strategies according to which the problem may be formulated. The *first category* applies scalar techniques that involve a mathematical transformation of the objectives. For example, weighted-sum objective functions are common for DARPs. Lehuédé et al. (2013) apply an advanced utility model, i.c. based on the Choquet integral, which also considers the interactions between multiple objectives. A hierarchical objective function (Schilde et al., 2011, 2014) only invokes an underlying objective if two solutions are equal in terms of the main objective. The epsilon constraint method optimizes only one objective while treating the other objectives as constraints (Parragh et al., 2009). The right hand side is varied to obtain and compare multiple results. Paquette et al. (2013) implement a reference-point model (Clímaco et al., 2006) that sums the distances between the objective values and an infeasible ideal point, computed using dynamics weights. However, all these models require a priori information on the relative importance of the different objectives. In addition, computing an at first sight readily interpretable objective value hides tradeoffs between different goals, which is essential information for a service provider to establish a targeted quality policy. In contrast, the *second category* consists of direct Pareto-based models, delivering a set of solutions that do not weakly dominate each other. This Pareto frontier does not contain any pair of solutions for which one solution is better at one objective and not worse at the other objective. Solution qualities of different sets can be compared using Pareto-compliant sorting criteria, like the hypervolume indicator (Zitzler and Thiele, 1999) or the multiplicative epsilon indicator (Zitzler et al., 2003). The MDLS strategy presented in Chapter 4 is directly based on this Pareto concept. Note that the Pareto frontier may also be approximated by applying solution approaches from the first category for multiple weight combinations (Parragh et al., 2009; Paquette et al., 2013), although this is only effective to discover supported solutions on the convex hull of the frontier (Tricoire, 2012).

2.2.4 Stochastic or dynamic information

The standard DARP makes two assumptions regarding the nature of the information available to the service provider. First, information is assumed to be static, which implies that all relevant

data (e.g. requests, travel times, ...) is known before the route planning process starts. It remains unchanged during the entire time horizon. Second, the data is assumed to be deterministic, meaning that it is not subject to variability or uncertainty. However, both assumptions rarely hold in real-life systems. Most service providers face dynamic changes in inputs and external conditions tend to induce stochasticity into the system. This section presents different causes that complicate the availability of information and explains how the subject has been addressed in the literature.

2.2.4.1 Travel times

Fu (2002) argues that in urban environments, a system's reliability can be increased by considering *stochastic* and *time-dependent* travel times. A normal distribution is assumed for the travel times on each arc. The average travel time varies with the departure time, whereas the standard deviation is assumed constant. The expected start of service in a node can be computed recursively and should respect the time constraints with a given probability. Routes being feasible in a deterministic context may be rejected if they exhibit a large variance. To estimate the travel times, Fu and Teply (1999) suggest three approaches, based on zones, distance and an artificial neural network. Xiang et al. (2008) include stochastic travel and service times in a dynamic problem context with possible breakdowns of vehicles. Hu and Chang (2015) consider deterministic time-dependent travel times, such that travel times between nodes are fixed for given time intervals. Schilde et al. (2014) model accidents as expanding and shrinking circles, causing congestion on the covered arcs.

2.2.4.2 Requests

Most authors assume that all requests are known in advance, such that static routes and schedules can be constructed. In the *dynamic* problem variant, additional information may be revealed during the routing or execution phase. The most studied case involves (part of) all requests being received in real time. These users either follow the usual reservation principle or ask for immediate service, in which case a maximum position shift may be imposed to respect the order of booking (Psaraftis, 1980). The service provider should be able to decide instantaneously whether an additional request can be inserted (Attanasio et al., 2004). For this purpose, Berbeglia et al. (2012) present a constraint satisfaction problem formulation which can be used to prove the infeasibility of a problem. Hyytiä et al. (2010) point out the risk of congestive collapse whenever the capacity of the control policy is exceeded, which suddenly causes an unacceptably high rejection rate. Specific problem contexts involving additional requests may be considered. For example, Hanne et al. (2009) and Beaudry et al. (2010) study transportation systems in a hospital context, where emergency requests should be serviced within a very limited time frame. Coslovich et al. (2006) focus on unexpected users asking for service during the stop of a vehicle. Cremers et al. (2009) consider subcontracting requests

to taxi services during peak moments. Apart from additional requests, several unexpected events related to users may be taken into account, including no-shows, cancelations or changed requests. For a review on dynamic DARPs, the reader is referred to Berbeglia et al. (2010).

Despite technological advances like vehicle localization systems or increased processing power, responding to newly revealed information remains a time-critical task. As pointed out by e.g. Santos and Xavier (2015), dividing the time horizon into smaller periods and repeatedly solving a static problem entails the risk of not being able to serve requests because of the slow response. Therefore, most authors focus on repairing existing solutions. In general, such repair heuristics first seek for a feasible solution when new information is revealed. Then, optimization is continuously performed until the next event occurs. Parallel computation may be applied (Attanasio et al., 2004). Nevertheless, a problem's real-time nature may heavily affect the efficiency with which it can be solved. This is reflected in the *competitive ratio*, being the worst-case proportion of an algorithmic result and the associated static optimum (Feuerstein and Stougie, 2001; Ascheuer et al., 2000; Lipmann et al., 2004; Feuerstein and Stougie, 2001; Krumke et al., 2006; Yi et al., 2006; Yi and Tian, 2005).

Apart from the dynamic DARP, problem variants with a limited availability of information may involve known requests with a *stochastic* or *probalistic* nature. Schilde et al. (2011) observe that users may be unable to specify their return time in advance. Rather than considering such inbound trips as dynamic requests, a statistical distribution can be used to anticipate possible inbound trips at various times. Ho and Haugland (2011) consider requests that are served with a given probability. In real-life cases, such a problem arises when fixed routes are executed on a regular basis, but users are absent with a known probability. For example, elderly people may not feel fit enough to go to the daycare center on a particular day. In this case, the order of the remaining nodes in the route remains unchanged. The authors construct routes such that expected costs are minimized, using a recourse function that takes into account the skipping probabilities.

2.2.4.3 User behavior

Reliability may be impacted by *stochastic user behavior*. Heilporn et al. (2011) consider users showing up late. In this case, the vehicle leaves at the scheduled time and a taxi is called, causing a cost which exceeds the savings of skipping the corresponding delivery. An arc-based mixed-binary linear program includes an expected delay cost. The probability of being late decreases as a user is visited later in his time window and thus also depends on the probability of skipping preceding nodes. Some users may be scheduled early in their time window to serve the majority of users as late as possible. This generates considerable savings over a deterministic optimum with expected

delay costs, even if the scheduling procedure is adapted.

2.2.5 Discussion and research opportunities

The literature analysis in the preceding sections reveals a trend of including more real-life problem characteristics. Particularly problem variants considering heterogeneity (e.g. multiple types of users, configurable vehicles), more complex routing properties (e.g. transfers, breaks) and dynamic or stochastic information gained interest in recent years, as demonstrated by the number of contributions in these categories. Although it is encouraging that authors base their work on a practical operating context, some challenges should be pointed out in this respect.

Recent surveys on service quality and user satisfaction (Paquette et al., 2009, 2012) still indicate an insufficient match with actual needs and concerns expressed by users. For example, users mention driver consistency and the possibility of last-minute requests as highly desirable, whereas these problem characteristics are not commonly included in the academic contributions. Consulting users, service providers and other stakeholders remains advisable to ensure the practical applicability of theoretical research. The research domain also lacks unification in the sense that the integration of problem characteristics has been overlooked. The overview table in Appendix A clearly illustrates that authors fail to combine the advancements achieved in the three aforementioned categories. For example, no research has been performed on the realistic combination of stochastic travel times and configurable vehicles, despite all efforts to investigate both characteristics separately. Appendix A allows the interested reader to identify similar gaps in the state of the art. From a managerial point of view, it is insufficient to include real-life problem characteristics without analyzing their effect on operational costs and service quality. For example, a provider may require a quantitative analysis on the benefits of purchasing configurable vehicles or opening multiple depots before actually doing the investment. Such insights are currently unknown for many problem variants, although at least a lower bound on the impact of real-life characteristics can be found (Masson et al., 2014).

On the other hand, a positive trend of solving the DARP in combination with related problems has recently arisen. Instead of considering requests and time preferences as an input to the routing problem, Coppi et al. (2013) claim that mutual benefits may be obtained by integrating health care scheduling and vehicle routing. They propose to schedule patient treatments in a hospital in such a manner that the costs of transporting the patients to the hospital is minimized. In other words, the overall costs of the healthcare provider and the dial-a-ride provider can be decreased if their operational activities are planned jointly. Li et al. (2016) integrate the on-demand transportation of people and parcels and propose an adapted scheduling procedure to avoid that slack time is scheduled

while people are aboard the vehicle. Santos and Xavier (2015) discuss a dynamic integration of the DARP and ride sharing by private car owners who are willing to deviate from their intended routes. Such integrated approaches inspired on real-life examples are interesting topics to address in future research. In that respect, it is noteworthy that the effect of integrating multiple DARPs with each other has never been investigated. Chapter 6 analyzes the benefits of centralized decision making through joint route planning among service providers operating in the same area.

2.3 Solution methods

As a second contribution, this work identifies and classifies the solution methods applied in the literature to date. Section 2.3.1 provides an overview of exact solution methods. However, due to the complexity of the problem, research has mainly been directed at the approximate solution methods discussed in Section 2.3.2. Table 2.2 outlines a detailed classification scheme of all approaches and indicates the references associated with each category. Papers delivering multiple contributions may appear in several categories. Remark that the solution approaches address different variants of the DARP, which can be seen in Appendix A. In addition, many solution approaches have only been tested on real-life instances of the authors themselves. Since the codes of the algorithms are usually not shared within the scientific community, authors rarely build upon each other's algorithms. For all these reasons, assessing and comparing the quality of the solutions obtained by all approaches is impossible. In fact, only solution approaches that have been applied to the standard benchmark instances of Cordeau and Laporte (2003) and/or Røpke et al. (2007) are directly comparable to each other. They are printed in italics in Table 2.2. The fact that these approaches obtain competitive results on the benchmark data can be seen as an indicator of their quality. Section 2.3.3 presents these benchmark data sets and includes an overview of the best-performing algorithms. An analysis of scheduling procedures, used to assess the time-feasibility of routes, is presented in Section 2.3.4. Section 2.3.5 includes a summarizing discussion and identifies opportunities for future research.

2.3.1 Exact methods

2.3.1.1 Branch-and-cut

Cordeau (2006) applies a *branch-and-cut* algorithm to an arc-based mixed-binary linear program. Preprocessing steps include time window tightening, elimination of arcs which violate physical or time-related requirements, as well as variable fixing due to incompatible requests. The formulation is strengthened using valid inequalities related to time and load variables, subtour eliminations, rounded capacity, precedence, generalized order, order matching and infeasible paths. Røpke et al. (2007) add inequalities related to fork structures and reachability. An initial pool of inequalities is

2.3.1 Exact methods

2.3.1.1 Branch-and-cut

Braekers et al. (2014), Braekers and Kovacs (2016), *Cordeau (2006)*, Häll et al. (2009), Heilporn et al. (2011), Liu et al. (2015), *Parragh et al. (2009)*, *Parragh (2011)*, *Røpke et al. (2007)*

2.3.1.2 Column generation

Garaix et al. (2011), *Parragh et al. (2012)*

2.3.1.3 Branch-(cut-)and-price

Gschwind and Irnich (2014), Parragh et al. (2014), *Qu and Bard (2014)*

2.3.1.4 Dynamic programming

Desrosiers et al. (1986), Häme (2011), Häme and Hakula (2013), Lois et al. (2007), Psaraftis (1980), Psaraftis (1983), Ziliaskopoulos and Kozanidis (2006)

2.3.2 Approximate methods

2.3.2.1 Classical heuristics

Borndörfer et al. (1999), Coslovich et al. (2006), Diana and Dessouky (2004), Ioachim et al. (1995), Jaw et al. (1986), Karabuk (2009), Luo and Schonfeld (2011), Maalouf et al. (2014), Teodorovic and Radivojevic (2000), Toth and Vigo (1997)

2.3.2.2 Metaheuristics based on local search

Aldaihani and Dessouky (2003), Attanasio et al. (2004), Baugh et al. (1998), Beaudry et al. (2010), *Braekers et al. (2014)*, Braekers and Kovacs (2016), *Cordeau and Laporte (2003)*, *Chassaing et al. (2016)*, Detti et al. (2016), *Guerriero et al. (2013)*, Häll and Peterson (2013), Ho and Haugland (2011), *Kirchler and Wolfler Calvo (2013)*, Lehuédé et al. (2013), *Li et al. (2016)*, *Masmoudi et al. (2016)*, Masson et al. (2013), *Masson et al. (2014)*, Mauri and Lorena (2006), Melachrinoudis et al. (2007), Muelas et al. (2013), Muelas et al. (2015), *Paquette et al. (2013)*, *Parragh et al. (2010b)*, Parragh et al. (2014), *Qu and Bard (2013)*, *Ritzinger et al. (2016)*, Schilde et al. (2011), Schilde et al. (2014), Toth and Vigo (1997), Wolfler Calvo and Touati-Moungla (2011)

2.3.2.3 Metaheuristics based on population search

Chevrier et al. (2012), Cubillos et al. (2007), Cubillos et al. (2009), Hanne et al. (2009), Jørgensen et al. (2007), *Masmoudi et al. (2017)*, *Parragh et al. (2009)*, Rekiek et al. (2006), *Zhang et al. (2015)*

2.3.2.4 Matheuristics

Berbeglia et al. (2012), Crawford et al. (2007a), Crawford et al. (2007b), *Jain and Van Hentenryck (2011)*, *Parragh and Schmid (2013)*, *Ritzinger et al. (2016)*

Table 2.2: Classification of papers with respect to the type of solution method applied, printed in italics if they have been tested on common benchmark data.

generated and the LP relaxation of the formulation is solved iteratively. If the root node solution is infeasible, additional cuts are generated using separation heuristics and branching is performed on the routing variables. Parragh (2011) and Braekers et al. (2014) adapt both the rounded capacity and strengthened capacity inequalities to a heterogeneous problem variant, defining them for each resource type. Häll et al. (2009) discuss which arc eliminations are (in)valid with the integration of regular public transportation. Liu et al. (2015) incorporate configurable vehicles into the capacity inequalities and define path elimination constraints for consecutive trips. The L-shaped algorithm of Heilporn et al. (2011) is a branch-and-cut application in a stochastic problem context.

2.3.1.2 Column generation

Column generation is a particularly useful technique to solve larger-scale instances, since the problem is split into a master problem and a subproblem. The master problem consists of solving a set partitioning/covering formulation, making use of a subset of promising routes. With multiple vehicle types, a different subset is assumed for each type (Parragh et al., 2012; Garaix et al., 2011). The subproblem consists of generating additional routes with a negative reduced cost, such that they may improve the objective value when added to the master problem. Although the subproblem may be approached in a heuristic (Coppi et al., 2013) or metaheuristic (Parragh and Schmid, 2013) manner, this discussion focuses on exact techniques. Garaix et al. (2011) and Parragh et al. (2012) represent the subproblem as an elementary shortest path problem with additional constraints, which can be solved using a labeling method based on dynamic programming. In addition to the cost upon reaching a node and a reference to its predecessor, other data should be stored to identify dominated labels (Røpke and Cordeau, 2009). This includes time and load information, supplemented with information related to the specific problem characteristics. The maximum user ride time constraint is implicitly included in the time windows. This rather strict approach excludes solutions which are considered feasible in the standard definition. For example, if a user defines a preferred pickup time for an inbound trip, the allowed deviation from this preference time is ignored in the computation of the user ride time. In other words, the ride time is assumed to start at the preference time, which is justifiable if the user has no interest in departing later than requested. Parragh et al. (2012) only launch the exact procedure if six pricing heuristics fail, two of which simplify the exact algorithm.

2.3.1.3 Branch-(cut)-and-price

The *branch-and-cut-and-price* approach of Gschwind and Irnich (2014) integrates column generation into a branch-and-cut algorithm, based on the observation that most variables in the solution are nonbasic. Compared with the earlier column generation algorithms, new dominance rules are developed to handle both time windows and maximum user ride time constraints in the subproblem.

Two pricing heuristics, operating on a reduced network and neglecting ride times, are applied before invoking the exact method. Qu and Bard (2014) present a branch-and-cut-and-price algorithm that takes into account configurable vehicles. Parragh et al. (2014) design a *branch-and-price* algorithm, embedding column generation into a branch-and-bound algorithm, for a problem variant with split requests and profits. In the subproblem, the positive reduced profit columns are identified using a classical labeling algorithm (Røpke and Cordeau, 2009). Hu and Chang (2015) apply the branch-and-price technique to a problem with time-dependent travel times and use a large neighborhood search metaheuristic, besides dynamic programming, to generate negative reduced cost routes.

2.3.1.4 Dynamic programming

Pure *dynamic programming* algorithms were mainly studied in early contributions on the single-vehicle problem (Psaraftis, 1980, 1983; Desrosiers et al., 1986). They may e.g. be used to optimize pairs of routes in a multi-vehicle problem (Ziliaskopoulos and Kozanidis, 2006; Lois et al., 2007) and to prove whether additional requests can feasibly be inserted into a route (Häme and Hakula, 2014). The advanced insertion algorithm of Häme (2011) considers all feasible insertion positions and preserves all resulting partial routes after each iteration, which is only possible in the presence of small time windows. A restricted variant selects partial routes which maximize the available time slack, such that future insertions are facilitated. Some authors also present heuristic variants in the multi-vehicle context (Häme and Hakula, 2013, 2014).

2.3.2 Approximate methods

Given that the DARP is an NP-hard problem (Baugh et al., 1998), the optimal solutions are generally not expected to be found in polynomial time. Consequently, problems of a realistic size are most often solved using approximate solution methods. Section 2.3.2.1 reviews two common types of classical heuristics, being insertion heuristics and cluster-first-route-second heuristics. Because of their inability to escape local optima, modern heuristics are usually incorporated into metaheuristic frameworks that are based on local search (Section 2.3.2.2) or population strategies (Section 2.3.2.3). Finally, Section 2.3.2.4 focuses on matheuristics, which represent a recent trend of embedding exact mathematical programming models into a (meta)heuristic algorithm or vice versa.

2.3.2.1 Classical heuristics

The sequential *insertion heuristic* of Jaw et al. (1986) sorts users according to their earliest pickup time. In each iteration, the first-sorted user is inserted at the best feasible position in the first route for which a feasible insertion is found. However, this position is not necessarily globally optimal. Diana and Dessouky (2004) address this myopic behavior in two ways. First, they use a

parallel insertion strategy in which the first-sorted user is inserted at the best feasible position over all routes. A seed request is chosen for each route, based on geographic decentralization. Second, a regret criterion prioritizes requests whose non-immediate insertion may harm the eventual solution quality. Toth and Vigo (1997) initialize routes with a difficult request in terms of user requirements and spatial or temporal situation. Furthermore, they introduce a dynamic component which takes into account the decentralization from initial requests in other routes. Generally, the insertion order of the remaining users may be tailored to the specific problem context (Wong and Bell, 2006) or it may be based on time windows or randomness. If requests cannot be inserted because of earlier myopic assignments, time-overlapping users may be relocated to another route (Luo and Schonfeld, 2007; Rubinstein et al., 2012). Insertion heuristics are particularly useful in a dynamic context, as they can add new requests without recomputing the entire solution (Coslovich et al., 2006). The rolling time horizon approach may be used (Luo and Schonfeld, 2011) and fuzzy logic approaches (Teodorovic and Radivojevic, 2000; Maalouf et al., 2014) may account for multiple objectives.

A *cluster-first route-second* heuristic solves the problem in two separate phases. Ioachim et al. (1995) first use column generation to construct clusters, solving a set partitioning master problem. In the subproblem, being a constrained pickup and delivery shortest path problem, clusters with a minimal reduced cost are generated using dynamic programming in a simplified network that only considers arcs between similar requests. In a second phase, a similar column generation algorithm is applied to chain these clusters. Borndörfer et al. (1999) generate clusters by complete enumeration. Efficient feasible routes are selected heuristically and a branch-and-cut approach is used to solve the resulting set partitioning problem. A distinction is made between different vehicle types. Karabuk (2009) develops a nested column generation approach which integrates the clustering and routing decisions, rather than composing routes after the construction of the clusters.

2.3.2.2 Metaheuristics based on local search

A *tabu search* (TS) framework investigates moves which change a single solution attribute. In each iteration, the most improving or least deteriorating neighbor is selected, which allows to escape local optima. Cycling is avoided using a short-term memory, called tabu list. Cordeau and Laporte (2003) iteratively relocate a single request to another route using the critical vertex rule, which first inserts the node having the tightest time window. Infeasible intermediate solutions and long-term repetitive insertions are penalized in an evaluation function with dynamic weights. An aspiration criterion allows tabu moves which improve the best-found solution containing that tabu attribute. Additional intra-route local search is performed after a given number of iterations. Various authors (Melachrinoudis et al., 2007; Beaudry et al., 2010; Ho and Haugland, 2011; Paquette et al., 2013;

Detti et al., 2016) apply a comparable tabu search strategy, often adapted to a richer problem context. Attanasio et al. (2004) present a parallel implementation in a dynamic context. Wolfler Calvo and Touati-Moungla (2011) and Kirchler and Wolfler Calvo (2013) suggest granular TS. Based on the cost of feasible request combinations, an assignment problem is solved to obtain clusters of close requests. A granular neighborhood is defined using reduced cost information. The authors require intermediate feasibility and ensure diversification through a variable length of the tabu list and an adaptive granular threshold. Except for short running times, however, classical TS is preferred because of its more extensive search area. Toth and Vigo (1997) implement tabu thresholding, which is a variant which requires no memory structure. Cycling is avoided by splitting the neighborhood into subsets and selecting the most improving or least deteriorating neighbor within a single subset. The relocate operator is supplemented with exchange and chain operators. Finally, TS constitutes the improvement phase in the hybrid greedy randomized adaptive search of Guerriero et al. (2013).

The *variable neighborhood search* (VNS) framework exploits the insight that neighborhoods are defined with respect to a particular operator. Thus, switching between multiple operators may allow an escape from local optima. Parragh et al. (2010b) alternate between three inter-route operator types with different sizes, being the exchange operator, chain operator and zero-split operator. The latter selects a natural sequence of nodes, meaning that the vehicle is empty at the start and at the end of this sequence. As in Cordeau and Laporte (2003), a dynamic evaluation function penalizes infeasible intermediate solutions. Promising solutions are subjected to additional (intra-route) local search, which is a common intensification technique in (meta)heuristic approaches. Additional local search is also applied to a small percentage of other solutions to foster diversification. Muelas et al. (2013) propose additional relocate and exchange neighborhoods and sort their operators according to past performance. Their distributed VNS variant (Muelas et al., 2015) solves large-scale applications containing up to 16,000 requests by creating independent and equally sized subsets. Detti et al. (2016) pay specific attention to operators that perturbate slightly infeasible solutions in order to attain feasibility. Schilde et al. (2011, 2014) adapt VNS to a problem variant with stochastic requests, using a simple average indicator to handle the different future scenarios. In a bi-objective context, Parragh et al. (2009) apply iterated VNS to optimize a weighted-sum objective function.

The *large neighborhood search* (LNS) framework removes a considerable percentage of requests, after which an attempt is made to insert these requests differently. Typical destroy (ruin) operators (Schrimpf et al., 2000; Røpke and Pisinger, 2006) are random removal, worst removal, sequential removal, route removal and related removal, also referred to as Shaw (Shaw, 1998) removal. Common repair (recreate) operators (Schrimpf et al., 2000; Røpke and Pisinger, 2006) are random insertion, greedy insertion, k-regret insertion, most-constrained-first insertion and space-time-related inser-

tion. Operators are usually tuned and evaluated for a specific problem variant (Häll and Peterson, 2013) and may even be tailored to this context. For example, Masson et al. (2014) develop operators which take into account transfers, whereas Lehuédé et al. (2013) exploit the specific context of common destination locations and Braekers and Kovacs (2016) create a multi-period variant with driver consistency. Ritzinger et al. (2016) propose block-based operators, which remove zero-split sequences (Parragh et al., 2010b). These sequences are either reinserted as a whole or decomposed into individual requests. In a multi-objective context, Lehuédé et al. (2013) include parameterized repair operators, which reduce the computational efforts by focusing on a single objective. Furthermore, the basic LNS framework is often extended. Qu and Bard (2013), Li et al. (2016), Masmoudi et al. (2016) and Masson et al. (2014) implement adaptive LNS (Røpke and Pisinger, 2006), in which the probability of selecting an operator depends on its past performance. The latter authors also suggest a multi-start strategy and solve a maximum diversity problem to select initial solutions.

The *deterministic annealing* (DA) framework accepts deteriorations smaller than a gradually lowered threshold. Braekers et al. (2014) apply relocate, exchange, 2-opt*, r-4-opt and route elimination operators in a random order. They obtain the best-known pure local search solutions for the common benchmark instances of the single-depot and multi-depot problem (see Section 2.3.3). *Simulated annealing* (SA) accepts deteriorating solutions according to a stochastic process. It is usually combined with characteristics from other metaheuristic frameworks. For example, Baugh et al. (1998) and Mauri and Lorena (2006) apply an acceptance criterion based on SA, but also encourage diversification by integrating a tabu list and three different neighborhoods (intra-route relocation, inter-route relocation and exchange), respectively. Masmoudi et al. (2016) integrate a population-based component into DA and SA by means of a hybrid bee algorithm, which outperforms the results of Braekers et al. (2014) for the multi-depot DARP.

Similarly, *evolutionary local search* is a hybridization of the *iterated local search* metaheuristic, starting each iteration with a population of perturbated solutions. All are subjected to local search and the best result serves as the new incumbent solution. Chassaing et al. (2016) invoke six local search operators, having dynamically adapted activation probabilities that stimulate convergence. Their clone detection principle avoids that previously obtained solutions are visited too often. This approach improves the results of Braekers et al. (2014) for some instances of the standard DARP.

2.3.2.3 Metaheuristics based on population search

Genetic algorithms (GA) combine desirable characteristics of two parent solutions, making use of a specific encoding scheme and crossover technique. Grouping GA (Rekiek et al., 2006; Jørgensen

et al., 2007; Hanne et al., 2009) use an encoding scheme which clusters users into routes and requires a separate heuristic to determine the order of service within a route. Jørgensen et al. (2007) perform crossovers by selecting random user-vehicle combinations from both parents. Rekiek et al. (2006) insert clusters from the first parent into the second and correct for double requests. Other GA (Cubillos et al., 2007, 2009; Chevrier et al., 2012; Atahran et al., 2014; Zhang et al., 2015) propose encoding schemes in which both the clustering of the users and their order in the route is determined. However, such representations make it even more difficult to define tailored crossover operators that strategically aim at combining desirable solution characteristics. General crossovers are usually invoked, such as uniform crossover, one-point crossover, two-point crossover and partially matched crossover. To reduce the number of infeasible solutions found, Cubillos et al. (2009) propose preprocessing techniques, a list of incompatible users and the use of a precedence table. Zhang et al. (2015), who consider a multi-trip problem variant, combine trips (segments starting and ending at the depot) from both parents. When trips cannot be inserted, the remaining requests are added individually using k-regret insertion. All aforementioned authors include sporadic mutations which foster diversification. The parent selection may be based on a tournament principle or (in multi-objective approaches) crowding distance.

Hybrid GA, also referred to as memetic algorithms, extend an evolutionary framework with local search. For example, Chevrier et al. (2012) add a 2-opt operator and Zhang et al. (2015) apply relocate and exchange operators after each crossover. The highly effective hybrid GA of Masmoudi et al. (2017) includes two crossover operators, based on a sequencing strategy using a one-point crossover and a merge strategy selecting individual genes from the parent solutions, an additional local search phase and four different mutation operators. This approach outperforms the results of both Chassaing et al. (2016) and Braekers et al. (2014).

Parragh et al. (2009) invoke *path relinking* (PR) to intensify their approximation of the Pareto frontier in a bi-objective problem context. PR is a stepwise conversion between two solutions in the solution set. Combining desirable characteristics of both, new non-comparable or even dominating solutions might be discovered. Mapping the routes from both solutions is done randomly, according to the number of identical requests or according to the number of transformations needed. In the actual procedure, a random weighted-sum function determines the most appropriate move.

2.3.2.4 Matheuristics

Hybrid solution approaches embedding mathematical programming models in a (meta)heuristic algorithm are defined as *matheuristics*. Archetti and Speranza (2014) describe them as an emerging

trend in vehicle routing. Matheuristics combine the advantages of exploiting the specific problem structure and using high-performance mathematical solvers. They can be classified into three classes, being decomposition approaches, improvement heuristics and column generation approaches (Archetti and Speranza, 2014). Despite their excellent performances, matheuristics are relatively scarce in the literature on the DARP.

An example of a matheuristic *decomposition approach* can be found in Oberscheider and Hirsch (2016). They apply a cluster-first route-second strategy in which the clustering is performed using an exact algorithm. First, all feasible request combinations are identified and the requests are optimally combined by solving a set partitioning problem. Second, these combinations are used as an input to a TS metaheuristic to solve the routing problem.

The class of *improvement heuristics* is represented by hybrid large neighborhood search (hybrid LNS) algorithms. Jain and Van Hentenryck (2011) apply constraint programming to complete the partial solutions in which users have been removed. The insertion order is based on the remaining number of feasible vehicles and insertion points, as well as on the insertion cost. At the end of each iteration, non-improving complete solutions may also be accepted. Ritzinger et al. (2016) perform insertions using a dynamic programming algorithm, adopting a giant tour representation to deal with multiple vehicles. A restricted version selects promising states throughout the procedure.

In the class of *column generation* (CG) approaches, Parragh and Schmid (2013) identify feasible routes with a negative reduced cost in a non-exhaustive way. Variable neighborhood search (VNS) is performed on individual routes. After a given number of CG iterations, large neighborhood search (LNS) is applied on the complete solution, invoking classical operators (Røpke and Pisinger, 2006). All newly generated routes may be used in a subsequent execution of the CG procedure. Crawford et al. (2007a,b) implement hybrid ant colony optimization (hybrid ACO) to select columns which represent clusters of requests. Guided by a short-term memory (pheromone trail) regarding the profitability of columns, ants select columns until a complete solution is reached. Constraint programming is integrated to reduce the number of possible additional columns through the identification of constraint violations.

Apart from the aforementioned classification of matheuristics, solution approaches may combine full versions of exact and (meta)heuristic algorithms. This is particularly beneficial in a dynamic context. Berbeglia et al. (2012) combine tabu search (TS) (Cordeau and Laporte, 2003) and constraint programming (CP) (Berbeglia et al., 2011) to determine whether a newly received request can be accepted. It is rejected if CP proves that the insertion is infeasible or both methods cannot

find a feasible insertion within a certain time frame. As CP performs well in the presence of tight constraints and TS easily discovers insertions if constraints are weak, the strengths of both methods are combined. After the insertion of a request, TS keeps running continuously to improve the unexecuted part of the solution.

2.3.3 Benchmark data

Since many contributions in the literature are based on practical problems, algorithms are often tested on case-specific *real-life instances*. Such an approach is useful to demonstrate the practical applicability of solution methods, but causes difficulties in comparing their efficiency. Consequently, two sets of *artificial benchmark data* have been proposed to perform computational tests. This section discusses their characteristics and the efficiency of various algorithms in solving them.

The data set introduced by Cordeau (2006) and extended by Røpke et al. (2007) consists of 42 instances, including 16 to 96 requests. An instance is structured as follows. Half of all requests are outbound, whereas the others represent inbound requests. Origins and destinations are randomly and independently generated in a square region $[-10, 10]^2$. The depot is located in the center. The data set can be subdivided into two groups of 21 instances. The a-instances consider a single user to be picked up or delivered at each location, whereas the b-instances assume that users may travel in group. A uniform distribution is used to determine the number of users picked up or delivered at each location and the corresponding service duration. Time windows of 15 minutes are created for destinations of an outbound request, as well as for origins of an inbound request. Travel times equal Euclidean distances. A fixed maximum user ride time of 30 minutes and a fixed service duration of 3 minutes are taken into account. Vehicle capacity is fixed at 3 customers. The maximum route duration is dependent on the instance and ranges from 240 to 720 minutes. Several authors extended this data set to richer problem variants, including heterogeneous customers and multiple vehicle types (Parragh, 2011), multiple depots (Braekers et al., 2014) and breaks (Masmoudi et al., 2016). Berbeglia et al. (2011, 2012) assume that part of the user requests are dynamically revealed. Parragh et al. (2014) use the b-instances to test the effect of split requests.

The data set of Cordeau and Laporte (2003) consists of 20 instances, containing between 24 and 144 requests. The requests are generated in a comparable manner as described before, but clustered around a number of seed points. The coordinates of the depot are the averages over those of the seed points. A single user is picked up or delivered at each location. Half of the instances have time windows of 30 minutes, whereas wider time windows of 60 minutes are defined for the other half. A fixed maximum user ride time of 90 minutes and a fixed service duration of 10 minutes are imposed.

Vehicle capacity is fixed at 6 customers and the maximum route duration equals 480 minutes for all instances. Richer problem characteristics have also been introduced into this data set, such as heterogeneity, multiple depots and breaks (Masmoudi et al., 2016) and the combination of parcels and people (Li et al., 2016). Ho and Haugland (2011) define probabilities with which certain requests may be submitted and Masson et al. (2014) extend the instances with one or several transfer points.

Exact solution approaches are usually tested on the data of Røpke et al. (2007). To date, the most efficient results have been obtained by the branch-and-cut-and-price method of Gschwind and Irnich (2014), solving all instances to optimality. Approximate solution methods are often tested on both artificial data sets. State-of-the-art heuristics are the hybrid genetic algorithm of Masmoudi et al. (2017) and the evolutionary local search of Chassaing et al. (2016). Note that such efficient combinations of local search and population characteristics are a recent development. Other good results have been obtained by solution methods based on pure local search, such as the deterministic annealing metaheuristic of Braekers et al. (2014).

Although the artificial data sets discussed in this section have widely been applied to assess the performance of solution algorithms, their effectiveness of reflecting realistic systems may be questioned. Two particular characteristics are quite unlikely in practice. First, inbound and outbound trips are not paired, whereas most real-life users request a round trip on the same day. Second, the maximum ride time is equal for all users and does not depend on the distance traveled. To address these and other shortcomings, Chapter 6 of this thesis introduces a new, larger artificial data set including additional real-life characteristics.

2.3.4 Scheduling procedures

This section compares various procedures for solving the scheduling subproblem, which consists of determining the start time of the service at each node in a route, given all time-related constraints and the order by which the nodes are visited. Composing a correct time schedule is an essential condition to prove the feasibility of a route. Most of the scheduling procedures discussed below have a general applicability, rather than being restricted to a specific problem variant or solution method.

Most frequently invoked is the eight-step scheduling procedure of Cordeau and Laporte (2003), adopting the forward time slack principle (Savelsbergh, 1992) to successively minimize violations of time windows, route duration and violations of maximum user ride time. As the last two phases do not influence violations in the previous phase(s), this procedure may also be used in algorithms that preserve feasibility at all times. Parragh et al. (2009) introduce a more conservative forward time

slack computation in order to approximate the minimal total user ride time, even though this modification may cause incorrect infeasibility declarations. Both procedures are discussed extensively in Chapter 5, presenting a new scheduling heuristic to minimize the total user ride time. Whereas the forward time slack principle causes $O(n^2)$ complexity, Hunsaker and Savelsbergh (2002) argue that a feasible schedule with maximum waiting times can be found in linear time. Starting from the earliest schedule without ride time constraints, violated ride times are corrected by postponing pickups and shifts of waiting time. Tang et al. (2010) and Haugland and Ho (2010) detect incorrect infeasibility declarations due to the fact that delivery times remain unchanged. Tang et al. (2010) propose a correction increasing the complexity to $O(n^2)$, although this worst-case scenario rarely occurs (Gschwind, 2015). Haugland and Ho (2010) propose a correction with $O(n\ log(n))$ complexity. Firat and Woeginger (2011) solve a shortest path problem on a vertex-weighted interval graph. Their formulation of linear inequalities uses a single decision variable, indicating waiting time before leaving a node. All constraints are expressed as a difference constraint system. A feasible schedule is obtained with linear complexity, as confirmed by computational tests in Chassaing et al. (2016).

Finally, some specific contributions to the scheduling subproblem are presented. The heuristic of Jaw et al. (1986) divides routes into schedule blocks, being continuous active periods. It relies on the simplifying assumption that no waiting time is scheduled while users are aboard the vehicle. This approach is particularly useful with nonconstant travel times (Fu, 2002; Schilde et al., 2014), as blocks can be scheduled independently. Braekers et al. (2014) include preliminary checks which may prove the infeasibility of a route before any scheduling procedure is executed. They compute the earliest arrival and latest departure in each node based on travel times, time windows and service durations, not taking into account user ride times. Finally, specific procedures have been presented for certain problem variants. Masson et al. (2012) consider route interdependencies due to transfers and show that evaluating feasibility is equivalent to checking the consistency of a simple temporal problem. Zhang et al. (2015) deal with scheduling for the multi-trip problem variant. Garaix et al. (2010) approach scheduling as a multi-objective problem in which nodes are connected by multiple Pareto-optimal paths. For example, costs and travel times should be balanced when congestion or multimodal transportation are taken into account. Gschwind (2015) generalizes the forward time slack principle and two of the aforementioned scheduling procedures (Tang et al., 2010; Firat and Woeginger, 2011) to a problem variant with both minimum and maximum ride time constraints.

2.3.5 Discussion and research opportunities

Over three decades of research on the DARP has resulted in the development of numerous solution methods. Initially, exact solutions were obtained through dynamic programming. The use of

this technique is restricted to small instances, preferably involving a single vehicle. During the past decade, branch-and-cut approaches on efficient two-index formulations have been proposed (Røpke et al., 2007). Formulations and valid inequalities have been extended to various rich problem variants (Braekers et al., 2014; Liu et al., 2015). Because of their efficiency, column generation and branch-(and-cut)-and-price (Gschwind and Irnich, 2014) approaches have recently gained attention. As the latter approach currently obtains the fastest exact solutions for artificial instances with up to 96 requests and 8 vehicles, its application to richer problem variants and larger instances is, at least from an academic perspective, an important direction for future research. From the perspective of service providers, being able to compute the optimal solution to their routing problem may not be the main concern. Service providers may rather prefer a fast and efficient solution software that is able to flexibly deal with complicated real-life conditions, including dynamic events (e.g. an additional request). Approximate methods with tailored search operators may be more suitable to meet these expectations.

Anyway, the NP-hardness of the DARP explains why in general, larger-sized practical applications cannot be solved exactly within reasonable computation time, which forces them to rely on a range of ever-improving approximate solution methods. Initially, a variety of classical heuristics was presented in the literature. It mainly concerned insertion heuristics and cluster-first-route-second heuristics focusing on the standard problem characteristics. Insertion heuristics are also particularly useful to update solutions in a dynamic setting. The further development of approximate solution methods was characterized by two major shifts of attention. The first one occurred when Cordeau and Laporte (2003) proposed an efficient implementation of tabu search. Ever since, authors have designed metaheuristics for various problem variants. However, the efficiency of such a metaheuristic framework is not an intrinsic feature. Numerous design choices, such as the representation of the solution, the definition of neighborhoods and the balance between intensification and diversification should be tailored to the specific problem context. Particularly metaheuristics based on local search, such as variable neighborhood search (Parragh et al., 2010b) and deterministic annealing (Braekers et al., 2014), used to obtain good results within realistic computation times. The common feature of all state-of-the-art algorithms in this category is the fact that, regardless of the selected metaheuristic framework, they invoke a combination of well-known neighborhood types. Many operators have proved their usefulness for solving a wide range of vehicle routing problems. Frequent examples are operators performing small relocations or exchanges of requests (Savelsbergh, 1992), operators reconnecting natural sequences or other segments from different routes in an alternative manner (Lin and Kernighan, 1973; Parragh et al., 2010b) and operators destroying complete routes or natural sequences (Braekers et al., 2014; Parragh et al., 2010b). Large neighborhood searches are typically based on the algorithm of Røpke and Pisinger (2006) for the PDPTW, invoking random,

worst and related removal operators in combination with random order, greedy and regret insertion operators. Solution methods for rich problem variants benefit from additional operators that take advantage of the specific structure, which is demonstrated by the transfer-related destroy and repair operators in Masson et al. (2014). The further development of such operator types for different rich problem variants may be addressed in future work. Urra et al. (2015) suggest using a hyperheuristic to determine the best high-level search strategy, e.g. with respect to operator choices or acceptance strategies. The second, more recent shift of attention is the integration of multiple types of solution approaches. Several state-of-the-art metaheuristics benefit from integrating a population concept into local search approaches (Chassaing et al., 2016; Masmoudi et al., 2016) or vice versa (Masmoudi et al., 2017). Implementations of pure population-based metaheuristics proved less effective in the past, since it is difficult to define crossover operators which combine desirable characteristics of two parent solutions in a strategic manner. Yet, combining multiple types of crossovers and mutations in a genetic algorithm with additional local search operators can be highly effective (Masmoudi et al., 2017). Moreover, hybridizations of exact and approximate solution methods have been proposed in the literature. Results of the first implementations, such as the hybrid column generation approach of Parragh and Schmid (2013), show that this is a promising, relatively unexplored field which offers interesting opportunities for future research.

2.4 Conclusion

Recent literature on dial-a-problems takes into account increasingly more real-life problem characteristics that generalize the standard definition. This chapter presents a thorough classification of existing problem variants. Particularly variants involving heterogeneity, more complicated routing properties and stochastic or dynamic information have gained interest in recent years. Although this enhances the practical applicability of solution methods, surveys among users still reveal an insufficient match with their actual expectations. This may be due to the fact that authors fail to combine real-life characteristics from different categories. From a managerial point of view, research should to a larger extent focus on analyzing the impact of particular problem characteristics on the tradeoff of the operational efficiency and service quality. In addition, there is a recent trend of integrating the vehicle routing aspect of dial-a-ride systems with related problems (e.g. healthcare scheduling, parcel deliveries, ride sharing) in order to optimize the overall operational costs.

From an algorithmic point of view, the wide range of solution methods proposed in the literature is reviewed in a structured manner. In the past decade, exact solutions for relatively small artificial instances have been obtained using branch-and-cut, column generation or branch-(and-cut)-and-price approaches. Due to the NP-hardness of the problem, approximate methods have been invoked

to solve larger problems with diverse characteristics. Different types of metaheuristics based on local search have achieved good results within realistic computation times. Their performance on rich problem variants may still be improved using operators that exploit the specific problem structure. In that respect, it may also be useful to investigate whether certain search operators and algorithmic components perform better in a particular problem context. Furthermore, hybridization is a recent trend to improve the performance of solution methods. For example, combinations of local search and population strategies, as well as combinations of exact and approximate methods constitute a promising and relatively unexplored research area with a view to rapidly obtaining efficient solutions.

Solution methods are often tested on real-life instances. To compare them among each other, two sets of artificial benchmark data are commonly used for a limited number of problem variants. For other variants, either no artificial benchmark data exist or data have been created to the sole discretion of authors. A more general applicability is desirable to compare the solution techniques for all rich problem variants. In addition, the number of users and vehicles in the artificial instances is relatively small in comparison with real-life systems and some reasonable assumptions are ignored, such as the fact that users normally request corresponding inbound and outbound trips on the same day. In that respect, computational analyses should devote more attention to the robustness of the solution methods, verifying whether they produce equally efficient solutions if different operational characteristics apply.

Chapter 3

Operational effects of service level variations

— *Short summary* —

In the standard DARP, the operational costs are minimized subject to a minimum service quality that is enforced by service level requirements. Section 3.1 illustrates that little research has been conducted into the effect of service level variations on the operational costs incurred by the service provider. Section 3.2 explains how the present chapter aims to fill that gap. The evolution of operational costs is investigated for 78 combinations of two service level parameters, being the maximum deviation from a user's preference time and the relative maximum exceedance of a user's direct ride time. This sensitivity analysis is performed on common benchmark data from the literature, using the *deterministic annealing* metaheuristic algorithm of Braekers et al. (2014). Section 3.3 visualizes the resulting tradeoff between quality and costs. The effect of different operating circumstances is quantified, including the size of the service provider, the traffic conditions in the service area and the heterogeneity of the users. Service level variations are found to considerably impact the routing solutions in all cases, evidencing the relevance of the research questions addressed in thesis. Section 3.4 illustrates the financial consequences of these findings, which should encourage service providers to make informed decisions on the service level they offer. In that regard, Section 3.5 provides them with appropriate recommendations, taking into account the preferences and behavior of users. Note that this chapter is based on Molenbruch et al. (2017a).

3.1 Introduction

The standard DARP described in Chapter 2 includes two parameters having a direct influence on the service level experienced by the users. The first parameter is the width of the time windows at the pickup and delivery locations, which is determined by the allowed *deviation from the user's preference time*. The second parameter is the maximum user ride time. In this chapter, it will be computed for each individual user trip by delimiting the *relative exceedance of the corresponding direct ride time*. Paquette et al. (2009) confirm that both parameters are important quality criteria. Almost all solution methods in the literature, which are often based on the characteristics of a real-life system, include a possibility to limit the deviation from the user's preference time and the time spent aboard the vehicle. The customer survey by Paquette et al. (2012) even identifies narrowing the time windows as one of the frequently mentioned desires of users to improve service quality in a Canadian context. Ben-Akiva et al. (1996) find that shortening ride times significantly increases both the perceived service utility and the number of trips made by users. Nevertheless, surprisingly little research has been performed into the relationship between the tightness of those quality restrictions and the operational costs incurred by implementing them, with the exception of some simulation models. For example, Quadrifoglio et al. (2008) assess the effect of varying the width of the user's time window, generating demand distributions on the basis of a heterogeneous system in the Los Angeles County. An approximately linear relationship is found between the time window settings and five performance indicators, being fleet size, total distance traveled, empty mileage, passenger miles and idle time. Oberscheider and Hirsch (2016) investigate five different quality levels with respect to time windows, maximum ride times and exclusive transports for non-emergency ambulance services in Lower Austria. The first phase of their matheuristic solution approach performs an exact optimization of users that can be combined in the same vehicle, which allows them to conclude that stringent service level criteria increase the total operating time through a reduced number of possible request combinations.

The main contribution of this study is to increase service providers' awareness of the effect on operational costs caused by service level variations, considering different operating circumstances. The service level is defined as a combination of (i) a maximum deviation from the user's preference time and (ii) a relative maximum exceedance of the direct ride time. A pattern in the evolution of the operational costs is identified, considering 78 different quality scenarios in which both service level criteria are varied simultaneously. Having insight into this tradeoff between quality and costs enables service providers to make an informed decision on potential changes in the service level they offer, taking into account financial consequences. The results of the analysis depend on a number of operating circumstances, such as the size of the service provider, the traffic conditions in the service

area and the heterogeneity of the users. The operational costs in terms of the total distance traveled are computed using the *deterministic annealing* metaheuristic of Braekers et al. (2014).

The sensitivity analysis is performed on well-known data from the literature (Cordeau and Laporte, 2003; Cordeau, 2006), adopting the standard problem characteristics of Cordeau and Laporte (2003). The effect of the aforementioned operating circumstances is introduced by considering targeted modifications or extensions to these data. However, since these sets are artificial in nature, attention should be devoted to the interpretation of the results. The objective of this study is to draw conclusions on a conceptual level, illustrating which factors service providers need to consider when balancing service level and costs. In practice, a system's performance may be influenced by a number of other factors, not taken into account in these benchmark data. Therefore, the magnitude of the relative differences in costs mentioned throughout this study might differ in real-life systems. Nevertheless, the financial impact of the results for idealized circumstances is illustrated by means of an example, which intends to encourage service providers in any specific operating context to investigate their sensitivity for service level variations in a targeted manner.

3.2 Methodology

Each of the 78 quality scenarios considered in this study is a combination of a maximum deviation from the user's preference time and a relative exceedance of the direct ride time. For the first criterion, a realistic range of 13 different settings between 0 and 60 minutes is considered, with intervals of 5 minutes. For the second criterion, a realistic range of 6 different settings between 0% and 100% of the direct ride time is taken into account, with intervals of 20 pct points. All possible combinations of both criteria result in a total number of 78 different scenarios. For each of them, the deterministic annealing metaheuristic of Braekers et al. (2014) is applied to construct efficient vehicle routes and time schedules, minimizing operational costs. As in most solution approaches, it is assumed that operational costs vary linearly with the total distance traveled, measured in terms of Euclidean distances, even though this may not fully correspond to real-life systems. A detailed discussion on the solution method of Braekers et al. (2014) can be found in the original paper, but its main characteristics are summarized below.

A feasible initial solution is found by performing 1000 runs of a parallel insertion heuristic. The first time, users are ordered according to the lower bound of their pickup time window. They are sequentially inserted at the best possible position across all existing routes, such that the total distance traveled is minimized. Only if no feasible insertion is found, a new route is activated. For all following executions, users are treated in a random order. The best solution over all runs becomes

the initial solution. The actual deterministic annealing procedure seeks a feasible neighboring solution by means of a local search technique. Five runs are executed for every quality scenario. In each of the 350,000 iterations per run, the five operators discussed below are applied in random order. A solution proposal has to be feasible and is only accepted if the total distance traveled improves or deteriorates by less than a threshold value, which is lowered each time no improving solution is found. As soon as the threshold value becomes negative, it is reset to a random positive value equal to or smaller than its original value. After a number of iterations without discovering a new best solution, the search resumes from the current best one. The time feasibility of the routes is checked using the procedure of Cordeau and Laporte (2003), finding a feasible schedule whenever one exists.

Five different local search operators are involved. The *relocate inter-route operator* removes a single request from a randomly selected route and inserts it at the best possible position in another route. The *exchange inter-route operator* tries to exchange a request from a randomly selected route with another request in a different route. The pickup and delivery nodes of the first user are inserted in the second route at the same position as the second user's pickup and delivery nodes, whereas the second request is inserted optimally in the first route. The *2-opt inter-route operator* (Potvin and Rousseau, 1995) considers arcs which are traversed by an empty vehicle. Two such arcs from different routes are selected randomly and the resulting parts of the routes are connected crosswise. The reason why only the aforementioned selection of arcs is relevant, is the fact that a user's origin and destination need to be visited in the same route. The *r-4-opt intra-route operator* considers each set of four successive arcs in a randomly chosen route and tries to rearrange the intermediate nodes in the most efficient way. This is a simplification of the traditional 4-opt operator by Lin and Kernighan (1973), which does not require successive arcs. Lastly, the *eliminate operator* removes all users in a random route and attempts to reinsert them sequentially at the best possible position in another existing route, thereby reducing the required fleet size.

3.3 Experimental results

This study is implemented on the commonly used benchmark data sets of Cordeau and Laporte (2003) and Røpke et al. (2007). Some slight modifications were applied for this chapter only. The service duration for the instances of Cordeau and Laporte (2003) is changed from 10 to 3 minutes, since it seems unrealistic to assume that picking up or delivering a user requires half of the time needed to cross the entire service area. This modification will allow to observe the effect caused by the service duration constraint more clearly. It does not harm the representativeness of this study, which has no intention to compare multiple solution techniques. In addition, the constraints on the fleet size are omitted for both data sets. The required number of vehicles to solve an instance heavily

depends on the settings of the service level parameters. However, it is difficult to determine the most appropriate fleet size in advance. Since the analyses are made in terms of the total distance traveled in the different quality scenarios, they should not be biased by an overly restrictive choice of the fleet size. In other words, providing additional vehicles should not allow a further reduction in the total distance traveled. Therefore, the most accurate comparison of different scenarios can be made without imposing a maximum on the fleet size. Anyway, no unreasonably large number of vehicles will be activated in an efficient solution, since (1) a large number of connections to the depot would increase the total distance and (2) redundant routes can easily be removed by the *eliminate operator*.

The findings of this study will be presented in several subsections. First, the overall results are discussed in Section 3.3.1. A breakdown according to the size of the service providers is presented in Section 3.3.2. Different results are found when considering urban traffic conditions or heterogeneous customer types, as is reported in Sections 3.3.3 and 3.3.4, respectively. In order to facilitate the comparisons, the operational cost in a certain quality scenario, which is the average over the five runs of the deterministic annealing metaheuristic, is expressed as a percentage of the result in the corresponding baseline scenario[1]. This baseline scenario reflects a taxi service for which neither a deviation from the user's preference time, nor an exceedance of the direct ride time is allowed. In other words, it is an ideal scenario in which all users receive the highest level of service.

3.3.1 Overall results

In general, the operational costs can be expected to decrease whenever any of the two quality criteria is loosened. As a user is allowed to remain aboard the vehicle for a longer period of time or his preference time may be respected less strictly, there is an increasing probability that the total distance traveled can be reduced by combining or interweaving this request with another request. The objective of this subsection is to confirm this hypothesis and to identify the underlying pattern.

An overview of the results over all data sets, containing 62 instances, is presented in Figure 3.1. The axes in the horizontal plane represent the allowed deviation from the user's preference time and the allowed relative exceedance of the direct ride time. In the remainder of this chapter, a reference to a particular quality scenario is made as follows. The notation (20, 0.4) refers to the scenario in which the deviation from the user's preference time is at most equal to 20 minutes, whereas the exceedance of the direct ride time is limited to 40% of the direct ride time for the corresponding

[1]Very similar results are obtained if the best or worst result over five runs is used to determine the operational cost in each scenario. The variability of the objective values over the five runs is small, even for large instances. In addition, the corresponding baseline scenario is also based on the best or worst result over five runs.

trip. As explained before, the total distance is expressed as a percentage of the result in the baseline scenario. For example, the result in scenario (20, 0.4) shows that the total distance traveled by all vehicles equals 94.20% of the result in scenario (0, 0.0).

Figure 3.1 confirms the expectation that any loosening of a quality criterion consistently reduces the operational costs. Except for the range in which the allowed exceedance of the direct ride time is small, the graph exhibits a convex pattern in both directions, which indicates that the marginal benefits of consecutive reductions in the quality level become progressively smaller. In other words, reducing the service quality has a smaller cost effect when the quality level is already poor, indicating that there is no point in performing repeated blind quality reductions to obtain operational savings. For limited values of the allowed exceedance of the direct ride time, the effect of the service duration reshapes the graph into a concave pattern. As it takes some minutes to board or to leave the vehicle, it is useless to allow a user's direct ride time exceedance smaller than that amount of time. After all, the time margin created is not sufficient to pick up or deliver another user, implying that the solution will not be affected. If the allowed direct ride time exceedance was expressed as an absolute amount of time, this would result in a part of the graph being insensitive to the ride time criterion. However, it is expressed relative to the direct ride time. Consequently, the effect of the service time cannot be delineated exactly, although it clearly slows down the initial influence of increasing the maximum ride time allowed.

Figure 3.1 provides a tradeoff between service levels and operational costs. However, since the service level is expressed in terms of the maximum user inconvenience with regard to both criteria investigated, the actual service quality experienced by the users is not represented. For example, the fact that the deviation from the user's preference time can be at most 20 minutes in the scenario (20, 0.0) tells the provider nothing about the extent to which this inconvenience is actually exploited in the solution. Therefore, overviews of the *average* deviation from the user's preference time and the *average* exceedance of the direct ride time in the different scenarios are represented in Figures 3.2 and 3.3, respectively. With regard to the interpretation of these graphs, it should be noted that they are not the result of a minimization of user inconvenience, since no such incentive is included in the objective function. They represent the average user's real service experience in a certain quality scenario, given the current solution method.

When the maximum deviation from the user's preference time is increased, no convex pattern can be observed for the effect on the actual deviation experienced by an average user, contrary to the effect on the operational costs. Within the ranges investigated by this study, the relationship in Figure 3.2 is linear. As a result, service providers trying to cut their operational costs by increasing

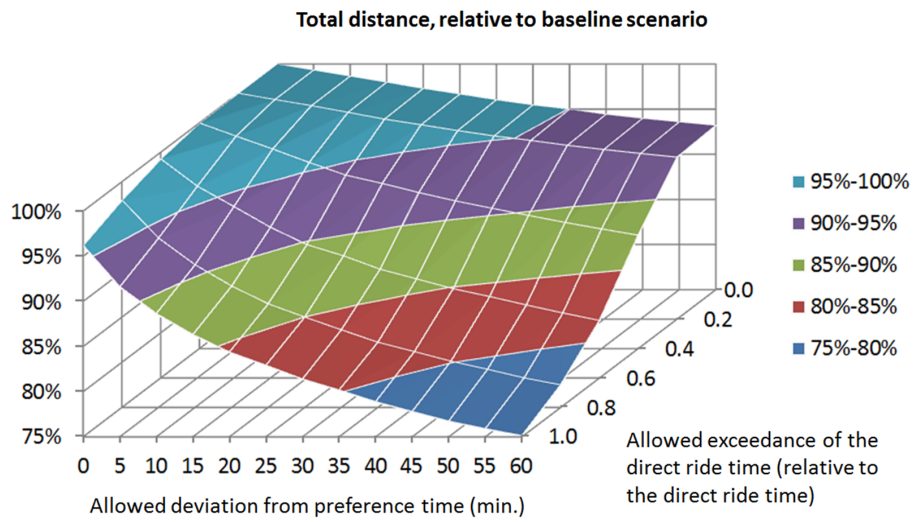

Figure 3.1: Total distance for a given quality scenario as a percentage of the result in the baseline scenario.

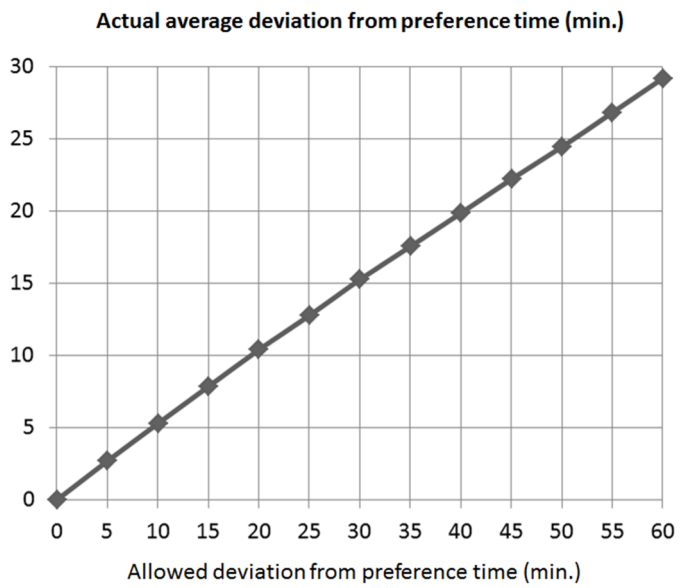

Figure 3.2: Actual average deviation from the user's preference time for a given maximum deviation, both expressed in minutes.

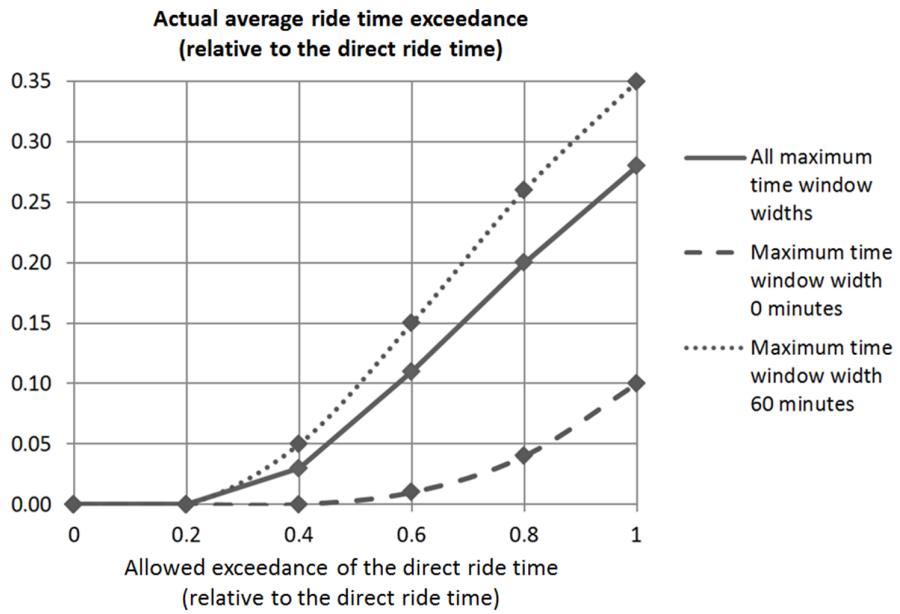

Figure 3.3: Actual average ride time exceedance for a given maximum ride time exceedance, both expressed relative to the direct ride time.

the maximum deviation from the user's preference time should consider their actions carefully. Particularly when their service level is already poor, the ratio between the marginal user inconvenience and the marginal operational savings will be undesirably large. More generally, the actual average deviation is situated at approximately half of the deviation allowed. Further analysis shows that the results in Figure 3.2 are completely insensitive to the associated maximum exceedance of the direct ride time. This means that exploiting the allowed deviation from the user's preference time does not become easier when the allowed ride time exceedance is large.

An analogous linear relationship applies for the effect of variations in the maximum ride time exceedance on the actual exceedance experienced by an average user, although the pattern is initially slowed down by the influence of the service duration, which was explained before. This relationship is shown by the solid line in Figure 3.3. Again, it can be concluded that any additional loosening of this quality criterion will cause relatively more marginal user inconvenience in comparison with the extra operational savings. Overall, only a small part of the allowed ride time exceedance is exploited.

It should be noted, however, that further analysis reveals that the results in Figure 3.3 are to

a certain extent influenced by the associated maximum deviation from the user's preference time, particularly when the maximum ride time exceedance is large. For example, the actual ride time exceedance experienced by an average user equals 0.10 times the direct ride time in the scenario (0, 1.0) and 0.35 times the direct ride time in the scenario (60, 1.0). This effect is not reflected in the average result over all six (..., 1.0) scenarios, being 0.28 times the direct ride time. Apparently, exploiting the allowed ride time exceedance becomes easier as the allowed deviation from the user's preference time increases. For illustration, the range of the actual average ride time exceedance is depicted by the dashed line and the dotted line in Figure 3.3, which correspond to a maximum time window width of 0 minutes and 60 minutes, respectively.

It can be assumed that these conclusions are generalizable to other systems having comparable characteristics, particularly with respect to the levels of service quality, the random user locations and the random time windows.

3.3.2 Classification according to the size of the service provider

Since larger service providers handle more requests than their smaller competitors, it is obvious that a variation in service level criteria has a larger absolute influence on the operational costs for larger providers. However, an increase in the number of requests is likely to exert a *non-linear effect* on the number of possible request combinations as well, meaning that larger service providers are also expected to experience a stronger effect in relative terms. The objective of this subsection is to verify this hypothesis.

For this purpose, the data is split up into four classes of about the same magnitude. Instances containing up to 32 requests are categorized as the smallest providers. Those including between 36 and 50 requests are small providers. Instances in which 56 to 72 requests are involved are classified as large providers, whereas the largest providers serve at least 80 requests. These four categories include 14, 16, 16 and 16 instances, respectively. Since all requests in the data set are generated randomly in a square area, it seems reasonable to base the classification of service providers only on the number of trips served, rather than considering the length of the trips involved.

The results in Table 3.1 illustrate that the relative effect of variations in service level criteria is indeed noticeably influenced by the size of the service provider. For the four categories defined above, the average operational costs over all 78 quality scenarios amount to respectively 92.61%, 91.47%, 90.11% and 88.93% of the result in the corresponding baseline scenario, compared to an average of 90.21% for the overall analysis in Section 3.3.1.

Instance category	**Requests per instance**	**Number of instances**	**Average distance (all 78 scenarios)**
Smallest	16-32	14	92.61%
Small	36-50	16	91.47%
Large	56-72	16	90.11%
Largest	80-144	16	88.93%
All		**62**	**90.21%**

Table 3.1: Average total distance over all 78 quality scenarios as a percentage of the result in the baseline scenario, split up according to the size of the service provider.

To test whether the differences between these categories are statistically significant, an *analysis of variance (ANOVA)* is performed. Its null hypothesis assumes that the population means of the four groups are equal ($\mu_{smallest} = \mu_{small} = \mu_{large} = \mu_{largest}$). As was shown in Table 3.1, each group is represented by a sample of independent instances. The ANOVA test verifies whether it is likely that all samples are drawn from the same distribution (Anderson et al., 2010). Certain conditions should be fulfilled to apply this test. First, variances should be equal for all groups, which can be verified using the *Levene's test*. For the four categories defined in this section, the Levene statistic equals 0.832 with a significance of 0.482 (> 0.05). Consequently, the hypothesis of homogeneity of variance is not rejected and it can safely be assumed that the condition of equal variances is fulfilled. A second condition requires that all variables are normally distributed or that the size of each sample is sufficiently large for the Central Limit Theorem to apply. The latter states that a large number (usually $n \geq 30$) of independent and identically distributed stochastic variables with a finite variance approximately follow a normal distribution. None of the four categories contains at least 30 instances, but the *Kolmogorov-Smirnov test with Lilliefors significance correction* does not reject the null hypothesis of a normal distribution for any group. The significances equal 0.200, 0.200, 0.065 and 0.200 (all > 0.05), respectively. In addition, the ANOVA test is known to remain accurate for small sample sizes if the group sizes are comparable. Based on these observations, it seems justified to use the ANOVA test in the given context. Note that more details on all statistical tests performed throughout this chapter can be found in Anderson et al. (2010).

The ANOVA test delivers an F-statistic of 13.313 and a significance of 0.000 (< 0.05), which implies that the null hypothesis of equal population means should be rejected. In other words, the categorical variable of provider size significantly influences the effect of service level variations on

Instance size	**Average distance (all 78 scenarios)**
500 requests	81.95%
750 requests	81.01%
1000 requests	80.36%

Table 3.2: Average total distance over all 78 quality scenarios as a percentage of the result in the baseline scenario, based on new large instances.

the operational costs and the population means cannot be the same. To find out which groups are statistically different from each other, is not advisable to perform the usual *Student t-test* for all six combinations of two groups, since this would considerably reduce the overall reliability of the test. A single t-test with 95% reliability causes a type I error in 5% of the cases, meaning that it wrongly rejects the null hypothesis of equal population means. As a result, the probability of not making at least one such error in six tests is only 0.95^6 (73.51%). The *Bonferroni test* corrects for this risk by testing each individual hypothesis at a higher reliability level, such that the sum of all type I error risks equals 5%. Applying this test, significant differences are found for the provider pairs 'smallest-large' ($0.001 < 0.05$), 'smallest-largest' ($0.000 < 0.05$) and 'small-largest' ($0.000 < 0.05$). For these pairs, the provider's size significantly influences the effect of service level variations on the operational costs. For other pairs, the null hypothesis of equal population means cannot be rejected.

Even though the size of the benchmark instances used is limited to 144 requests, dial-a-ride systems in practice may involve more requests. In order to obtain an understanding of the operational effects of service level variations for very large systems, some larger instances were generated. They have been created using the same clustering procedure (Cordeau et al., 1997) as the one invoked by Cordeau and Laporte (2003), assuming four seed points. Five instances have been generated, each containing 1000 requests having the same characteristics as the instances of Cordeau and Laporte (2003). The average total distance over all 78 quality scenarios, again expressed as a percentage of the result in the baseline scenario, is considered with (a) all 1000 requests, (b) 750 requests and (c) 500 requests taken into account. In (b) and (c), a random selection of requests is made, given that the number of outbound requests should still equal the number of inbound requests.

The average results over these five new instances are displayed in Table 3.2. Compared to the average distance percentage of 88.93% for the largest service providers in the data of Cordeau and Laporte (2003), the percentage of operational costs relative to the baseline scenario is consistently lower. This reconfirms that a larger number of requests increases the potential for request combina-

tions, which explains why the operational costs are more strongly affected by service level variations. However, the results in Table 3.2 clearly demonstrate that this effect does not continue infinitely as the size of the service provider increases. For example, increasing the number of requests from 500 to 1000 causes a reduction in the average distance percentage of merely 1.59 percentage points (from 81.95% to 80.36%). Once a substantial number of requests is involved, the number of potential request combinations seems sufficiently large and adding more requests reinforces the effect of service level variations to a much lesser extent.

3.3.3 Influence of traffic conditions

Several kinds of traffic conditions, such as traffic density, road infrastructure and traffic lights, may reduce the average vehicle speed in an urban environment when compared to a rural setting. The resulting extension of travel times reduces possible request combinations, thereby limiting the effect of variations in service level criteria. In order to verify this claim, the objective of this subsection is to mimic an urban environment. A large-scale socioeconomic survey organized by the Belgian government, reported in Verhetsel et al. (2009) (page 72), revealed that residents of the main urban areas experience an average car speed between 27.0 and 38.8 km/h, whereas this average speed is comprised between 56.2 and 69.4 km/h for residents of rural areas. Based on these observations, an urban environment is simulated through doubling all travel times in the network without altering the magnitude of the variations in service level criteria. This implies that a request's maximum ride time exceedance is still calculated as a function of the original direct ride time. From a conceptual point of view, remark that this approach does not intend to mimic congestion on a specific part of the service area's road network, either caused by rush hour traffic or by an accident. As discussed in Schilde et al. (2014), taking into account such traffic circumstances requires the introduction of stochastic travel times, as well as the possibility to avoid congestion by taking an alternative route. From a technical point of view, it should be remarked that this operation caused three requests to be unreachable from the depot without violating time constraints. For these instances only, a slight modification of the earliest departure from the depot was made.

The results in Figure 3.4 indicate that lowering the vehicle speed considerably limits the effect of variations in service level criteria on the operational costs. Although the pattern in this graph is comparable to Figure 3.1, the average operational cost over all 78 quality scenarios equals 92.07% of the operational cost in the baseline scenario, compared to an average of 90.21% for the original design. For illustration, a breakdown into different instance sizes is shown in Table 3.3, having the same interpretation as Table 3.1 in Section 3.3.2.

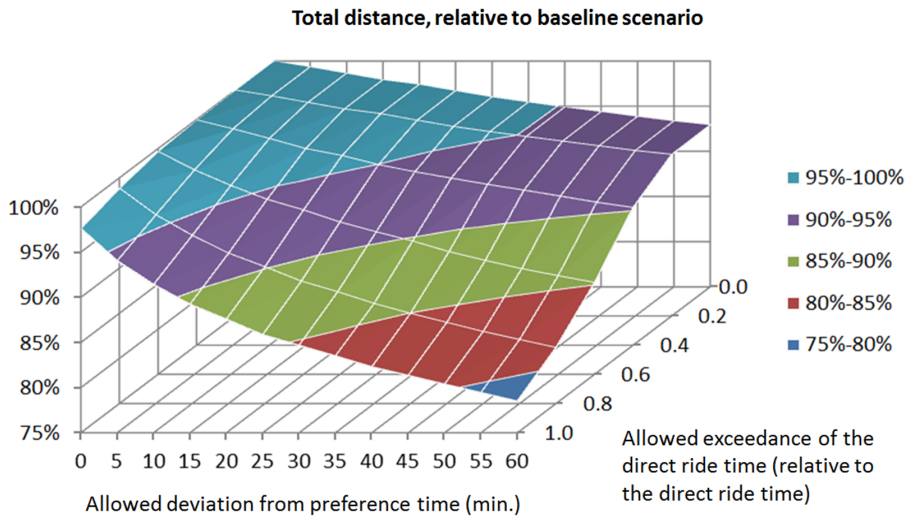

Figure 3.4: Total distance for a given quality scenario as a percentage of the result in the baseline scenario, taking into account a reduction of the vehicle speed.

Instance category	**Requests per instance**	**Number of instances**	**Average distance (all 78 scenarios)**
Smallest	16-32	14	94.33%
Small	36-50	16	93.09%
Large	56-72	16	92.11%
Largest	80-144	16	90.85%
All		**62**	**92.07%**

Table 3.3: Average total distance over all 78 quality scenarios as a percentage of the result in the baseline scenario, split up according to the size of the service provider and taking into account a reduction of the vehicle speed.

To test whether the influence of the traffic conditions is statistically significant, a *paired-samples t-test* is performed (Anderson et al., 2010). For every instance i, the average distance percentage before doubling the travel times (X_i) and the average distance percentage after doubling the travel times (Y_i) are compared. Consequently, the test consists of two paired samples, meaning that they are not drawn independently from each other. For all i, the corresponding costs are subtracted from each other and the resulting differences $Z_i = X_i - Y_i$ are investigated. The null hypothesis assumes that there is no significant difference between both settings, which would imply that $\mu_X = \mu_Y$ or equivalently $E(Z) = 0$. With $\bar{Z}$ being the average over all differences Z_i in the sample, s_Z the corresponding standard deviation and $n = 62$ the sample size, the t-value is computed as $t = \sqrt{n} * \bar{Z}/s_Z$. Note that this test actually requires that Z is distributed normally. However, it seems safe to make this assumption since the sample is large enough for the Central Limit Theorem to apply ($n \geq 30$). This is not contradicted by a significance of 0.200 (> 0.05) in the *Kolmogorov-Smirnov test with Lilliefors significance correction.*

For this paired-samples t-test, the resulting t-value equals -14.839, with a 2-tailed significance of 0.000. Thus, the value 0 is not included in the 95% confidence interval for Z, being [-0.020; -0.015]. The null hypothesis can be rejected, which implies that traffic conditions significantly influence the effect of service level variations on the operational costs.

3.3.4 Influence of heterogeneous users

The heterogeneous DARP (Parragh, 2011) considers additional requirements for the transportation of elderly and disabled, assuming multiple user categories which have different transportation needs. This is reflected by vehicles disposing of multiple resource types, which complicates the construction of a feasible solution. Since certain users may not be accommodated in particular types of seats, part of the vehicle capacity cannot be used as flexibly as in previous operating conditions. Due to this observation, possible request combinations may be more restricted in comparison with a homogeneous context, thereby reducing the effect of variations in service level criteria. The objective of this subsection is to confirm this hypothesis.

The tests in this subsection are based on the a-instances of Røpke et al. (2007) as adapted by Parragh (2011) in data set E. Although Parragh (2011) only presents instances containing up to 48 requests, this study considers heterogeneous variants of all 21 a-instances, containing up to 96 users (http://alpha.uhasselt.be/kris.braekers). Users fall into three categories, requesting transportation in a normal seat, in a wheelchair or on a stretcher with prefixed probabilities of 50%, 25% and 25%, respectively. Additionally, 10% of all users require the presence of an accompanying staff member.

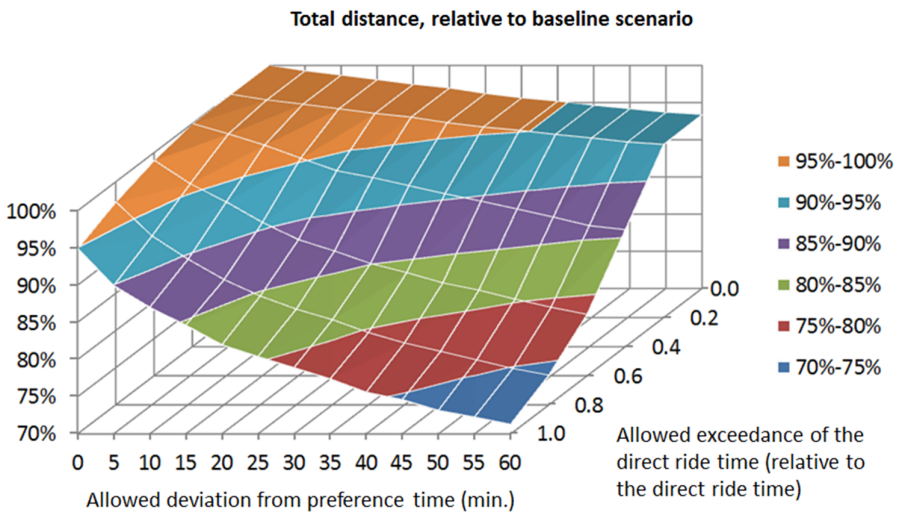

Figure 3.5: Total distance for a given quality scenario as a percentage of the result in the baseline scenario, only considering the *a*-instances of Røpke et al. (2007).

Vehicles are still identical and dispose of two staff seats, a normal seat, a wheelchair space and a stretcher. Except for the wheelchair category, upgrading conditions apply. This means that a user may be assigned to a lower resource type than the requested one. For example, users able to travel in a normal seat may be accommodated in a normal seat or on a stretcher, but not in a staff seat.

Since the heterogeneous instances are only based on the *a*-instances of Røpke et al. (2007), there is no use in comparing the results in this subsection to those in Figure 3.1, which takes into account all data sets. An analogous graph is first constructed for the *a*-instances of Røpke et al. (2007) only, as shown in Figure 3.5. Its pattern is comparable to Figure 3.1. The average distance traveled over all 78 scenarios amounts to 89.03% of the result in the baseline scenario (0, 0.0).

The influence of heterogeneity is clearly reflected in Figure 3.6. Although this graph exhibits the same pattern as before, the average distance traveled over all 78 scenarios is limited to only 91.10% of the result in the baseline scenario (0, 0.0). This confirms the assumption that heterogeneous users reduce the effect of variations in service level criteria. Note that the average gap of 2.07 percentage points is influenced by the fact that particularly scenarios imposing a small exceedance of the direct ride time do not produce any differences. After all, capacity restrictions of either homogeneous or heterogeneous resources play no role if users cannot be interwoven. The differences increase to 5.71

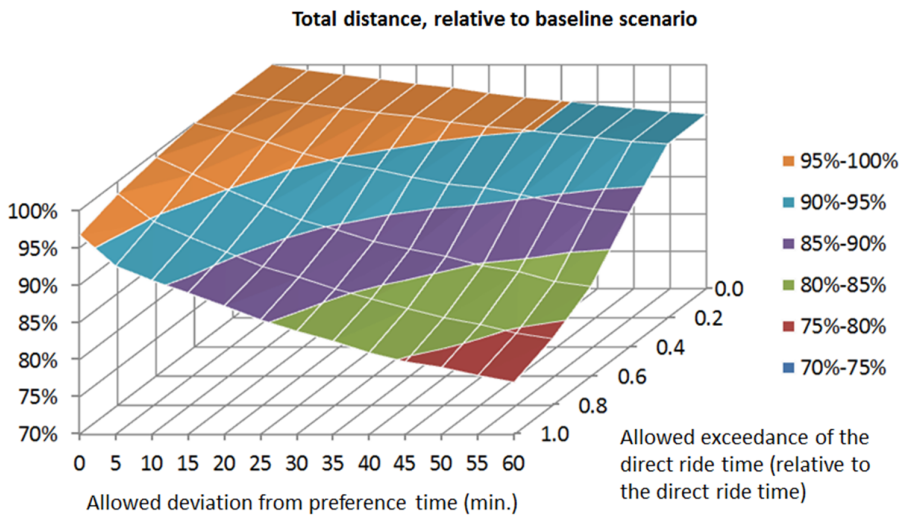

Figure 3.6: Total distance for a given quality scenario as a percentage of the result in the baseline scenario, taking into account heterogeneous user types.

percentage points in scenario (60, 1.0).

As in Section 3.3.3, the statistical significance of the results is illustrated using a *paired-samples t-test*. For each benchmark instance i, the average distance percentage assuming homogeneous users (X_i) and the average distance percentage assuming heterogeneous users (Y_i) are compared and the difference $Z_i = X_i - Y_i$ is investigated. The null hypothesis assumes that there is no significant difference between both settings, such that $E(Z) = 0$. Since only 21 benchmark instances are used, concerns may arise regarding the accuracy of the test. It actually requires that Z is normally distributed or that the sample size is large enough for the Central Limit Theorem to apply ($n \geq 30$). Although this condition is not satisfied, the *Kolmogorov-Smirnov test with Lilliefors significance correction* does not reject its null hypothesis that Z is normally distributed. The significance equals 0.200 (> 0.05). Hence, the results of the test are presented for the information of the reader.

The paired-samples t-test delivers a t-value of -9.945, a 2-tailed significance of 0.000, as well as a 95% confidence interval for Z being [-0.025; -0.017]. This indicates that the heterogeneous user types significantly influence the effect of service level variations on the operational costs.

3.4 Illustration of financial consequences

In this section, an illustration is made of the financial consequences and the quality impact that different policies can bring about for an average service provider in the data sets employed by this study. To this end, the operational context proposed by Cordeau and Laporte (2003) and Røpke et al. (2007) is interpreted as a real-life system. The distances are expressed in kilometers, resulting in a square network measuring 20 by 20 km. The travel times are interpreted in minutes, leading to an average speed of 60 km/h in normal traffic circumstances and 30 km/h in an urban environment. These average speed values might seem rather high at first sight, but it should be noted that stops for pickups or deliveries are not included, since they are modeled by the service duration. A set of diverse scenarios, being (0, 0.0), (15, 0.4), (30, 0.4), (30, 0.6), (45, 0.6) and (60, 1.0), is arbitrarily selected for this illustration.

The differences between these six scenarios are expressed in monetary terms. The total cost in a certain scenario is the sum of the vehicle costs (e.g. fuel) and the wages of the drivers. For the first aspect, a vehicle cost of €0.30 per kilometer is assumed, which is a realistic value for the use of a typical passenger car. For example, this corresponds to the compensation obtained by Flemish volunteers who deploy their personal car to transport people being unable to use regular transportation modes (Minder Mobielen Centrale). Therefore, it is certainly no overestimation for the usually well-equipped vans deployed by dial-a-ride service providers. For the second aspect, the wage paid to a driver is estimated at a conservative amount of €25 per hour, including taxes. Assuming that the average work duration for each driver corresponds to a common eight-hours shift[2], a daily wage cost of €200 per driver (= 8∗€25) is taken into account. Since both costs are based on conservative estimates, the resulting differences between scenarios will certainly not overestimate the real effect of service level variations.

The results are shown in Tables 3.4 and 3.5. The former assumes a rural environment with an average speed of 60 km/h, whereas the latter considers an urban environment with an average speed of 30 km/h. In both tables, the second column represents the operational costs incurred during a single working day for an average provider from the overall data set. For example, an average rural provider whose service corresponds to the scenario (0, 0) faces a total distance of 826.87 km and a required fleet size of 11.14, as is shown in the fourth and the fifth column. Multiplied by the corresponding costs, this results in a daily operational cost of €2476.06 (= 826.87∗€0.30+11.14∗€200).

[2]Since the demand of the users in this artificial data set is not clustered in time, implying that no peak and off-peak hours can be distinguished, it is assumed that only full-time drivers are employed and their shifts cannot be split. In contrast, the real-life case study that will be discussed in Chapter 6 considers part-time drivers and split shifts in order to account for differences in demand throughout the day.

Scenario	Daily cost	Daily savings	Required fleet	Total distance	Deviation pref. time	Excess ride time
(0, 0.0)	€2476.06	-	11.14	826.87	0.00	0.00
(15, 0.4)	€2061.14	€414.92	9.12	790.45	7.87	0.02
(30, 0.4)	€1806.87	€669.19	7.89	762.90	15.22	0.03
(30, 0.6)	€1709.89	€766.17	7.47	719.62	15.34	0.12
(45, 0.6)	€1536.38	€939.68	6.64	694.59	22.00	0.14
(60, 1.0)	€1280.34	€1195.72	5.47	621.14	28.72	0.35

Table 3.4: Comparison of different service levels in rural traffic circumstances.

Scenario	Daily cost	Daily savings	Required fleet	Total distance	Deviation pref. time	Excess ride time
(0, 0.0)	€2799.04	-	12.73	843.45	0.00	0.00
(15, 0.4)	€2412.82	€386.22	10.84	816.08	7.78	0.01
(30, 0.4)	€2122.32	€676.72	9.42	794.40	14.92	0.02
(30, 0.6)	€2073.67	€725.37	9.22	765.58	15.15	0.07
(45, 0.6)	€1875.66	€923.38	8.27	738.88	22.39	0.09
(60, 1.0)	€1550.64	€1248.40	6.76	662.15	28.99	0.29

Table 3.5: Comparison of different service levels in urban traffic circumstances.

When this average provider changes the service level from scenario (0, 0.0) to scenario (15, 0.4), the total distance reduces to 790.45 km, which corresponds to the relative value of 95.60% in Table 3.1, and the required fleet size reduces to 9.12. Multiplied by the corresponding costs, a total operational cost of €2061.14 is obtained. In the third column, the monetary savings in each scenario are computed with respect to the baseline scenario (0, 0.0).

As a side remark, it should be repeated that indications of the required fleet size do not result from a minimization of the fleet size. One of the deterministic annealing operators aims at reducing the number of vehicles used, but no explicit incentive is included in the objective function. Rather, these results represent the actual number of vehicles an average service provider would need in a certain quality scenario, given the current solution approach. This information is useful to estimate the required number of drivers and, even though the number of vehicles may be fixed in the short run, to make medium-term tactical decisions on the required fleet size.

As shown earlier by Figures 3.2 and 3.3, savings obtained by loosening quality criteria go hand in hand with an increase in the inconvenience experienced by an average user. The actual average deviation from the user's preference time and the actual average excess ride time (expressed as a multiple of the direct ride time) for the scenarios investigated here are repeated in the fifth and the sixth column of Tables 3.4 and 3.5. From an operational point of view, this clearly illustrates the tradeoff between costs and quality that should be made by any service provider. The considerable magnitude of the differences between scenarios should encourage service providers to consider this trade-off thoroughly for their specific operating circumstances.

3.5 Conclusion

This study focuses on the effect of variations in service level criteria on the operational costs incurred by service providers. By investigating 78 different quality scenarios, being a combination of a maximum deviation from the user's preference time and a relative maximum exceedance of the direct ride time, a tradeoff between both quality criteria and total distance traveled is represented by means of a convex function. The results are found to depend on a number of specific operating circumstances. Larger providers can exploit relatively more request combinations as the service level criteria are loosened, which makes them more sensitive to cost savings than their smaller counterparts. Urban traffic conditions and the presence of heterogeneous user types both limit the relative effect of variations in service level criteria. This analysis is performed on artificial data, adopting deterministic problem characteristics and assuming that solutions are equally reliable. This implies that conclusions should only be interpreted from a conceptual point of view. The precise magnitude of the results cannot be generalized and depends on particular characteristics of a real-life system. For example, information might be limited due to dynamic requests or stochastic travel times and demand might be clustered in terms of both time and space. Moreover, the travel time between two nodes strongly depends on the available road infrastructure. It might be interesting to investigate to what extent such practical characteristics influence the conclusions drawn in this chapter.

The findings of this study are not surprising in the sense that all hypotheses are confirmed. Yet, they also confirm that the research topic of this thesis is highly relevant. Due to the considerable tradeoff of quality and costs, controlling this balance on an operational level is a prerequisite for service providers to implement a consistent strategic policy. In this respect, this chapter also aims to encourage providers to make informed decisions regarding the service policy they pursue. In order to identify which quality scenarios are actually eligible in a specific operating context, providers should also be aware of customer preferences. Particularly, the relative importance users attach to

both service level criteria and the acceptability of specific quality levels should be identified, possibly by means of a customer survey. User preferences generally depend on the health condition or the demographic characteristics of the clientele, but are also subject to individual variation. Especially private service providers should wonder about the elasticity of user behavior. If the desired quality level is not reached, users may be lost to competitors or their satisfaction and willingness to pay may decrease. In contrast, lower operational costs through reasonable quality reductions may allow providers to charge lower rates and to make their services affordable to people with limited budgets. Such information is crucial to weigh the potential for cost savings in various quality scenarios against the needs and the behavior of users.

Along with the research conducted in this chapter, it may also be advisable for service providers to establish advanced service designs which intrinsically reduce their operational costs. For example, it may be interesting to abandon the assumption that all users receive the same quality level. Service level expectations may depend on the reason why a trip is undertaken. Users attending an activity having a fixed time schedule (e.g. a doctor's appointment) often expect a rather strict compliance to their preference time, whereas this might not be important with trips for leisure purposes. Private companies might be able to implement a system in which rates are differentiated according to the service level required by the user. This practice would increase their operational efficiency without harming the users' quality perception. For public companies, this may violate legal regulations.

Chapter 4

Multi-directional local search for a bi-objective problem variant

— *Short summary* —
Although Chapter 3 revealed a major impact of the service level on the operational costs incurred by a service provider, most solution methods do not offer any insight into this fundamental tradeoff. This chapter presents a multi-directional local search (MDLS) algorithm for a bi-objective dial-a-ride problem, considering operational efficiency and service quality as two non-comparable objectives. It aims at delivering a set of non-dominating solutions to the service provider. Section 4.1 introduces the problem context under consideration, generalizing the standard DARP with two types of real-life combination restrictions arising in patient transportation. They prevent certain user combinations and limit the set of drivers to which particular users can be assigned, usually for medical reasons. Section 4.2 reviews the related literature and observes a lack of understanding regarding the effect of these real-life restrictions on the cost structure of a service provider. Section 4.3 explains both the general MDLS strategy and its specific implementation. Based on computational tests, Section 4.4 visualizes the considerable tradeoff of operational costs and service quality. In addition, both types of combination restrictions are shown to have a significant impact on the operational costs. Service providers may use such information to support their policy choices, e.g. related to service quality or medical education of drivers. Conclusions and opportunities for future research are identified in Section 4.5. Note that this chapter is based on Molenbruch et al. (2017c).

4.1 Introduction

The problem context addressed in this chapter is inspired on a real-life application of patient transportation in Belgium. It concerns a dial-a-ride provider who specializes in demand-responsive transportation for the purpose of hospital consultations, medical treatments, daycare activities or rehabilitation therapies. The drivers receive regular training in order to offer a high-quality service tailored to the users' specific needs. Whereas in the standard DARP (Cordeau and Laporte, 2003), users may be combined with any other user(s) and be assigned to any driver/vehicle, this chapter considers a generalization which incorporates two types of combination restrictions. The first type prohibits the combination of specific pairs of users in the same vehicle at the same time. This may be due to medical reasons (e.g. contamination risk, radiation therapy, mental health) or financial agreements between the service provider and a particular institution to make a vehicle exclusively available for its patients. The second type of combination restrictions is caused by the medical needs of certain users. Drivers must be qualified to service users who suffer from a disease which requires special treatment (e.g. behavioral disorder, mental impairment). The impact of these restrictions on the cost structure of the service provider will be analyzed.

Another extension to the standard problem definition relates to the objective function. Given that the cost incurred on a certain arc is proportional to its length, the standard problem minimizes the total distance traveled, which is an operational objective. Such a problem definition assumes that user quality is sufficient as soon as all service level requirements are respected, but does not provide any basic incentive to optimize the service level. This chapter emphasizes the fundamental nature of the DARP by taking into account the conflicting interests of service providers and users in the objective function. Therefore, an additional quality-related objective is added. This second objective minimizes the total user ride time, being the total time users spend aboard the vehicles. All other problem characteristics, including the maximum user ride time constraint, remain unchanged. Since assigning a priori weights on the importance of both objectives is difficult, an MDLS algorithm (Tricoire, 2012) is developed to approximate the Pareto frontier, which is the optimal set of non-dominating solutions. Even though only one solution is executed in practice, this approach provides useful insights into the tradeoff between operational costs and service quality. The operational effect of implementing a particular service level (and vice versa) can be analyzed, which ultimately allows a service provider to reflect on its service policy and to define well-considered quality guidelines. In other words, the solution approach developed in this chapter may serve as a strategic tool to support fundamental decisions related to the quality policy of a service provider, but is also an operational tool to design vehicle routes and time schedules which are in line with this quality policy.

The contribution of this chapter is threefold. First, the standard characteristics of the DARP are extended with real-life constraints in patient transportation. Their impact on the cost structure of the service provider is analyzed. Second, an implementation of an MDLS strategy to a bi/multi-objective DARP is presented, integrating the local search phase into a variable neighborhood descent framework (Hansen et al., 2010). Third, a candidate list is proposed to guide the VND operators, improving the ratio between solution quality and computation time.

4.2 Related literature

Although patient transportation has been a common application of the DARP for many years, the trend of incorporating associated real-life characteristics into models is rather recent. Particularly little attention has been devoted to combination restrictions. Beaudry et al. (2010) consider dynamic patient transportation between different hospital campuses. They take into account emergency requests, isolated transportation and restrictions in vehicle assignments due to the presence of specialized equipment. Applying a tabu search metaheuristic, they minimize a weighted sum of an operational and a service-related objective, being the total vehicle travel time and lateness/earliness. Xiang et al. (2006) impose driver qualifications based on the equipment in the vehicles and the users' disabilities. Their local search approach alternatively optimizes a primary and secondary objective, consisting of various components. A driver's wage is correlated with his qualifications, whereas the cost of a vehicle depends on its equipment. In a study carried out simultaneously with the present work, Detti et al. (2016) consider a similar problem context with combination restrictions among user types or between users and vehicles. They optimize a comprehensive weighted-sum objective function, consisting of fixed and variable distance costs, vehicle travel time, the number of patients transported, vehicle waiting time and user waiting time. Metaheuristics based on tabu search and variable neighborhood search are developed and compared using real-life instances and artificial instances based on this real-life context. However, none of the aforementioned authors analyze the *effect* of the combination restrictions on the cost structure of a provider, nor do they perform tests on the commonly used sets of benchmark data.

Despite its inherent bi-objective nature, most heuristics approach the DARP as a single-objective minimization of the total distance traveled, taking into account minimum service level requirements. Section 2.2.3.2 reviewed different approaches to handle multiple objectives. Most of them apply a mathematical transformation of the objectives, e.g. a weighted-sum objective function. However, the resulting objective value hides trade-offs between the different goals and requires a priori weight choices. Two local-search based approaches in the recent literature aim at approximating the Pareto frontier of non-dominated solutions. Parragh et al. (2009) apply an iterated variable neighborhood

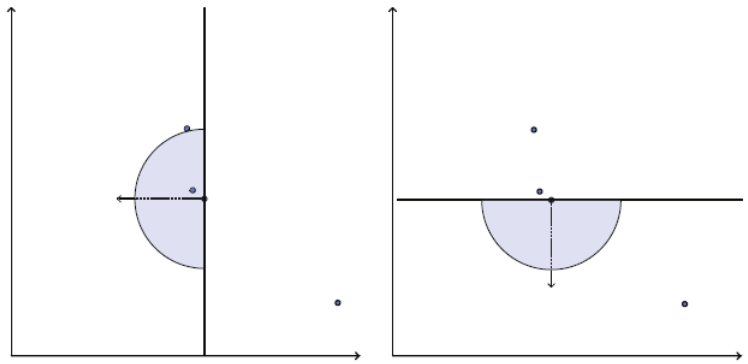

Figure 4.1: MDLS strategy as illustrated in Tricoire (2012), adapted to a minimization.

search to minimize a weighted sum of total distance traveled and mean user ride time for multiple weight combinations. Additional elements combining the characteristics of known solutions, particularly non-supported solutions located on the concave hull of the frontier, are discovered through path relinking. Except for the combination constraints, their problem definition corresponds to the one adopted in the present chapter. Therefore, their results serve as a benchmark in Section 4.4.2. Paquette et al. (2013) incorporate the reference point method of Clímaco et al. (2006) into a tabu search mechanism. A set of non-dominating solutions is constructed using a weighted-sum objective function in which dynamic weights guide the search. The total routing cost, user waiting time and user ride time are minimized. Note that both approaches rely on a repeated weighted-sum strategy, whereas the MDLS metaheuristic applied in this chapter uses a direct multi-objective approach.

4.3 Multi-directional local search algorithm

4.3.1 General strategy

The MDLS strategy was introduced by Tricoire (2012) as an efficient technique to approximate the Pareto frontier without requiring a priori choices regarding the importance of the objectives considered. Its application in this thesis is inspired by the fact that local-search based approaches have frequently been able to reach promising results on single-objective variants of the DARP, as was shown in Chapter 2. The general MDLS approach for a minimization problem is illustrated in Figure 4.1. Each iteration starts from a set of non-dominating solutions, one of which is selected for local improvement. The purpose is to obtain new solutions which are either dominating or uncomparable to the selected one, implying that they should improve the selected solution with respect to at least one objective. Therefore, it is sufficient to perform local search for each objective (in each

direction) separately, always restarting from the same initially selected solution. The different local search operators may be tailored to each direction, either embedded in a metaheuristic framework or not. After each MDLS iteration, all new solutions discovered during the local search are added to the approximation set and dominated or identical solutions are removed. The MDLS strategy intrinsically aims at delivering a well-spread set of solutions. Although a bi-objective problem was presented, more objectives can be included from a conceptual point of view, should this be desirable.

The structure of the method developed in the present study is outlined in Algorithm 1. Details are provided in the following sections. Section 4.3.2 explains how an initial solution set is obtained. Section 4.3.3 discusses the VND framework executed at each MDLS iteration. Section 4.3.4 focuses on the transition between MDLS iterations. Section 4.3.5 proposes an optional path relinking phase.

4.3.2 Initial solution set

After applying the preprocessing steps of Cordeau (2006), a (set of non-dominating) solution(s) is required to initiate the MDLS algorithm. These solutions are constructed by a parallel insertion heuristic (Jaw et al., 1986), using a weighted-sum objective function of total distance traveled and total user ride time. Various weight combinations will be applied to obtain a diverse initial set.

Two variants are distinguished with respect to the insertion order of the users. *Insertion heuristic 1* sorts users by their time preferences and inserts them one by one in such a way that the smallest possible increase in the objective value is obtained. The requests for which combination restrictions apply are inserted first, since finding a feasible insertion position is most difficult for these requests. *Insertion heuristic 2* considers users in a random order and inserts them one by one as explained before. Preliminary tests show that insertion heuristic 1 delivers a better initial result and supports an efficient performance of the algorithm in the long run. Sorting users by their time preferences provides solutions with a desirable structure. After an initialization with insertion heuristic 2, it is time-consuming to achieve a comparable solution quality, presumably due to a mismatch with the structure of the optimal solution. Consequently, insertion heuristic 1 will be applied to initialize all experiments in this study. Insertion heuristic 2 is only invoked if insertion heuristic 1 cannot find any feasible solution and thus generates an empty initial solution set.

4.3.3 VND-based local search

For each MDLS iteration, a single solution is randomly selected from the set of non-dominating solutions and local search is performed in each direction separately. This local search is embedded in a metaheuristic VND framework, consisting of the same three operator types for both directions.

Algorithm 1 - Structure of the complete metaheuristic

```
Initial insertion heuristic (Section 4.3.2)
construct set S of non-dominated solutions

Multi-directional local search (Section 4.3.1)
for i = 1 → n_mdls do
  randomly select s ∈ S
  for each objective do

    Destroy and repair (Section 4.3.4)
    if s optimized to this objective before then
      remove d% of all requests in s and apply 2-regret reinsertion
    end if

    Variable neighborhood descent (Section 4.3.3)
    repeat
      repeat relocate on s until s optimal to this operator
      repeat exchange on s until s optimal to this operator
      repeat natural sequences on s until s optimal to this operator
    until s optimal to all three operators

    Update solution set
    add all new solutions to S and remove dominated solutions

  end for
end for

Path relinking (Section 4.3.5)
for j = 1 → n_pr do
  randomly select s_i ∈ S and s_g ∈ S
  map their routes and transform s_i into s_g
  perform local search on promising solutions

  Update solution set
  add all new solutions to S and remove dominated solutions

end for

return S
```

The choice of VND is substantiated by the strong performances of single-objective variable neighborhood search metaheuristics based on local search in the literature. However, as mentioned in Chapter 2, the choice of effective local search operators that strategically explore different neighborhoods is even more important than the choice of the metaheuristic framework as such. Therefore, this section devotes ample attention to a description of the local search operators included in the VND framework. These operators are based on well-known strategies that perform well in multiple existing implementations of local-search based metaheuristic frameworks.

The *relocate* operator moves one user to the best feasible position in any route (Cordeau and Laporte, 2003). To determine which user is selected, a newly developed candidate list principle is proposed. When minimizing the total user ride time, the candidate list sorts users in descending order of their excess ride time, being the absolute travel time in addition to the direct travel time between their origin and destination. When minimizing the total distance traveled, the candidate list sorts users in descending order of the physical detour currently made to serve them, being the distance reduction that would be obtained if the user was removed. In both cases, the first-ranked user is the first whose relocation is investigated. If it is beneficial to relocate this user, a relatively large improvement in the objective value may be obtained due to his undesirable initial position. If it is impossible to find a better position for this user, relocations for the next users in the candidate list are investigated. One execution of the relocate operator ends as soon as a user can be moved or all users have been investigated in vain. Basically, this candidate list embodies the principle of a gradually enlarging neighborhood, which avoids spending too much time on weak operations.

The *exchange* operator swaps two users in the best feasible way (Braekers et al., 2014). The first user is selected according to the candidate list principle outlined above. The best exchange is found by attempting to insert the first user's origin and destination at the exact positions of any other user's origin and destination. The other user is in turn inserted optimally in the original route of the first user. If no improving exchange can be found involving the first-ranked candidate, the subsequent users in the candidate list are considered. One execution of the exchange operator ends as soon as two users can be swapped to better positions or all users have been examined unsuccessfully.

The *exchange natural sequences* operator performs the best feasible swap of two node sequences before and after which the vehicle is empty. Such natural sequences may be exchanged without any violation of pairing or precedence constraints (Parragh et al., 2010b). In order to select a first series of natural sequences, the candidate list sorts all routes in descending order of the total length of their empty arcs, being the arcs traversed by an empty vehicle. These long empty arcs indicate that the different parts of the route do not match well. Consequently, it may be beneficial to perform

the best exchange of a natural sequence from this route with any other natural sequence in another route. If no improving exchange can be found for any natural sequence in the first-ranked route of the candidate list, the subsequent candidates are analyzed. Again, one execution of this operator ends as soon as two sequences can be swapped to better positions or all routes in the candidate list have been investigated without improvement. At first sight, this operator may seem unusual when minimizing the total user ride time, since swapping natural sequences does usually not influence the user ride times. However, it should be noted that the ride time objective value normally quickly matches its lower bound (all delivery nodes served right after the corresponding pickup node), after which the associated total distance traveled is decisive for obtaining further improvements.

For both objectives in the MDLS framework, one execution of VND proceeds as follows. First, the *relocate* operator is applied as long as improving moves can be discovered and the candidate list is updated after each relocation. When a solution cannot be further improved, the VND principle states that a local optimum with respect to one neighborhood is not necessarily optimal with respect to another neighborhood. Hence, the *exchange* operator and the *exchange natural sequences* operator are applied subsequently in the same manner as explained for the *relocate* operator. The order of the operators is defined on the basis of their computational complexity, such that the first improvements of the selected solution, often requiring a number of simple operations, are performed by the fastest operator. The execution of the VND framework ends as soon as a local optimum is reached with respect to all neighborhoods considered. This implies that the different operator types might be invoked several times (always in the given order), applying them as long as improvements are discovered. Meanwhile, all intermediate solutions that are dominating or uncomparable to the approximation of the Pareto frontier are added to the solution set.

Since minimizing the total user ride time is one of the objectives in the VND framework, it would not be appropriate to assess the feasibility of routes using the traditional scheduling procedure of Cordeau and Laporte (2003). This procedure delivers a feasible schedule whenever one exists, but does not take into account service quality. The total user ride time in the resulting schedules may be unnecessarily large, such that the value of the second objective function would not be representative. Therefore, all computational tests in this chapter will invoke a newly developed scheduling heuristic that aims at minimizing the total user ride time. This procedure will be discussed in Chapter 5.

4.3.4 Transition to the next iteration

From the second MDLS iteration, the selected solution may already have been optimized with respect to one of the objectives in a previous iteration, in which case a diversification is applied first.

A *destroy and repair* operator removes and then reinserts a certain percentage of all requests. Both for the removal and the reinsertion phase, different strategies have been proposed in the literature (Røpke and Pisinger, 2006). The strategies applied in this study are explained below.

Due to the usual tightness of the constraints involved in a DARP, it seems necessary to ensure that the requests selected for removal possess sufficient similarities in order to be interchangeable. This increases the probability of constructing a different feasible solution, which is exactly the aim of the diversification phase. A destroy strategy which is inspired on the Shaw removal (Shaw, 1997) operator is applied. Starting from a random request i with pickup node i and delivery node $i+n$, the most similar requests j ($j \neq i$) are selected until a removal percentage is reached. The tuning of this percentage is explained in Section 4.4.1. Røpke and Pisinger (2006) introduced an extended Shaw removal operator applicable to pickup and delivery problems, which became a frequently used operator in the literature. Their measure of dissimilarity between two requests is based on the sum of (1) the spatial distance, (2) the difference in time of service, (3) the difference in load and (4) the difference in the number of eligible vehicles. Thanks to the small time windows involved with DARPs, the time of service in a node is accurately approximated by the center of its time window, which allows to compute time-relatedness in advance. In addition, the last two terms can be omitted. The difference in load is irrelevant, since each request involves a single user. The difference in the number of eligible vehicles is also neglected, since - at least for the experiments in this chapter - each pair of requests can be served by at least half of the total fleet. Consequently, the dissimilarity r_{ij} of requests i and j is defined as:

$$r_{ij} = \frac{d_{i,j}+d_{i+n,j+n}}{d_{max}} + \frac{|T_i-T_j|+|T_{i+n}-T_{j+n}|}{t_{max}}$$

$d_{a,b}$	Distance between nodes a and b
d_{max}	Length of the longest arc in the network
T_a	Center of the time window of node a
t_{max}	Difference between the largest upper time window bound and the smallest lower time window bound over all nodes
$i, i+n$	Origin and destination nodes of request i

Requests of which both nodes are enclosed between the pickup and the delivery node of a selected request are removed as well. Removing complete clusters of requests may stimulate diversification, since it avoids an easy reconstruction of the original solution.

After the removal phase, a complete solution is reconstructed by a 2-regret insertion heuristic, which partly remedies the myopic behavior of a classical insertion heuristic (Diana and Dessouky, 2004). For each unassigned request, the best and second-best insertion positions are compared in terms of their effect on the objective value. The request with the largest difference between both objective values, which is referred to as the regret value, is inserted with first priority. Postponing this insertion would pose a large risk to the eventual solution quality. Preliminary tests show that, although the 2-regret insertion heuristic consumes more computation time than a classical insertion heuristic, its well-structured solutions require less iterations of the subsequent VND operators. The 2-regret insertion heuristic is less suitable to construct an initial solution to the MDLS procedure, since the regret values should be computed for all requests and updated after each iteration.

4.3.5 Path relinking

To obtain a better and more complete solution set, the MDLS procedure may be supplemented with a subsequent path relinking (PR) phase (Parragh et al., 2009). At the start of each iteration, an initial solution and a guiding solution are randomly selected from the current set of solutions. Each route in the initial solution is mapped to a different route in the guiding solution. Following the findings of Parragh et al. (2009), this mapping is based on the number of identical requests in the routes. The requests in the initial solution being in the wrong route with respect to the guiding solution are relocated one by one to their intended route. The present study applies the principle of best insertion (as with the VND operators), whereas Parragh et al. (2009) adopt the critical vertex rule[1] (Cordeau and Laporte, 2003). A random weighted-sum objective function, taking into account the total distance traveled and the total user ride time, determines the insertion cost. Intra-route relocations are performed on nodes which are wrongly positioned with respect to the guiding solution.

During each phase of the procedure, new (uncomparable or better) solutions may be discovered on the linking path, since intermediate solutions may combine desirable characteristics of the initial solution and the guiding solution. To foster this strategy, an additional local search is performed on solutions within a promising area. This area is delineated by the set of Nadir points (Parragh et al., 2009), taking into account any solution in the current solution set. For such promising solutions, all requests are removed and reinserted one by one in a random order, again applying the principle of best insertion. This additional local search is performed for each objective separately.

[1]This means that for each request to be inserted, they first determine the best position for the node having the smallest time window, which is the delivery node of outbound requests and the pickup node of inbound requests.

4.4 Computational experiments

The MDLS algorithm has been implemented in Python 3.4.3 and executed on a HP ProBook 6570b 2.60 GHz laptop with 4 GB of RAM, running on operating system Windows 7. Cython 0.22, a module running Python code in C and allowing variables to be declared as C types, is invoked to speed up the scheduling procedure. Computation times will be compared with Parragh et al. (2009). Their IVNS algorithm was implemented in C++ and run on a 3.2 GHz Pentium D computer with 4 GB of RAM. The Python language allows for a rapid development of object-oriented programs especially suitable to examine algorithm potential. Unfortunately, these programs tend to execute relatively slowly. Therefore, the savings in computation time obtained using the proposed MDLS algorithm (see Section 4.4.2) should be considered as a lower bound. All computational experiments will be performed on the a-instances of Røpke et al. (2007), discussed in Chapter 2.

4.4.1 Parameter tuning

The destroy and repair operation proposed in Section 4.3.4 requires an informed choice of the destroy percentage. Removing a small part of the requests may be insufficient to change undesirable structures in the solution, whereas removing too many requests may destroy all good characteristics of the solution and undo the improvements obtained in previous iterations. The destroy percentage is tuned in the single-objective context of minimizing the total distance traveled. In a one-directional MDLS, the set of non-dominating solutions can be replaced by a single best-found solution. The selection of the solution prior to each MDLS iteration is therefore not influenced by randomness. A destroy and repair operation is always required, except for the very first iteration. Consequently, performing this single-objective analysis for various destroy percentages reveals the direct effect of the destroy operator and its influence on the course of subsequent iterations. The tuning analysis considers a subset of instances, consisting of $a2$-16, $a2$-24, $a4$-32, $a4$-48, $a6$-48, $a6$-72, $a8$-64 and $a8$-96. This subset is well-diversified in terms of the number of requests, fleet size and time horizon and therefore seems representative of the entire data set. For a range of destroy percentages, Table 4.1 shows the average optimality gap and the average computation time over all instances after an arbitrary number of 100 MDLS iterations. The instances were run five times in order to account for the heuristic nature of the solution method. Weighing the solution quality against the computation time, removing and reinserting 30% of the requests seems the most reasonable choice. This percentage will be applied throughout all of the following experiments. In general, the choice of the destroy percentage may depend on the structure of the benchmark instances and the destroy strategies used.

The effect of the sorted candidate list, introduced in Section 4.3.3, is verified in the same manner. Its assumed benefits lie in the fact that it prioritizes moves which correct inefficient structures in

Destroy	**Opt. gap**	**Time (s)**
20%	1.22%	37.20
25%	0.62%	43.79
30%	**0.48%**	51.61
35%	0.49%	56.07
40%	0.98%	62.70

Table 4.1: Tuning of the destroy percentage based on solution quality and computation time.

	Sorted candidate list		*Random candidate list*	
Iterations	**Opt. gap**	**Time (s)**	**Opt. gap**	**Time (s)**
0	16.98%	0.24	16.98%	0.24
10	2.06%	6.00	2.11%	6.82
20	1.27%	11.25	1.59%	12.11
30	1.02%	16.54	1.19%	18.11
40	0.85%	21.80	0.94%	23.80
50	0.77%	26.70	0.80%	29.53
100	0.48%	51.61	0.52%	57.88

Table 4.2: Effect of the sorted candidate list on solution quality and computation time.

the current solution, causing relatively large improvements of the objective value. The results in the first two columns of Table 4.2 are obtained using the sorted candidate list. The last two columns present comparable results when the next candidate is selected randomly, instead of selecting the first candidate from the sorted list. For the instances considered, these results confirm the idea that a sorted candidate list avoids spending much computation time on operations expected to bring about little progress, thereby improving the ratio between solution quality and computational effort if a first-improvement strategy is applied. The effect is particularly pronounced in the beginning of the procedure, which allows to quickly reach good solutions and thus explore their neighborhood more extensively within the same number of iterations. This sorted candidate list will be included in all of the following experiments. More advanced applications can be considered to fully exploit its efficiency, as will be shown in Section 4.4.2.

4.4.2 The bi-objective DARP without combination restrictions

In order to analyze the impact of the combination restrictions on the benchmark data, the results *without* combination restrictions are computed as a basis of comparison. These results also allow to compare the performance of the MDLS algorithm with the state of the art. In the bi-objective context without combination restrictions, Parragh et al. (2009) have obtained optimal solution sets for small instances, using the epsilon constraint method. Moreover, they have computed heuristic solution sets for all instances, applying iterated VNS (IVNS) with an optional subsequent path relinking phase (IVNS+PR). To compare the solution sets obtained by MDLS (without or with path relinking) with the optimal or best-known results from Parragh et al. (2009), the two most common Pareto-compliant sorting criteria are used, being the hypervolume difference (Zitzler and Thiele, 1999) and the multiplicative epsilon indicator (Zitzler et al., 2003). The indicators are explained in Figure 4.2. The Pareto frontier and its approximation are represented by the white and black dots, respectively. The objective values are normalized as proposed by Parragh et al. (2009). This implies that the total user ride time is normalized with respect to the multiplication of the maximum user ride time and the number of users, whereas the total distance traveled is normalized with respect to the sum of the lengths of the longest arcs leaving each node. Given this normalization strategy, the sorting criteria can be interpreted as follows. The hypervolume difference is the part of the feasible objective space (light-gray plus dark-gray shaded area) which is not dominated by the approximation set (dark-gray shaded area). A value close to zero implies that the approximation set is rather complete, close to the exact Pareto frontier and well-diversified. The multiplicative epsilon value is the smallest common factor by which the coordinates of all points in the exact Pareto frontier need to be multiplied in order to be dominated by the approximation set. The smaller this epsilon, the better the solution and the smoother the pattern of the approximation set. Its best value equals one.

In order to start the MDLS algorithm, the initial insertion heuristic is executed for 11 different weight combinations. In the first execution, 100% of the weight is attributed to the total distance objective. In each following execution, this weight is reduced by 10 percentage points, whereas the weight of the ride time objective is increased by 10 percentage points. Consequently, the last execution attributes the entire weight to the ride time objective. The actual number of initial solutions is not predictable, since various weight combinations might deliver identical solutions. The number of 11 different weight combinations is an arbitrary choice. However, considering more combinations is rarely useful, since the probability of finding additional initial solutions becomes very small.

An appropriate number of MDLS iterations is determined by means of preliminary experiments on the subset of a-instances introduced in Section 4.4.1, again performing five runs per instance. The

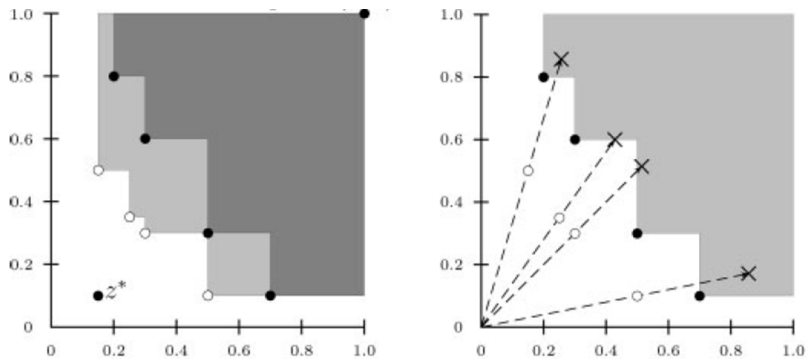

Figure 4.2: Hypervolume (left) and multiplicative epsilon (right) indicator as illustrated in Parragh et al. (2009).

left part of Table 4.3 presents the average hypervolume difference, multiplicative epsilon indicator, size of the solution set and computation time after up to 100 MDLS iterations. As can be observed, the largest improvements in solution quality are brought about by the VND operators during the first iterations. The contribution of each additional iteration in terms of solution quality has a tendency to decrease, whereas the increase in computation time is approximately linear. Performing 80 MDLS iterations seems a reasonable tradeoff between solution quality and computation time. Detailed results for all a-instances are discussed further on in this section.

Computation time is more important for the bi-objective problem than for the single-objective variant. If only the total distance traveled is minimized, deteriorating solutions are recognized and rejected before the time schedule needs to be computed. When minimizing the total user ride time, the time schedule should first be constructed in order to assess a new solution proposal. Although the computation times obtained in the left part of Table 4.3 are suitable for a static environment, a more intelligent use of the sorted candidate list may reduce the computation time. Recall that the candidates likely to bring about a large improvement are located in the front part of the candidate list. This implies that *ignoring the rear part of the candidate list* may reduce the computation time without causing a considerable sacrifice of solution quality. Observations show that the majority of the promising moves involve candidates in the first half of the candidate list. Consequently, the aforementioned assumption can be tested by deleting the second half of the candidate list immediately after its (re)construction. The right part of Table 4.3 indicates that the computation time strongly reduces. The optimality gap in terms of the multiplicative epsilon indicator remains fairly stable, whereas the hypervolume difference increases. This apparent contradiction is explained by a less balanced spread of the solutions in the set, even though their quality is within the same range

	Candidate list 100%				*Candidate list 50%*			
Iter. MDLS	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Time (s)**	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Time (s)**
0	0.03303	1.152	6	0.82	0.03303	1.152	6	0.82
10	0.00690	1.058	22	35.73	0.00962	1.062	22	20.35
20	0.00500	1.050	25	68.37	0.00700	1.052	25	38.63
30	0.00436	1.046	27	99.05	0.00606	1.047	27	57.18
40	0.00371	1.042	27	130.58	0.00559	1.044	27	75.30
50	0.00326	1.039	28	161.00	0.00513	1.042	29	93.18
60	0.00295	1.038	29	191.88	0.00487	1.041	29	111.65
70	0.00277	1.037	29	222.67	0.00464	1.040	29	129.30
80	**0.00259**	**1.035**	**29**	**253.39**	**0.00430**	**1.039**	**30**	**146.18**
90	0.00245	1.035	30	283.71	0.00420	1.038	31	163.62
100	0.00240	1.034	31	314.40	0.00408	1.038	31	180.61

Table 4.3: Results for the bi-objective DARP without combination restrictions (based on 5 runs for each selected instance), considering the full candidate list (left) or its first half (right).

as before. Nevertheless, to exploit the full potential of the proposed MDLS algorithm, all results in the remainder of this chapter will be computed using the full candidate list.

The left part of Table 4.4 presents detailed MDLS results for the bi-objective DARP without combination restrictions. The average hypervolume difference, multiplicative epsilon indicator, size of the solution set and computation time after 80 MDLS iterations are obtained by performing five runs of each instance, applying the full sorted candidate list. An overall comparison is made with the average results of IVNS. Its detailed results on the instance level are available in the original paper. The results in the left part of Table 4.4 illustrate that MDLS strongly outperforms IVNS in terms of both computation time and solution quality, regardless of the sorting criterion. The right part of Table 4.4 provides results after 500 iterations of additional PR. As expected, the exploration of linking paths between good solutions leads to a strong increase in the size of the solution set. Moreover, the average hypervolume difference decreases from 0.00254 to 0.00182, whereas the average multiplicative epsilon indicator changes from 1.036 to 1.022. Even though MDLS+PR still outperforms IVNS+PR on multiple instances, the average improvement in solution quality caused by the PR phase is smaller than in Parragh et al. (2009). Yet, their PR phase has a fixed duration of only 15 seconds, which is less than the average duration of the PR phase is the present study. This

	Results before PR				*Results after PR*			
Inst.	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Time (s)**	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Time (s)**
a2-16	0.00041	1.024	12	6.97	0.00008	1.011	12	9.28
a2-20	0.00013	1.010	12	12.18	0.00004	1.005	17	21.68
a2-24	0.00029	1.013	13	13.14	0.00029	1.006	16	18.66
a3-18	0.00119	1.048	13	9.84	0.00005	1.013	13	13.93
a3-24	0.00071	1.022	9	20.42	0.00054	1.013	14	23.01
a3-30	0.00122	1.025	18	25.37	0.00088	1.020	23	28.72
a3-36	0.00153	1.023	30	36.47	0.00129	1.017	34	43.55
a4-16	0.00039	1.020	8	8.35	0.00030	1.018	7	9.77
a4-24	0.00205	1.042	22	23.99	0.00054	1.018	28	30.25
a4-32	0.00121	1.029	24	39.12	0.00035	1.014	38	44.64
a4-40	0.00254	1.034	21	73.73	0.00227	1.027	29	85.41
a4-48	0.00362	1.043	38	136.46	0.00218	1.022	52	169.34
MDLS	**0.00127**	**1.028**	**18**	**33.84**	**0.00073**	**1.015**	**24**	**41.52**
IVNS	***0.00333***	***1.050***		***241.05***	***0.00090***	***1.011***		***256.05***
a5-40	0.00262	1.033	29	94.76	0.00192	1.022	40	106.56
a5-50	0.00226	1.031	35	130.10	0.00159	1.022	53	144.84
a5-60	0.00438	1.045	42	255.25	0.00287	1.026	53	291.72
a6-48	0.00219	1.035	30	150.84	0.00177	1.025	40	165.62
a6-60	0.00422	1.044	40	266.39	0.00337	1.029	54	301.94
a6-72	0.00394	1.046	39	486.83	0.00300	1.032	57	557.30
a7-56	0.00482	1.051	32	229.84	0.00361	1.029	49	257.78
a7-70	0.00265	1.034	45	367.00	0.00189	1.026	67	416.50
a7-84	0.00498	1.052	56	676.59	0.00370	1.034	64	802.46
a8-64	0.00556	1.052	36	326.45	0.00480	1.034	51	363.58
a8-80	0.00349	1.057	43	651.85	0.00252	1.037	59	751.18
a8-96	0.00461	1.040	48	951.67	0.00380	1.032	73	1120.51
MDLS*	**0.00254**	**1.036**	**29**	**208.07**	**0.00182**	**1.022**	**39**	**240.76**
IVNS*	***0.00388***	***1.053***		***322.81***	***0.00179***	***1.016***		***337.81***

$*$ = *average computed over all instances (small and large)*

Table 4.4: Detailed results for the bi-objective DARP without combination restrictions, performing 80 MDLS iterations without (left) and with (right) 500 iterations of additional PR (based on 5 runs for each instance).

suggests that applying the best insertion principle instead of the critical vertex rule (see Section 4.3.5) is not advisable. Considering a large number of linking paths turns out to be more important than the investigation depth of these paths. Applying the critical vertex rule would allow more PR iterations, i.e. the construction of more linking paths, within the same time frame.

Summarizing all results in this section, the proposed MDLS algorithm is a competitive approach to the bi-objective DARP. Therefore, it can be considered as a valid tool for analyzing the effect of adding combination restrictions in Section 4.4.3.

4.4.3 The bi-objective DARP with combination restrictions

This section quantifies the effect of the combination restrictions in patient transportation. Nine different variants of the benchmark instances of Røpke et al. (2007) have been created. Specifically, the experiments are based on a full factorial design with two factors having three levels each. The first factor relates to the impossibility of combining certain users (percentage c). It is assumed that 0%, 12.5% or 25% of all users cannot share a vehicle with a random selection of other users. This selection contains half of all users in the given instance. For $c = 12.5\%$, the constraint is imposed on the *first* user in each series of eight users in the instance. For $c = 25\%$, the constraint applies to each *first* and *fifth* user. The second factor takes into account the driver restriction (percentage d). It is assumed that 0%, 12.5% or 25% of all users cannot be transported by half of the drivers. Given that an instance contains s vehicles, the drivers of the first $s/2$ vehicles are supposed not to be qualified for transporting these users. For $d = 12.5\%$, this constraint is imposed on the *first* user in each series of eight users in the instance. For $d = 25\%$, this constraint applies to each *first* and *fifth* user. Combining the three levels for both factors results in nine different scenarios. Scenario (0%, 0%) corresponds to the original problem without combination restrictions.

In order to ensure a fair comparison, the analysis is performed on the set of eight representative a-instances from Section 4.4.1. The fleet size in these instances is even and the number of requests is divisible by eight, which is required to apply the aforementioned approach. As a first summary, Table 4.5 quantifies the average effect of combination restrictions over these instances. The average hypervolume difference and multiplicative epsilon indicator after 80 MDLS iterations and 500 iterations of additional PR are computed with respect to the optimal or best-known solution set *without* combination restrictions. The instances were solved five times to account for the heuristic nature of the solution method. As there is little reason to assume that the algorithm in itself would become less efficient when combination restrictions apply, the difference between the optimality gap in a certain scenario and the optimality gap in the (0%, 0%) scenario is likely to be explained by the

	d 0%	**d 12.5%**	**d 25%**
Hypervolume difference			
c 0%	0.00202	0.00330	0.00800
c 12.5%	0.00570	0.00734	0.01137
c 25%	0.01379	0.01501	0.01867
Multiplicative epsilon indicator			
c 0%	1.022	1.026	1.037
c 12.5%	1.029	1.036	1.049
c 25%	1.061	1.067	1.081
Size of the solution set			
c 0%	46	42	37
c 12.5%	39	38	36
c 25%	32	31	29

Table 4.5: Results for the bi-objective DARP with combination restrictions, performing 80 MDLS iterations and 500 iterations of additional PR (based on 5 runs for each selected instance).

combination restrictions. The results show that tightening the combination restrictions consistently increases this gap, while the size of the solution set decreases as solutions may become infeasible. This effect may be different for other data sets or deviating service level criteria, since it depends on the extent to which users are combined in the solution without combination restrictions. Results on the instance level will be discussed later in this section.

To assess the significance of the combination restrictions' impact, inferential statistic tests are applied. The characteristics of the data set should be taken into account when selecting these tests (Anderson et al., 2010). On the one hand, all nine scenarios are tested on an identical set of eight benchmark instances, varying only the strength of the combination restrictions. In other words, the samples in the scenarios (groups) are paired, rather than drawn independently from each other. On the other hand, it cannot be assumed that differences in the values of the sorting criteria (dependent variable) are normally distributed, particularly since the sample size is too limited for the Central Limit Theorem to apply. Therefore, the various groups may be compared using the *Friedman test*, which is the non-parametric variant of the *ANOVA with repeated measures*. For each instance, the Friedman test ranks the values of the dependent variable in all scenarios. It then sums the ranks over all instances. Assuming that the combination restrictions have no effect, the test computes the probability that the differences in the sums of ranks are at least as large as observed. In this case, a

statistical significant difference is found when the combination restrictions are varied. Considering the hypervolume difference as the dependent variable, $\chi^2 = 54.308$ and $p = 0.000$. Considering the multiplicative epsilon value as the dependent variable, $\chi^2 = 48.584$ and $p = 0.000$.

The Friedman test confirms the presence of an overall effect due to combination restrictions. Additional executions of the *Wilcoxon signed-rank test*, which is a non-parametric variant of the *paired t-test*, are required to reveal which pair(s) of scenarios are significantly different from each other. Only a symmetric distribution of the dependent variable is required. The Wilcoxon signed-rank test computes the absolute difference of the dependent variable for each instance. These differences are ranked and each rank receives the sign of the original difference. Assuming that the sum of these ranks follows a symmetric distribution around zero, the probability of the encountered deviation is assessed. In this case, the test reveals that each possible tightening of a combination restriction, i.c. each rightward or downward move in Table 4.5, causes a significant increase of the hypervolume difference, taking into account a 5% significance level. Considering the multiplicative epsilon value as the dependent variable, the scenarios (c 0%, d 12.5%) and (c 12.5%, d 0%) are not significantly different from the scenario without combination restrictions: $p = 0.106$ and $p = 0.075$, respectively. However, each other rightward or downward move in Table 4.5 leads to a significant increase of the multiplicative epsilon value, again taking into account a 5% significance level.

Figure 4.3 visualizes the impact of the combination restrictions for a randomly chosen instance (a6-48), showing the exact Pareto frontier without combination restrictions (Parragh et al., 2009) and the MDLS approximations in the scenarios (c 0%, d 0%) and (c 25%, d 25%). Comparing both extremes of the exact Pareto frontier illustrates a major tradeoff between service quality and solution cost. The total distance traveled ranges between 604.48 and 743.44, whereas the corresponding total user ride time varies between 818.81 and 486.32. Although a service provider can execute only one solution from the entire set, this bi-objective approach may be used to gain insight into its operational cost structure. For example, a single-objective minimization of the total distance traveled would deliver the solution (604.48, 818.81). However, slightly increasing the total distance traveled to 609.77 (+0.88%) allows to improve the total user ride time to 707.40 (−13.61%). Such information enables a service provider to analyze the operational effect of implementing a certain level of service (and vice versa), which ultimately supports a well-considered quality policy. Comparing the approximation set in scenario (c 25%, d 25%) with the exact or approximate frontier without any combination restrictions, the lower bound on the total user ride time remains unchanged, since users' origins and destinations can still be visited consecutively. The corresponding total distance traveled may increase in the presence of driver restrictions, since all users requiring qualified drivers should be assigned to a more limited number of vehicles. Even with 25% of such users, matching the

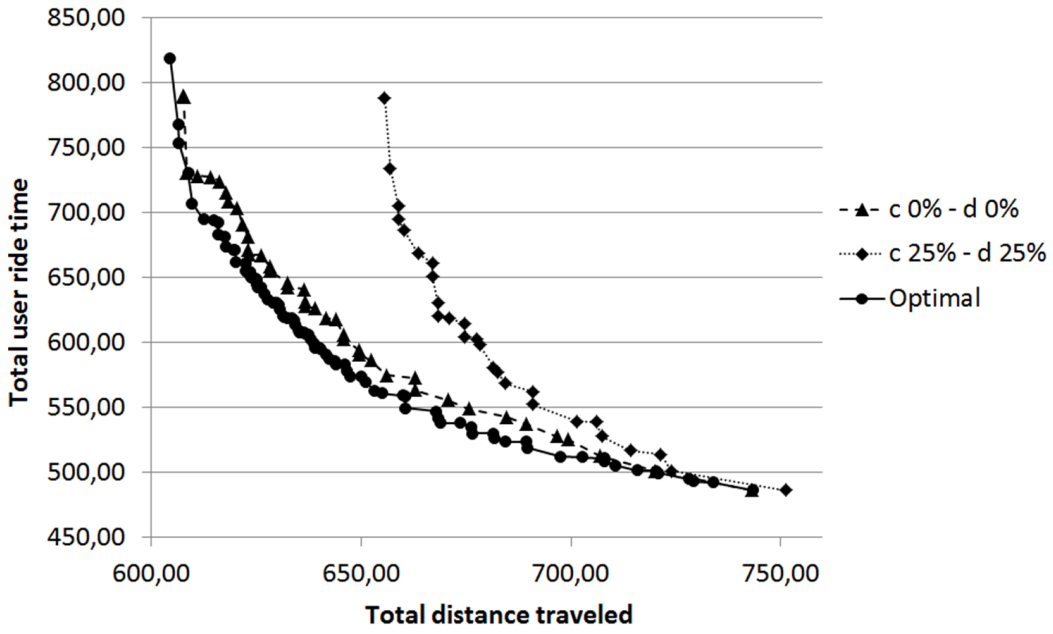

Figure 4.3: Visualization of the exact Pareto frontier and approximate solution sets for instance $a6$-48, without and with combination restrictions.

lower bound on the total user ride time only causes a rather limited increase in the total distance traveled from 743.44 to 751.24 (+1.05%). However, the best-found solution in terms of the total distance traveled is heavily influenced. Compared to the scenario without combination restrictions, the best-found total distance traveled increases from 604.48 to 655.62 (+8.46%). Particularly the impossibility of combining certain user pairs in the same vehicle may require large detours. These conclusions can be generalized to all instances.

Table 4.5 does not provide any information regarding the variability of the results. Particularly, it is interesting to investigate to what extent both sorting criteria are sensitive to the choice of the users for whom combination restrictions apply. For this purpose, the same analysis is repeated three more times. In the second replication, the constraints $c = 12.5\%$ and/or $d = 12.5\%$ (if applicable) affect the *second* user in each series of eight users, whereas the constraints $c = 25\%$ and/or $d = 25\%$ (if applicable) affect each *second* user and each *sixth* user. Analogously, a third replication imposes constraints on each *third* user and each *seventh* user and a fourth replication does so on each *fourth* user and each *eighth* user. Table 4.6 compares the average results for scenarios (c 12.5%, d 12.5%) and (c 25%, d 25%) over all four replications with the baseline scenario (c 0%, d 0%). Again, each

	(c 0%, d 0%)			*(c 12.5%, d 12.5%)*			*(c 25%, d 25%)*		
Inst.	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Hypervol. difference**	**Multiplic. epsilon**	**#**	**Hypervol. difference**	**Multiplic. epsilon**	**#**
a2-16	0.00008	1.011	12	0.00831	1.033	10	0.03519	1.107	7
a2-24	0.00015	1.006	16	0.00978	1.043	16	0.02632	1.093	11
a4-32	0.00035	1.014	68	0.00604	1.029	29	0.01649	1.063	23
a4-48	0.00218	1.022	52	0.00765	1.034	39	0.01774	1.073	37
Avg.	**0.00069**	**1.013**	**37**	**0.00795**	**1.035**	**24**	**0.02394**	**1.084**	**20**
a6-48	0.00177	1.025	40	0.00648	1.039	36	0.01679	1.080	32
a6-72	0.00300	1.032	57	0.00793	1.047	50	0.01519	1.072	50
a8-64	0.00480	1.034	51	0.00823	1.046	39	0.01390	1.073	38
a8-96	0.00380	1.032	73	0.00815	1.044	68	0.01175	1.061	63
Avg.*	**0.00202**	**1.022**	**46**	**0.00782**	**1.039**	**36**	**0.01917**	**1.078**	**33**

$*$ = *average computed over all selected instances (small and large)*

Table 4.6: Average results for the bi-objective DARP with combination restrictions, performing 80 MDLS iterations and 500 iterations of additional PR (based on 4 replications with 5 runs for each selected instance).

replication consists of five runs for each selected instance. Table 4.7 represents the observed spread for both sorting criteria, being the difference between the largest and the smallest result among all four replications. The results confirm the intuition that the range of the effect due to combination restrictions is more substantial for the smallest instances. This is explained by a lack of flexibility in assigning users to vehicles, mainly caused by the driver restriction. For example, all users who cannot be transported by half of the drivers should be assigned to one particular vehicle when the total fleet consists of two vehicles. Exceptionally, this may impede the feasibility of the instance, necessitating an extension of the fleet size or an undesirable rejection of users. For larger instances, a more targeted assignment can be made among a number of allowed vehicles, which explains why these instances are less susceptible to the impact of combination restrictions.

4.5 Conclusions and future research

This chapter considers a generalization of the standard dial-a-ride problem, incorporating real-life characteristics of a patient transportation system in Belgium. It studies the impact of two types

	(c 12.5%, d 12.5%)		*(c 25%, d 25%)*	
Inst.	**Hypervol. difference**	**Multiplic. epsilon**	**Hypervol. difference**	**Multiplic. epsilon**
a2-16	0.01925	0.068	0.05063	0.143
a2-24	0.01835	0.081	0.01902	0.069
a4-32	0.00605	0.019	0.01920	0.060
a4-48	0.00453	0.016	0.00832	0.036
Avg.	**0.01205**	**0.046**	**0.02429**	**0.077**
a6-48	0.00496	0.025	0.00731	0.036
a6-72	0.00825	0.031	0.00985	0.048
a8-64	0.00370	0.014	0.00688	0.037
a8-96	0.00419	0.019	0.00465	0.028
Avg.*	**0.00866**	**0.034**	**0.01573**	**0.057**

$*$ = *average computed over all selected instances (small and large)*

Table 4.7: Spread analysis for the effect of combination restrictions applied to different users (based on 4 replications with 5 runs for each selected instance.)

of combination restrictions that may apply for medical reasons. The first type prohibits combinations of specific pairs of users, whereas the second type requires that particular users are assigned to a driver having medical qualifications. Although these are common restrictions in patient transportation, the literature lacked insights into their effect on the cost structure of a service provider. Therefore, the present study captures this effect in a general pattern. Computational experiments analyze different scenarios in which the strength of both constraints is varied, comparing them with a baseline scenario in which no combination restrictions are imposed. Combination restrictions are shown to have a considerable impact on the cost structure of a service provider, although the precise magnitude may depend on the characteristics of the data. This approach is ready for tuning policies in real-life dial-a-ride systems. For example, it allows service providers to take an informed decision with respect to the number of drivers that should receive medical training in order to optimize the overall costs, taking into account the effect on service quality.

A bi-objective approach is adopted to solve the problem variant under consideration. Contrary to most single-objective methods, which optimize an operational objective subject to given service level requirements, the fundamental nature of the DARP is emphasized by means of an additional, service-level related objective of minimizing total user ride time. An MDLS algorithm is developed

to deliver a set of non-dominating solutions. Even though only one solution can actually be executed, this approach provides useful insights into the tradeoff between operational costs and service quality. The operational effect of implementing a particular service level (and vice versa) may be analyzed, which ultimately allows a service provider to reflect on its service policy and to define well-considered quality guidelines. The local search within the MDLS algorithm is embedded in a VND framework, guided by a sorted candidate list which reduces the computation time. Ignoring the rear part of the candidate list results in additional computational benefits without sacrificing much solution quality. Such advanced applications of this sorted candidate list principle may be exploited in future work. MDLS is completed with a path relinking phase to obtain an even more extensive and diversified solution set. Since MDLS is extendable to a larger number of objectives, additional objective components may be defined based on suggestions of stakeholders in daily practice. Both operational objectives, such as minimizing the required fleet size during peak periods, and quality-oriented goals, such as minimizing the waiting time faced by users, may be considered.

Chapter 5 provides extensive details on the new scheduling procedure used for all computational tests performed in this chapter. This procedure minimizes the total user ride time in a route, such that the solution quality with respect to the second objective in the MDLS algorithm can be assessed in an accurate manner.

Chapter 5

Scheduling procedure minimizing the total user ride time

— *Short summary* —
Scheduling procedures are invoked to prove the time feasibility of a route. They compute the start of service in each node such that all time-related constraints are satisfied. Section 5.1 discusses the commonly used scheduling procedures from the literature. Cordeau and Laporte (2003) calculate the earliest schedule respecting time windows and travel times, after which the forward time slack principle eliminates violations of maximum user ride times or route duration. Since this procedure ignores service quality, Parragh et al. (2009) modify the forward time slack computation such that the total user ride time is minimized at the expense of occasional incorrect infeasibility declarations. Section 5.2 proposes a new scheduling heuristic that minimizes the total user ride time according to a different strategy. Starting from a schedule with minimal ride times for the given time windows, potential travel time shortages are eliminated while keeping ride time increases as limited as possible. A variant imposing constraints on the allocation of waiting time is introduced in Section 5.3. The experiments in Section 5.4 show that the procedure is faster and fails on fewer routes than the one of Parragh et al. (2009). In addition, feasible schedules exhibit smaller deviations from the optimal solution. Note that the proposed scheduling heuristic was applied in all experiments of Chapter 4.

5.1 Introduction

Proving the feasibility of a route requires the construction of a time schedule, since time-related constraints (time windows, maximum ride time and maximum route duration) can only be checked if

the start time of the service in each node is known. Due to the maximum user ride time, scheduling is more complicated for the DARP than for related routing problems, like the PDPTW (Cordeau and Laporte, 2007). An overview of applicable scheduling procedures was given in Chapter 2. Most solution methods invoke the procedure of Cordeau and Laporte (2003), which is based on the forward time slack principle (Savelsbergh, 1992). The procedure subsequently minimizes violations of time windows, maximum route duration and maximum user ride times. It was originally designed to explore infeasible regions of the search space at the expense of a penalty cost. However, since the last two phases do not influence violations in the previous phase(s), it may also be used if feasibility should be maintained at all times. The strategy is as follows. First, the start of service in each node is set equal to the maximum of (1) the lower bound of its time window and (2) the earliest time by which this node can be reached. Second, the start of service in the start depot is postponed as much as possible without causing additional time window violations, which minimizes the total route duration. This shift can never increase any user ride time. Third, as long as user ride time violations are found, an analogous shift is considered for all pickup nodes. The pickup nodes are investigated one by one, using a forward loop through the route. The start of service is postponed as much as possible, without causing additional time window violations, ride time violations or route duration increases. This shift may eliminate a possible ride time violation for the corresponding user. More details on the implementation of this scheduling procedure can be consulted in the original paper of Cordeau and Laporte (2003)[1]. It guarantees to construct a feasible schedule whenever one exists.

Parragh et al. (2009) observe that this sequential approach to minimize ride time violations does not necessarily minimize the total user ride time in the eventual schedule. Postponing the start of service in a user's pickup node may increase (but not violate) the ride times of other users aboard the vehicle at the moment of this pickup. However, when minimizing the total or average user ride time is one of the objectives, such as in Chapter 4, computing the minimal total user ride time of the routes is essential. Even with other objectives, service providers may also prefer these quality-oriented schedules. To obtain better approximations of the minimal total user ride time, Parragh et al. (2009) adapt the computation of the forward time slack such that user ride time increases are avoided during the procedure. However, this modification makes the procedure more restrictive in terms of feasibility. The elimination of a maximum ride time violation for one user will be overlooked when it increases another user's ride time, which may result in an incorrect infeasibility declaration.

Figure 5.1 presents an example route for which the schedule of Cordeau and Laporte (2003) exceeds the optimal total user ride time. Rather than preventing this deviation, the adapted procedure

[1] A slight correction to the mathematical representation is made in Parragh et al. (2010b), page 1133. Computing the forward time slack in a node, the ride time margin $L - P_j$ is only considered for users picked up before this node.

of Parragh et al. (2009) generates an incorrect infeasibility declaration. The route is represented as follows. Each user i is transported between a pickup node P_i and delivery node D_i. All trips involve a 20-minutes time window on either the pickup or the delivery, indicated in bold. A time window for the other node, indicated in italics, is computed based on the travel times (10 minutes between all nodes) and the maximum user ride time (40 minutes). Without loss of generality, service durations are assumed to be zero and the maximum route duration is sufficiently large. This example focuses on the first natural sequence in the route, ending at node D_2. The second natural sequence starts at node P_4. Given the time windows of both nodes, scheduling decisions in either natural sequence will not exclude any feasible schedule in the other sequence. Nevertheless, the amount of slack time allocated to the intermediate empty arc may influence the ride times of the users.

The procedure of Cordeau and Laporte (2003) is applied in the upper part of Figure 5.1. First, the earliest schedule respecting time windows and travel times is computed. Second, the start of service in the depot node is postponed by 20 minutes. A larger shift would violate the time window of node P_1. Third, since the maximum ride time of user 2 is still violated, the pickup nodes in the route are considered in a forward loop. The start of service in node P_1 already matches its upper time window bound, but the start of service in node P_2 can be postponed by another 20 minutes without violating any time-related constraint. This results in a feasible final schedule for which the sum of all user ride times in the first natural sequence equals 90 minutes. This is not the optimal schedule for the given route. The procedure of Parragh et al. (2009) is applied in the middle part of Figure 5.1. Its adapted computation of the forward time slack does not allow the last shift, which increases the ride time of user 1, such that no feasible schedule is found. This example illustrates why procedures based on the forward time slack principle may have difficulties to schedule routes in which complicated tradeoffs between the ride times of different users need to be made.

Therefore, this chapter presents a new scheduling heuristic that aims at minimizing the total user ride time according to a completely different strategy. The various steps of this procedure will be described in Section 5.2. It starts from a schedule in which the total user ride time equals its lower bound (step 1). This schedule may be infeasible due to insufficient travel times on particular arcs. Gradually, the procedure tries to establish feasibility while avoiding unnecessarily large increases in the total user ride time (steps 2 and 3). For the aforementioned example, the proposed scheduling procedure generates a feasible schedule in which the total user ride time matches the optimum of 80 minutes, as indicated in the lower part of Figure 5.1. More generally, it outperforms the procedure of Parragh et al. (2009) in terms of solution quality and computation time. The resulting schedules match the optimal solution more frequently, the remaining deviations are smaller and the risk of incorrect infeasibility declarations is strongly reduced. These claims will be supported by the com-

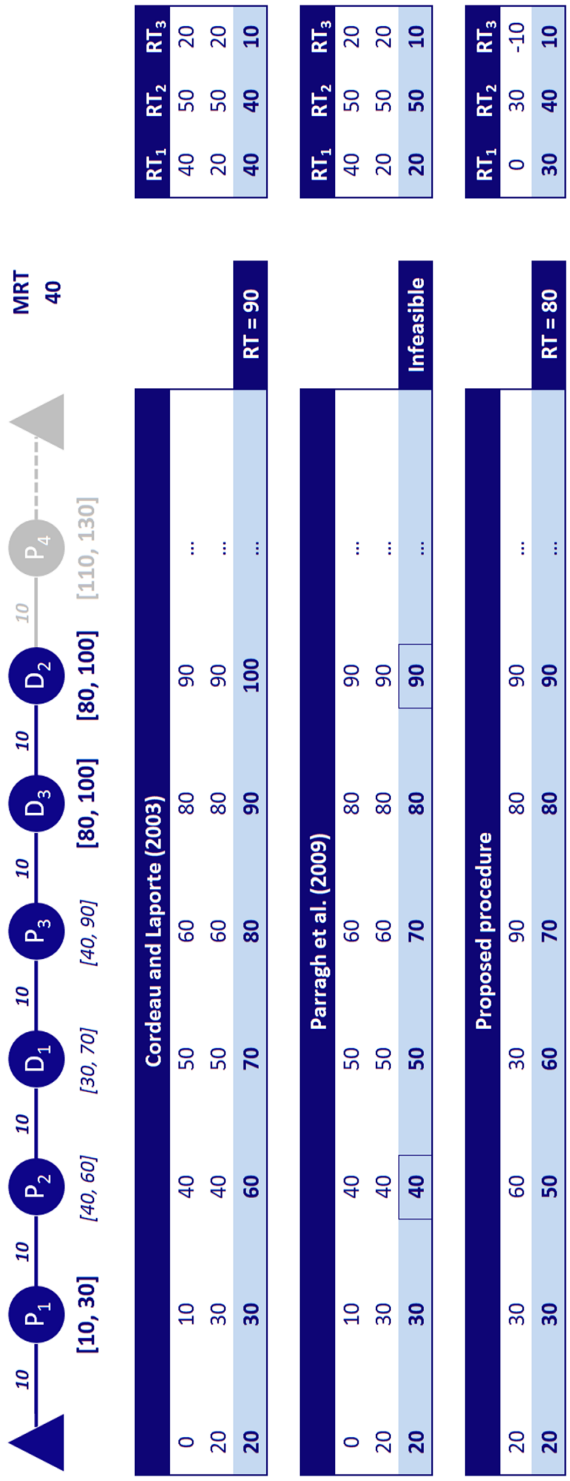

Figure 5.1: Example route causing a suboptimal schedule by Cordeau and Laporte (2003) and an incorrect infeasibility declaration by Parragh et al. (2009).

putational experiments in Section 5.4. Finally, note that Section 5.3 presents an extended version of the scheduling heuristic, taking into account additional real-life waiting restrictions to increase the perceived service quality. The effects of including these constraints will also be tested in Section 5.4.

To increase the computational efficiency of any scheduling procedure, preliminary time checks may prove the infeasibility of a route before the time schedule is constructed. The preliminary time checks proposed by Braekers et al. (2014) are adopted here. They compute the earliest arrival and latest departure in each node based on travel times, time windows and service durations, ignoring user ride times. The scheduling procedure only needs to be invoked for routes that are feasible with respect to the aforementioned constraints.

5.2 Scheduling procedure

The three steps of the proposed scheduling procedure are explained below. It was designed for a standard problem (Cordeau and Laporte, 2003) after applying the preprocessing steps of Cordeau (2006). A glossary of the notation used here can be found in the *List of symbols* section of this thesis.

Step 1

The aim of this step is to create a (possibly infeasible) initial schedule in which the ride times of all users match their lower bounds, taking into account time windows and travel times between nodes of the same type only. Making a forward loop through the route, an initial value is determined for the start of service in each delivery node. Each B_i (with $i \in D$) is set equal to the corresponding lower time window bound e_i, such that each user is delivered as early as possible. However, when i is situated immediately behind another delivery node $h \in D$, it is checked whether node i is reachable at $B_i = e_i$. If not, B_i needs to be scheduled later and is therefore postponed to $B_h + d_h + t_{hi}$. At the end of the loop, the start of service B_{2n+1} in the end depot is determined as if it was a delivery node. By making a backward loop through the route, an initial value is determined for the start of service in every pickup node. Each B_i (with $i \in P$) is set equal to the corresponding upper time window bound l_i, such that each user is picked up as late as possible. However, when i is situated just before another pickup node $j \in P$, a start of service $B_i = l_i$ must allow to reach node j in time. If not, B_i needs to be scheduled earlier and is therefore advanced to $B_j - d_i - t_{ij}$. At the end of the loop, the start of service B_0 in the start depot is determined as if it was a pickup node (typically $d_0 = 0$). No feasible solution exists if $B_{i+n} - (B_i + d_i)$ exceeds the maximum user ride time L for at least one user i. Analogously, the procedure can be aborted if $B_{2n+1} - B_0$ violates the maximum route duration T.

The resulting schedule is feasible in terms of user ride times, route duration and time windows. The total user ride time is situated at a lower bound, since each user is picked up as late as possible and delivered as early as possible. The current schedule represents the optimal solution if it is also feasible in terms of travel times, which is usually not the case. A shortage in travel time, meaning that the condition $B_j \geq B_i + d_i + t_{ij}$ is violated for two successive nodes i and j, might occur on the arcs going from a delivery node to a pickup node (type DP) and the arcs going from a pickup node to a delivery node (type PD). In the former case, no feasible solution exists and the procedure is aborted. In the latter case, the next steps show how a feasible schedule may be obtained by adapting the start of service in the adjacent nodes, while only increasing user ride times if necessary.

Step 1 is visualized in the upper part of Figure 5.2. The maximum user ride time in this slightly more complicated example route equals 50 minutes. First, a forward loop is performed to schedule the deliveries of all users. Users 1 and 2 are delivered at the lower bound of the corresponding time window. Since both user 3 and user 4 are delivered right after another delivery node, their start of service must be postponed if the travel time on the preceding arcs is insufficient. This explains why node D_4 is served at time 110 instead of 90. Second, a backward loop is performed to schedule all pickups. Users 4 and 3 are picked up at the upper bound of the corresponding time window. Since users 2 and 1 are picked up just before another pickup node, their start of service should be advanced if the travel time on the next arc is insufficient, which is not the case in the given example. After step 1, travel time shortages (M) of 50 and 10 minutes remain on arcs P_3D_1 and P_4D_2, respectively.

Step 2

The aim of step 2 is to eliminate travel time shortages on PD arcs by absorbing potential slack time on surrounding arcs. This can be done by (1) advancing the start of service in one or several immediately preceding pickup nodes, (2) postponing the start of service in one or several immediately succeeding delivery nodes or (3) a combination of both. These shifts should be performed without violating any time windows or maximum user ride times. Since they increase the ride time of at least one user, a specific strategy is developed to keep the total increase as small as possible.

All PD arcs with a travel time shortage are considered one by one in a forward loop through the route. The shortage on arc ij ($i \in P$, $j \in D$) can be quantified as $M_{ij} = max(0, B_i + d_i + t_{ij} - B_j)$. It may be resolved by shifting the start of service in the immediately preceding pickup node i and the immediately succeeding delivery node j. B_i is advanced by the minimum of (1) the travel time shortage M_{ij}, (2) the time window margin for node i, being $B_i - e_i$, (3) the ride time margin for

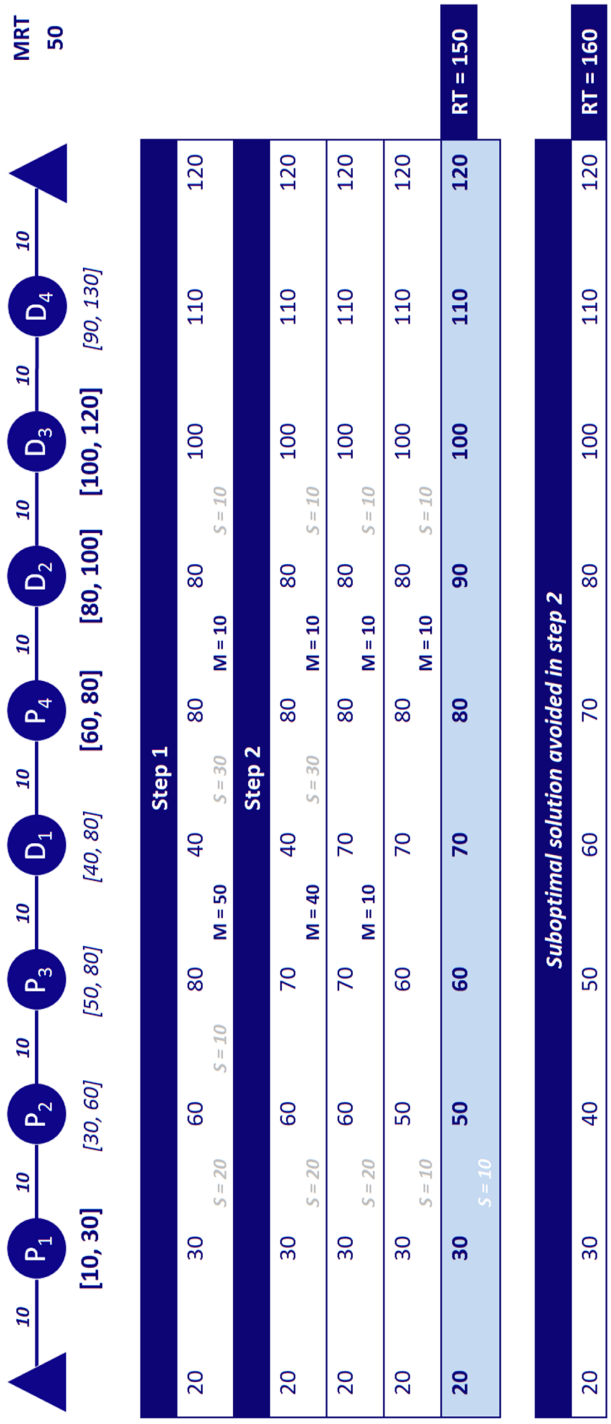

Figure 5.2: Example route for which the proposed procedure finds the optimal schedule in step 2.

user i, being $L - (B_{i+n} - (B_i + d_i))$, and (4) the travel time slack on the preceding arc hi, being $B_i - (B_h + d_h + t_{hi})$. If this shift does not entirely eliminate the travel time shortage on arc ij, an analogous strategy is applied to B_j. This implies that B_j is postponed by the minimum of (1) the remaining travel time shortage M_{ij}, (2) the time window margin for node j, being $l_j - B_j$, (3) the ride time margin for the user delivered in j, being $L - (B_j - (B_{j-n} + d_{j-n}))$, and (4) the travel time slack on the succeeding arc jk, being $B_k - (B_j + d_j + t_{jk})$. Each of both shifts involves an increase in the ride time of a single user. If any travel time shortage remains on the arc ij, it may be further reduced by simultaneously advancing the start of service in the two immediately preceding pickup nodes (if present) and/or by simultaneously postponing the start of service in the two immediately following delivery nodes (if present), according to the same principle. However, each of these shifts would increase the ride times of two users. This iterative pattern continues until one of the following conditions is met: (1) the travel time shortage on the arc ij is entirely eliminated, (2) the series of immediately preceding pickup nodes (resp. immediately succeeding delivery nodes) is interrupted by a delivery (resp. pickup) node or a depot node, or (3) no further shift can be performed without violating a time-related constraint. Then, the next PD arc with a travel time shortage is considered.

In the middle part of Figure 5.2, the shifts in step 2 are visualized for the given example route. Arc P_3D_1 is the first PD arc on which a travel time shortage is found. First, the start of service in node P_3 can be advanced by 10 minutes, which only increases the ride time of user 3. Analogously, the start of service in node D_1 can be postponed by 30 minutes, which only increases the ride time of user 1. Since these shifts are not sufficient to eliminate the full travel time shortage on arc P_3D_1, the remaining shortage of 10 minutes is eliminated by advancing the start of service in both nodes P_2 and P_3, thereby increasing the ride times of two users. Then, the only remaining PD arc with a shortage in travel time is arc P_4D_2. This shortage of 10 minutes cannot be eliminated by advancing the start of service in node P_4, but it can entirely be solved by postponing the start of service in node D_2. This results in the optimal schedule. Note that the order by which slack is consumed is essential to avoid unnecessarily large ride times. For example, shifts increasing the ride times of as few users as possible need to be executed first. To illustrate this, the lower part of Figure 5.2 shows an example of a suboptimal schedule that is found by first consuming all time slack on arc P_1P_2.

If the scheduled travel times after step 2 are sufficient for all PD arcs, a feasible solution has been found and the procedure terminates. Although this solution usually corresponds to the optimal result, a slight deviation might occur. Figure 5.3 shows an example route for which travel time shortages are found on arcs P_3D_1 and P_4D_2 after step 1. The travel time shortage of 40 minutes on arc P_3D_1 is encountered first during the forward loop in step 2. No time slack can be absorbed on the preceding arc P_2P_3. However, absorbing the entire time slack on the subsequent arc D_1P_4 by

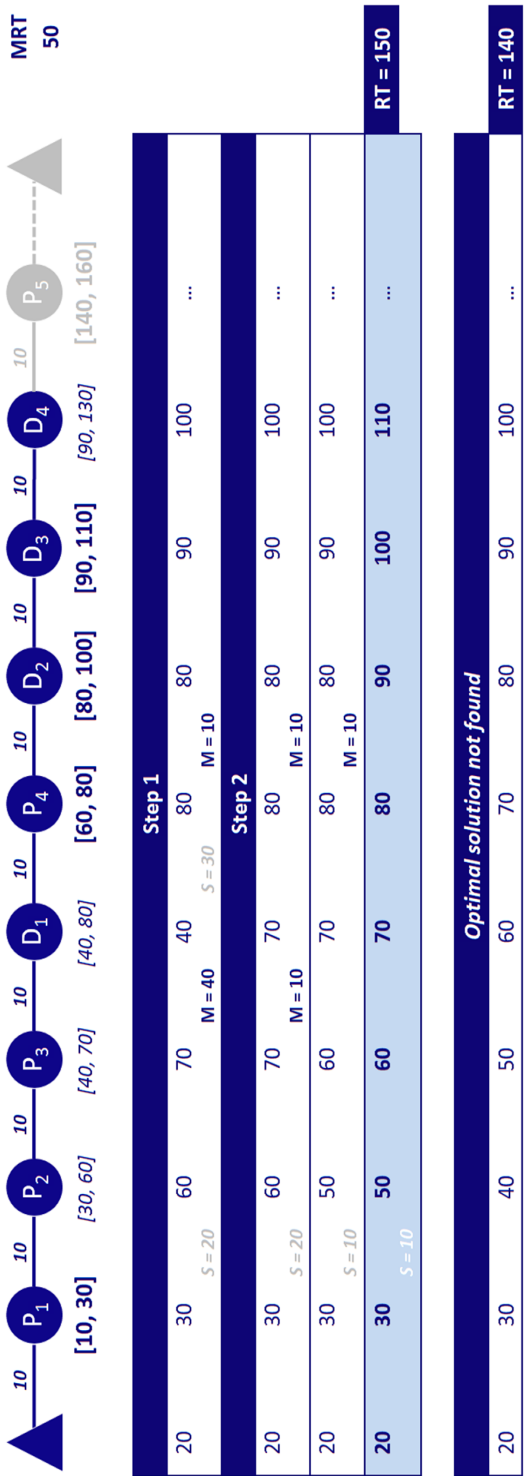

Figure 5.3: Example route for which the proposed procedure finds a suboptimal schedule in step 2.

postponing the start of service in node D_1 allows to eliminate 30 minutes of travel time shortage, while the ride time of only user 1 is increased. Absorbing 10 minutes of slack time on arc P_1P_2 by advancing the start of service in nodes P_2 and P_3 eliminates all remaining travel time shortage on arc P_3D_1, while increasing the ride times of both users 2 and 3. Although this decision is optimal in the isolated context of arc P_3D_1, absorbing all slack time on arc D_1P_4 implies that the travel time shortage of 10 minutes on arc P_4D_2 needs to be eliminated by postponing the start of service in nodes D_2, D_3 and D_4, thereby increasing the ride times of three users. This results in a total user ride time of 150 minutes for the first natural sequence. Yet, the lower part of Figure 5.3 proves that a total user ride time of 140 minutes can be obtained by absorbing all slack time on arc P_1P_2, i.c. taking a suboptimal decision for arc P_3D_1.

Step 3

If the travel times scheduled in step 2 remain insufficient for at least one PD arc, step 3 proposes two strategies according to which a feasible schedule may still be obtained. The first strategy considers the fact that unused slack time on the arcs in the front part of the route may be used to eliminate remaining travel time shortages in the rear part. The second strategy attempts to reach feasibility by shifting the start of service in the depot nodes and thus increasing the route duration, which option was ignored in step 2. Although the latter strategy may seem convenient to eliminate any remaining travel time shortages, it should be applied with care. On the one hand, it may give rise to suboptimal solutions. On the other hand, it reduces the room to maneuver at the opposite end of the route, due to the fact that the maximum route duration constraint should not be violated.

Step 3a

The aim of step 3a is to check whether travel time shortages at the end of the route can be eliminated by unused slack time earlier in the route. If the last PD arc in the route does not contain any travel time shortages, this step is redundant: the last pickup node is labeled as the breakpoint and the procedure immediately proceeds to step 3b. Otherwise, a backward loop starts from this pickup node and the start of service in each encountered node i is adjusted. Each B_i is advanced by the minimum of (1) the travel time shortage on the arc leaving node i (e.g. ij), being equal to $max(0, B_i + d_i + t_{ij} - B_j)$, (2) if $i \in P$, the available ride time margin, being $L - (B_{i+n} - (B_i + d_i))$, and (3) the time window margin $B_i - e_i$. As a result, maximum user ride times and time windows will not be violated, while the travel time shortage on the last PD arc is eliminated or temporarily transferred to its preceding arcs. The backward loop ends at the breakpoint, being the first node for which the proposed shift equals 0. In case a feasible schedule is reached, the procedure terminates. Otherwise, the procedure continues to step 3b.

The upper part of Figure 5.4 presents an example route for which a 10 minutes travel time shortage remains on arc P_4D_2 after step 2. Contrary to the previous example in Figure 5.3, the shortage is not addressed by jointly postponing the start of service in nodes D_2, D_3 and D_4, since this would now cause a route duration increase. Starting from arc P_4D_2, which is the last PD arc in the route, the starts of service in its preceding nodes are advanced such that the remaining slack time on arc P_1P_2 is absorbed, creating a feasible schedule. Applying these shifts before considering a potential route duration increase avoids suboptimal schedules as the one shown in the lower part of Figure 5.4.

Step 3b

For all arcs traversed after the breakpoint, the only remaining possibility to eliminate travel time shortages is to postpone the starts of service in the succeeding nodes, including the end depot. To this end, a forward loop starts from the breakpoint. For each encountered arc ij with a remaining travel time shortage, B_j is adjusted to $B_i + d_i + t_{ij}$. An infeasibility declaration is made as soon as such a shift (1) violates the corresponding upper time window bound, meaning that $B_j > l_j$, or (2) if $j \in D$, violates the maximum ride time for the user delivered at j, such that $B_j - (B_{j-n} + d_{j-n}) > L$. In these cases, the procedure is aborted. This also happens if the required shift of B_{2n+1} violates the maximum route duration, meaning that $B_{2n+1} - B_0 > T$. For all arcs before the breakpoint, the only remaining possibility to eliminate travel time shortages is to advance the start of service in one or more preceding nodes, possibly including the start depot. A backward loop is performed, starting from the breakpoint. For each encountered arc ij with a remaining travel time shortage, B_i is adjusted to $B_j - d_i - t_{ij}$. An infeasibility declaration is made if such a shift (1) violates the corresponding lower time window bound, meaning that $B_i < e_i$, or (2) if $i \in P$, violates the maximum ride time for user i, such that $B_{i+n} - (B_i + d_i) > L$. Moreover, the procedure is also aborted if the required shift of B_0 violates the maximum route duration, meaning that $B_{2n+1} - B_0 > T$.

Figure 5.5 presents an example route for which two travel time shortages remain after step 2. As there is no remaining slack time on any arc, it is immediately clear that a feasible schedule can only be reached by shifting the departure and arrival times at the depot. In this example, step 3a starts on arc P_4D_2, which is the last arc in the route for which a travel time shortage is found. It tries to advance the start of service in each node until the breakpoint is reached. In this case, the breakpoint is encountered in node D_1, since the start of service in this node cannot be advanced in a feasible manner, i.c. due to the lower time window bound of 70. This implies that any remaining travel time shortages before the breakpoint may only be solved by advancing the start of service in all preceding nodes, including the start depot. Shortages after the breakpoint may only be solved by advancing the start of service in all following nodes, including the end depot. According to this

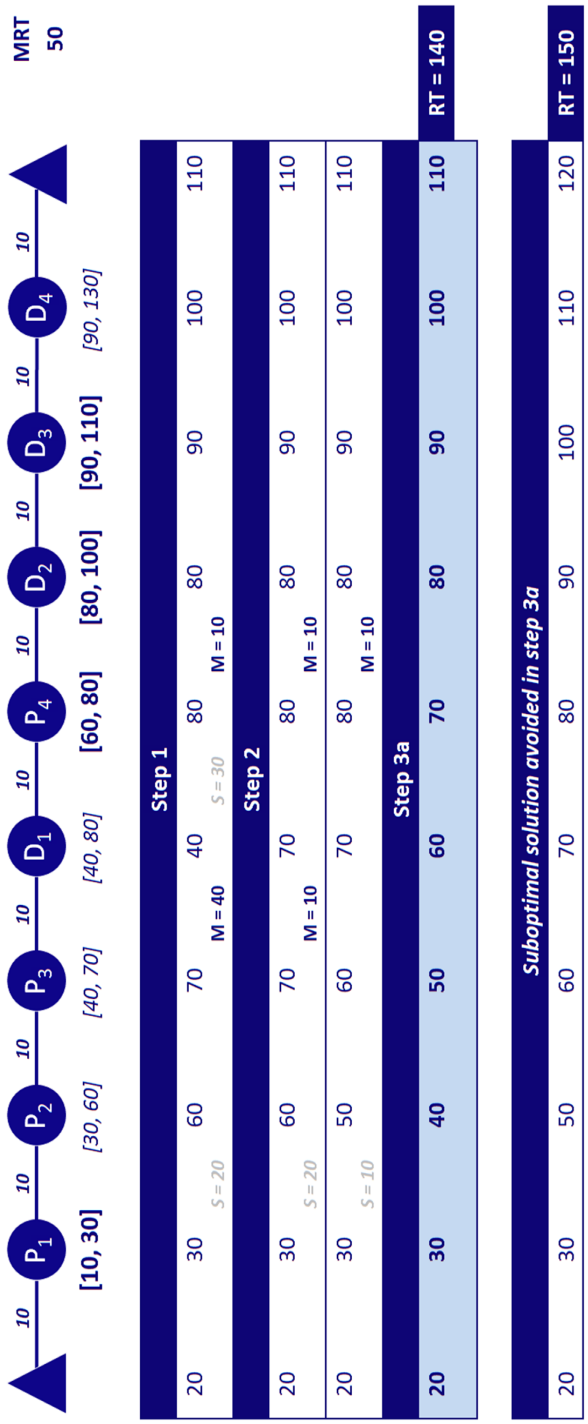

Figure 5.4: Example route for which the proposed procedure finds the optimal schedule in step 3a.

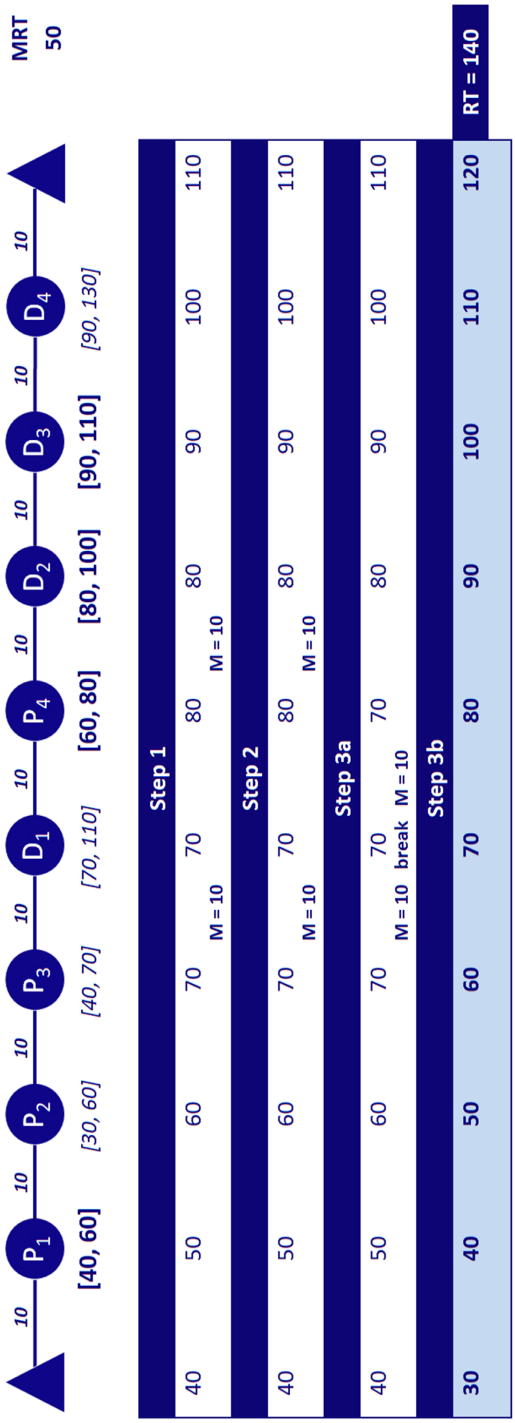

Figure 5.5: Example route for which the proposed procedure finds the optimal schedule in step 3b.

strategy, an optimal schedule can be found for the example route.

In general, if the procedure is not aborted during step 3, a feasible schedule is returned as the final solution, but the result is not necessarily optimal. Particularly the interplay of time windows and user ride times complicates the decision of which slack time should be used to eliminate a travel time shortage, although such exceptional shortcomings do not impede the practical applicability of the scheduling procedure.

5.3 Scheduling procedure with additional waiting constraint

The heuristic in Section 5.2 has the free choice of assigning slack time, i.c. time during which the vehicle idles, anywhere in the route. However, from the perspective of users aboard the vehicle, waiting may be unacceptable in particular cases. If the vehicle arrives at another user's pickup node before the start of the corresponding time window, it is understandable that the user to be picked up is not yet available. However, due to the structure of the scheduling procedure, waiting time prior to the pickup of another user may also be assigned within the corresponding time window, since this reduces his ride time. This may be inconceivable, both for users aboard the vehicle and for the user to be picked up. Therefore, the heuristic of Section 5.2 is adapted, imposing that a *non-empty* vehicle should service a pickup node immediately upon arrival in this node or, in case the vehicle arrives early, at the beginning of its time window. This constraint does not apply to an empty vehicle, since the driver may alternatively wait at any location on the arc that precedes the pickup node.

The heuristic in Section 5.2 has a desirable structure, since the slack time on arcs never increases throughout the subsequent steps. Rather, waiting time is gradually removed to remedy shortages in travel time on other arcs. Therefore, starting from a feasible schedule obtained by the standard procedure, a simple ex post intervention is appropriate to incorporate the additional waiting restriction.

Step 4

The standard procedure never schedules unnecessary waiting time before a delivery node. Consequently, only pickup nodes that immediately succeed an arc on which the vehicle is loaded need to be considered in a forward loop through the route. If the start of service in such a node violates the additional waiting constraint, it must be advanced to the maximum of (1) the lower bound of the time window in this pickup node and (2) the arrival time of the vehicle. In other words, if arc ij ($j \in P$) is traversed by a loaded vehicle, B_j should equal the maximum of e_j and $B_i + d_i + t_{ij}$. The same adaptation is done for all succeeding pickup or delivery nodes until the next arc traversed by

an empty vehicle. It should be stressed that this extension may cause an increase in user ride times or, in the worst case, even an infeasible route if a user's maximum ride time cannot be respected. Therefore, while proceeding through the route, the maximum ride time constraint should again be checked upon reaching the destination node of any user whose pickup was advanced. The impact of the additional waiting restriction is investigated in Section 5.4.4.

This additional step would impact the total ride time in the example route in Figure 5.2. In the optimal solution after step 2, the vehicle spends 10 minutes waiting at the pickup location of user 2 while user 1 is aboard the vehicle. In step 4, this schedule would be corrected to the suboptimal solution shown in the lower part of Figure 5.2.

5.4 Computational experiments

5.4.1 Experimental setting

The performance of the proposed scheduling procedure needs to be assessed on three criteria. First, incorrect infeasibility declarations should be avoided as much as possible. They obstruct the proper functioning of a solution approach, since feasible solutions are discarded whenever one of the routes involved cannot be scheduled. Second, deviations from the minimal total user ride time should remain limited. This is a prerequisite for bi-objective approaches like the one presented in Chapter 4 and contributes to a quality-oriented service provision. Third, the required computation time must be reasonable, as the scheduling procedure may be invoked millions of times throughout a solution procedure. Based on these performance criteria, both variants of the proposed scheduling heuristic are compared with the existing procedures of Cordeau and Laporte (2003) and Parragh et al. (2009). To ensure a fair comparison, all procedures have been coded in the same programming language, which is Python 3.4. The computational resources and services used were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation Flanders (FWO) and the Flemish Government, department EWI. All experiments are performed using Xeon E5-2680v2 CPUs at 2.8 GHz with 64 GB of RAM.

A thorough comparison of the scheduling heuristics should be performed on a large set of routes, preferably originating from various instances. This set is constructed by applying 20,000 iterations of the LNS metaheuristic presented in Chapter 6 to all instances of Røpke et al. (2007) and Cordeau and Laporte (2003) discussed in Chapter 2. All of the feasible route proposals encountered during the solution procedure are saved once, resulting in a total number of 113,007,097 unique routes.

5.4.2 Results

The results of the tests are shown in Table 5.1. The three parts compare the scheduling procedures on the aforementioned performance criteria. The first column indicates the set of instances used. The second column gives the total number of routes originating from solving these instances. As explained before, all of these routes are known to be feasible. The third column computes the percentage of routes that cannot be scheduled because of an incorrect infeasibility declaration. The fourth column mentions the percentage of routes exhibiting a total user ride time as small as possible, whereas the fifth column shows the percentage of routes for which a deviation is found. The latter category relates to feasible schedules which entail a suboptimal total user ride time. As this requires the optimal value to be known, a linear programming problem minimizing the total user ride time subject to all feasibility constraints is solved using CPLEX 12.6. This exact scheduling procedure is too slow to be applied beyond testing purposes, which is the very reason why efficient scheduling heuristics are needed. The average deviation over all feasible routes is indicated in the fifth and the sixth column, expressed in absolute terms (in minutes) and in relative terms. The last column indicates the average required CPU time per route (in milliseconds). For every scheduling procedure, the results over the different data sets are averaged in bold. The averages are weighted in such a manner that every single route has an equal contribution. Figure 5.6 visualizes the average performance of all three scheduling procedures.

As expected, the procedure of Cordeau and Laporte (2003) composes a feasible schedule for every route in the set, but the percentage of routes for which the optimal total user ride time is found amounts to only 16.61%. Among both procedures that aim at minimizing total user ride time, the procedure of Parragh et al. (2009) delivers twice as many incorrect infeasibility declarations as the proposed heuristic: 23.71% versus 10.67%. In addition, the proposed heuristic matches the optimal schedule in 83.25% of the cases, whereas this is only 39.84% for Parragh et al. (2009).

Since the latter two scheduling procedures aim at improving the service quality, it is interesting to investigate whether the average improvement in total user ride time justifies the fact that incorrect infeasibility declarations may be encountered. Table 5.1 shows that the average deviation from the optimal total user ride time reduces from 33.68 minutes for Cordeau and Laporte (2003) to 7.18 and 0.54 minutes for Parragh et al. (2009) and the proposed heuristic, respectively. This suggests that minimizing total user ride time is worthwhile from a quality-related perspective. However, the deviations should be interpreted with caution, since they are computed for slightly different sets of routes due to the exclusion of different sets of infeasible routes. A more comprehensive comparison is provided in Table 5.2, which focuses on both procedures that aim at minimizing total user ride

Instances	Number of routes	Percent. infeasible	Percent. optimal	Percent. deviating	Absolute dev. (m)	Relative dev. (%)	CPU time / route (ms)
Scheduling procedure Cordeau and Laporte (2003)							
Røpke a	14,061,229	0.00%	13.39%	86.61%	25.67	20.10%	1.08
Røpke b	13,691,368	0.00%	7.40%	92.60%	57.45	46.72%	0.74
C&L Ra	33,050,801	0.00%	13.43%	86.57%	34.54	10.35%	0.85
C&L Rb	52,203,699	0.00%	21.90%	78.10%	29.05	7.85%	0.93
Average		**0.00%**	**16.61%**	**83.39%**	**33.68**	**14.81%**	**0.90**
Scheduling procedure Parragh et al. (2009)							
Røpke a	14,061,229	9.71%	45.10%	45.19%	3.42	2.12%	1.15
Røpke b	13,691,368	7.68%	56.27%	36.05%	4.13	2.43%	1.11
C&L Ra	33,050,801	25.43%	35.11%	39.45%	7.78	2.00%	1.20
C&L Rb	52,203,699	30.59%	37.11%	32.30%	8.63	2.13%	1.29
Average		**23.71%**	**39.84%**	**36.45%**	**7.18**	**2.13%**	**1.22**
Proposed procedure							
Røpke a	14,061,229	3.14%	96.85%	0.01%	0.00	0.00%	0.56
Røpke b	13,691,368	2.41%	97.32%	0.27%	0.01	0.00%	0.53
C&L Ra	33,050,801	11.62%	80.37%	8.00%	0.59	0.12%	0.65
C&L Rb	52,203,699	14.26%	77.71%	8.03%	0.79	0.16%	0.74
Average		**10.67%**	**83.25%**	**6.08%**	**0.54**	**0.11%**	**0.66**

Table 5.1: Overall performance of the proposed scheduling procedure compared with the procedures of Cordeau and Laporte (2003) and Parragh et al. (2009).

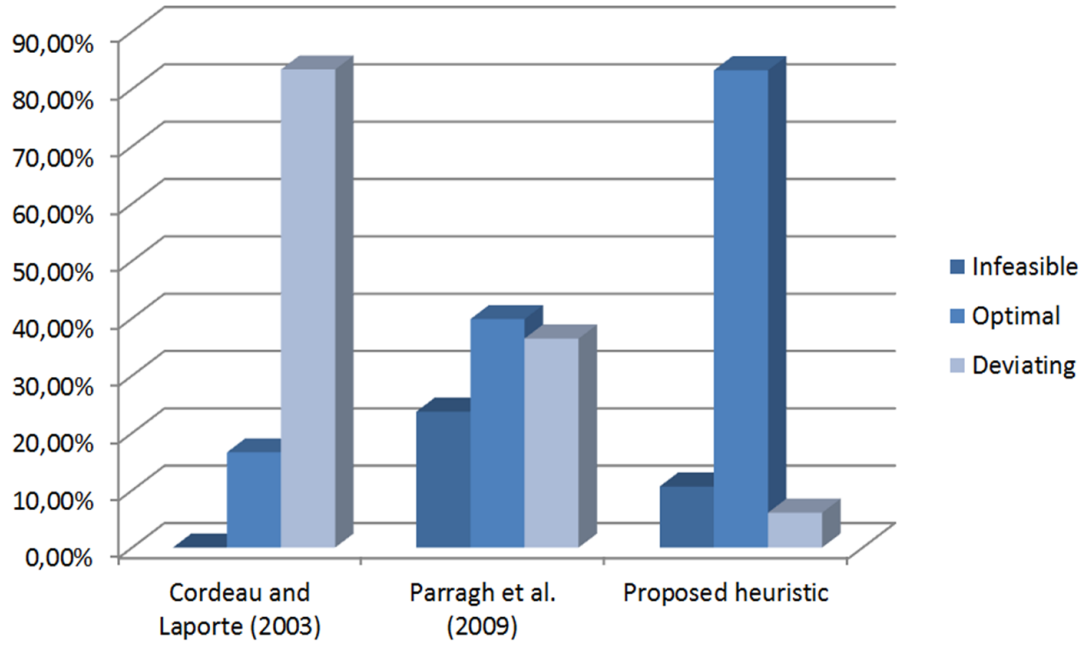

Figure 5.6: Overall performance of the proposed scheduling procedure compared with the procedures of Cordeau and Laporte (2003) and Parragh et al. (2009).

time. For each route in the set, it is assessed which of both procedures - if any - performs better. A procedure outperforms the other if (1) it delivers a feasible schedule and the other does not or (2) both procedures deliver a feasible schedule and the total user ride time of the first is lower. The result is assumed equal if no procedure is outperformed, i.c. when (1) both deliver an infeasible schedule or (2) both deliver feasible schedules with an identical total user ride time. The proposed procedure outperforms Parragh et al. (2009) for 48.91% of the routes (55,275,883 cases), while the opposite applies to a mere 2.72% of the routes (3,070,022 cases). A *sign test with matched samples* (Anderson et al., 2010) indicates that the corresponding p-value is smaller than 0.001 when equal distributions are assumed. In other words, the proposed scheduling procedure significantly outperforms Parragh et al. (2009) on the given set of routes.

Nevertheless, the proposed scheduling procedure still fails to construct a feasible time schedule for 10.67% of the routes. It is worth investigating whether the percentage of incorrect infeasibility declarations can be reduced through a combination of both procedures. Due to their completely different fundamental structure, they may fail on different types of routes. If this is the case, routes may only be discarded if both procedures cannot find a feasible schedule. Put differently, incorrect infeasibility declarations may be avoided at the cost of additional computation time. Unfortunately, this hypothesis does not hold. The rightmost column of Table 5.2 indicates that neither scheduling procedure can obtain a feasible schedule for 9.91% of the routes, which is hardly less than for the proposed procedure on its own. Hence, it can be concluded that the proposed scheduling procedure most often fails on routes that are also declared infeasible by the procedure of Parragh et al. (2009). For these routes, a feasible schedule can only be found using the procedure of Cordeau and Laporte (2003) or any other non-quality-oriented scheduling procedure in the literature. However, this may result in unnecessarily large user ride times. Another effective option is to minimize the total user ride time while allowing slight constraint violations, as will be illustrated in Section 5.4.3.

The patterns observed in Table 5.1 are consistent among all sets of instances. Yet, the absolute percentage of incorrect infeasibility declarations is different. Routes originating from the instances of Cordeau and Laporte (2003) turn out to be more difficult to schedule. This is explained by structural differences in the instances, which are summarized in Table 5.3. First, the data of Cordeau and Laporte (2003) are characterized by wider time windows and larger maximum user ride times, which enables more combinations of multiple requests. The users are more often combined in the same vehicle, rather than served one after the other. These routes are harder to schedule, since the maximum ride time may be a binding constraint for many users. Second, the instances of Cordeau and Laporte (2003) have a larger ratio of vehicle capacity and average load per request, which again increases the number of possible request combinations.

Instances	Better	Worse	Equal	Infeasible
Røpke a	51.77%	0.00%	48.23%	3.14%
Røpke b	41.29%	0.03%	58.68%	2.41%
C&L Ra	52.28%	2.98%	44.74%	11.04%
C&L Rb	48.01%	3.98%	48.00%	12.98%
Average	**48.91%**	**2.72%**	**48.37%**	**9.91%**

Table 5.2: Performance of the proposed scheduling procedure relative to the procedure of Parragh et al. (2009).

Instances	TW width	Max RT	Capacity	Load/req
Røpke a	15 min.	30 min.	3	1
Røpke b	15 min.	30 min.	6	$U(1,6)$
C&L Ra	30 min.	90 min.	6	1
C&L Rb	60 min.	90 min.	6	1

Table 5.3: Characteristics of the artificial data sets.

Finally, the results in Table 5.1 show that the proposed scheduling procedure is twice as fast as Parragh et al. (2009), which is explained by the fact that it avoids calculations of the forward time slack for each pickup node. In addition, the proposed heuristic can be aborted as soon as a feasible schedule is found, whereas Parragh et al. (2009) have to complete their entire procedure.

5.4.3 Sensitivity analysis

Thus far, time windows and maximum user ride times were considered as hard constraints, such that any violation causes a route to be infeasible. This approach does not provide any insight into the extent of infeasibility. Yet, service providers may prefer a schedule with minimized total user ride time and small constraint violations to a feasible schedule exhibiting a large deviation from the optimal total user ride time. Therefore, it is interesting to analyze how many incorrect infeasibility declarations are avoided if the width of the time windows (in the users' critical nodes) and/or the maximum user ride time increase by a limited amount of time. Figures 5.7 and 5.8 visualize this sensitivity analysis for the procedure of Parragh et al. (2009) and for the proposed procedure, respectively. The axes in the horizontal plane indicate the allowed exceedance of both constraints. Both the additional time window width (TW) and the additional allowed ride time (RT) are expressed in

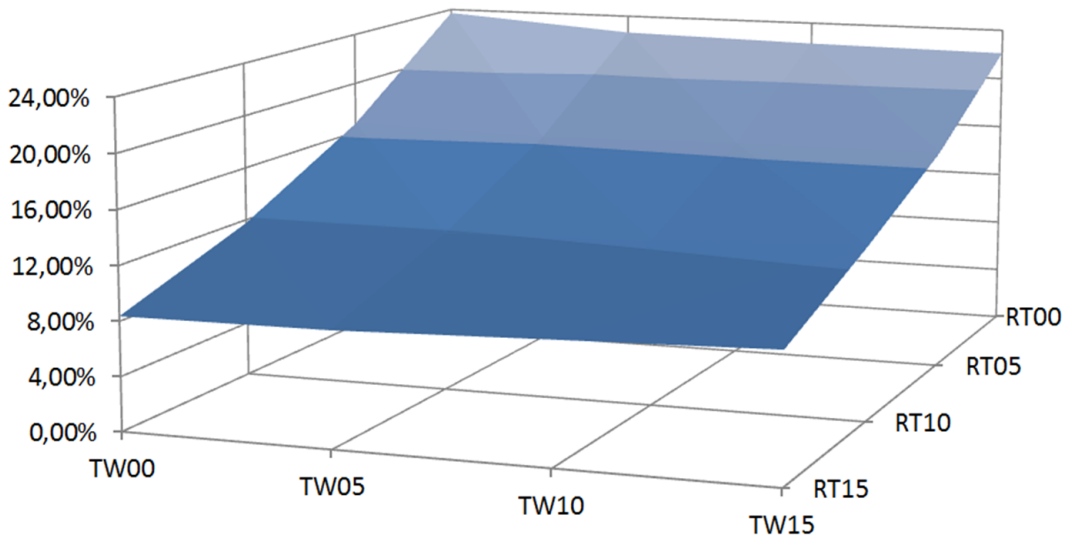

Figure 5.7: Effect of allowing constraint violations (in minutes) on the percentage of incorrect infeasibility declarations for the procedure of Parragh et al. (2009).

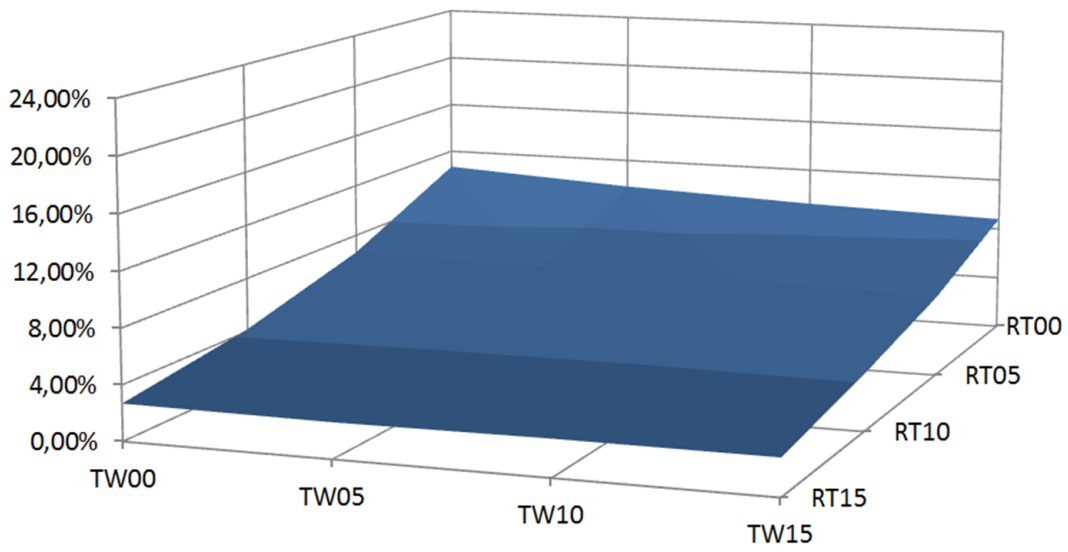

Figure 5.8: Effect of allowing constraint violations (in minutes) on the percentage of incorrect infeasibility declarations for the proposed scheduling procedure.

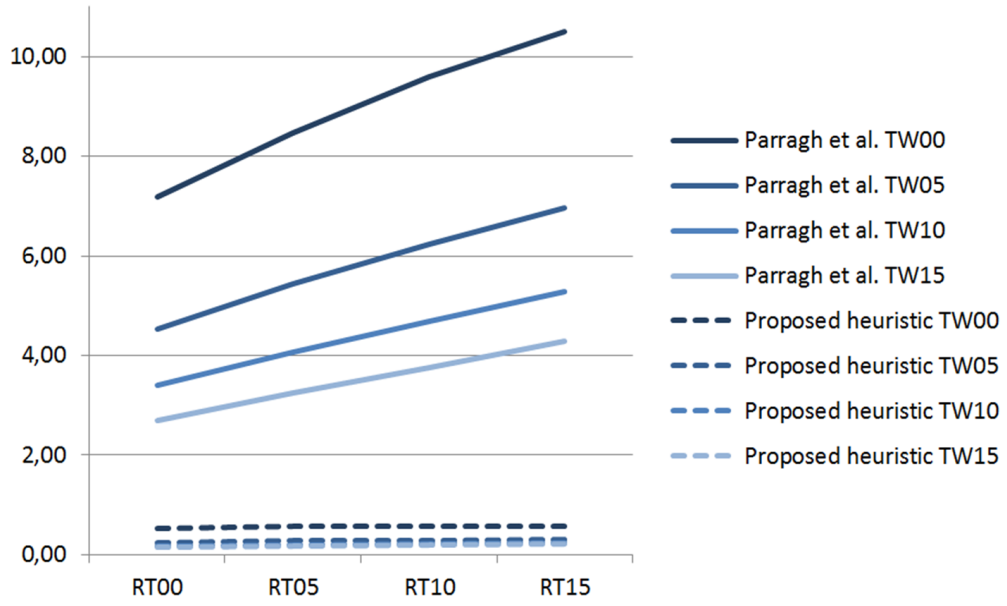

Figure 5.9: Effect of allowing violations of the maximum user ride time (in minutes) on the average deviation from the optimal total user ride time (in minutes), shown for different levels of allowed time window violations (in minutes).

minutes. The vertical axis shows the resulting percentage of incorrect infeasibility declarations. On the one hand, increases of the maximum user ride time have a considerable impact. For example, allowing an additional 5 minutes detour avoids approximately one third of the incorrect infeasibility declarations. On the other hand, enlarging the time windows hardly contributes to this objective. In combination with larger ride times, wider time windows even encourage incorrect infeasibility declarations due to an increased difficulty.

It is clear that small violations of the maximum user ride time are very appropriate to reduce incorrect infeasibility declarations by the scheduling procedures minimizing the total user ride time. However, this entails the risk of increasing the average deviation from the optimal total user ride time, which is contrary to the objective of the procedures. Figure 5.9 shows the average deviation as the allowed violation of the maximum user ride time is increased. The deviations are computed relative to the optimal schedules *without* constraint violations[2]. For the procedure of Parragh et al. (2009), the deviation from the optimal total user ride time increases as the allowed violation of the

[2]A zero deviation is registered for schedules with constraint violations that improve the optimal total user ride time without constraint violations.

maximum user ride time becomes larger. Contradictory, such a sacrifice in service quality allows to reduce the percentage of incorrect infeasibility declarations and hence to obtain more routes with efficient user ride times. For the proposed procedure, a negligible deviation from the optimal total user ride time is found for all constraint violations within the investigated range, which implies that these constraint violations can serve as an effective tool to reduce incorrect infeasibility declarations.

5.4.4 Effect of additional waiting constraint

Section 5.3 questioned the acceptability of waiting time prior to the pickup of an available user, particularly when still other users are aboard the vehicle. The impact of avoiding such undesirable waiting time is analyzed in Table 5.4, using the same set of routes as before. Part of these routes can no longer be scheduled in the presence of this additional waiting constraint. This percentage of *correct* infeasibility declarations is identified using an adapted version of the scheduling procedure of Cordeau and Laporte (2003). The forward time slack is only computed for pickup nodes that succeed an empty arc, rather than for every pickup node in the route. Such an approach delivers a feasible schedule whenever one exists, subject to the additional waiting restriction. The upper part of Table 5.4 shows that imposing this restriction causes 21.48% of the routes to be *correctly* infeasible. The feasible routes exhibit an average total ride time deviation of 34.65 minutes or 15.70%, expressed relative to the optimal schedule *without* waiting restrictions. The deviation is large, but comparable with the original version of the scheduling procedure of Cordeau and Laporte (2003), as both versions do not aim at minimizing the total user ride time.

In the lower part of Table 5.4, the routes are subjected to the procedure with waiting restrictions proposed in Section 5.3. These results show that 22.14% cannot be scheduled in a feasible manner. Given that 21.48% of the routes are indeed infeasible, only 0.66% of the routes give rise to incorrect infeasibility declarations. The presence of the waiting constraints actually simplifies the scheduling problem, since the choice in allocating waiting time is more limited. This explains why the proposed scheduling heuristic rarely fails. The feasibly scheduled routes exhibit an average total ride time deviation of 7.61 minutes or 1.83%, expressed relative to the optimal schedule *without* including waiting restrictions. This is much smaller than for the adapted scheduling procedure of Cordeau and Laporte (2003). However, compared with the standard version of the proposed procedure in Table 5.1, which resulted in deviations of 0.54 minutes and 0.11%, the additional waiting constraint turns out to have a considerable undesirable impact on the ride times experienced by users. Finally, note that computation times are somewhat larger than for the adapted scheduling procedure of Cordeau and Laporte (2003). The latter needs to compute fewer forward time slacks, whereas the proposed procedure is extended with an additional step to enforce the additional waiting restrictions.

Instances	Number of routes	Percent. infeasible	Percent. optimal	Percent. deviating	Absolute dev. (m)	Relative dev. (%)	CPU time / route (ms)
Cordeau and Laporte (2003) with additional waiting restriction							
Røpke a	14,061,229	4.09%	25.72%	70.19%	24.05	19.51%	0.87
Røpke b	13,691,368	4.16%	8.95%	86.89%	57.58	47.69%	0.65
C&L Ra	33,050,801	26.68%	12.17%	61.14%	38.29	12.23%	0.55
C&L Rb	52,203,699	27.41%	20.41%	52.17%	29.19	8.47%	0.61
Average		**21.48%**	**17.27%**	**61.25%**	**34.65**	**15.70%**	**0.63**
Proposed scheduling procedure with additional waiting restriction							
Røpke a	14,061,229	4.09%	90.46%	5.45%	0.19	0.11%	0.68
Røpke b	13,691,368	4.16%	78.40%	17.44%	1.58	0.80%	0.65
C&L Ra	33,050,801	27.18%	32.23%	40.59%	11.72	2.78%	0.77
C&L Rb	52,203,699	28.52%	42.07%	29.41%	8.60	1.95%	0.87
Average		**22.14%**	**49.61%**	**28.25%**	**7.61**	**1.83%**	**0.79**

Table 5.4: Effect of the additional waiting constraint on the schedules delivered by the procedure of Cordeau and Laporte (2003) and by the proposed procedure.

5.5 Conclusions and future research

This chapter proposes a new scheduling heuristic to assess the time feasibility of routes while optimizing the service level experienced by users. The procedure's fundamental structure aims at minimizing the total user ride time. It starts from a schedule in which the objective value matches its lower bound with respect to the given time windows, allowing the scheduled travel times between pickup and delivery nodes to be insufficient. Travel time shortages are gradually eliminated in such a manner that ride time increases remain as small as possible. The procedure may be terminated as soon as a feasible schedule is reached, which is efficient from a computational point of view. Relative to a comparable scheduling heuristic from the literature, solutions match the optimal schedule more frequently, remaining deviations are smaller and the risk of incorrect infeasibility declarations strongly reduces. Including the proposed scheduling heuristic into solution algorithms would allow service providers to significantly improve their service level through reduced ride times and faster response times. Moreover, an extension is proposed to avoid waiting at pickup locations while the user is available, should this be desirable. However, this additional constraint considerably impacts the feasibility of routes.

The discussion of two variants of the proposed scheduling procedure, allowing waiting time prior to the pickup of an available user or not, illustrates a shortcoming of the time-related constraints defined for the standard DARP. The current approach fundamentally distinguishes between the waiting time that a user needs to spend in his critical node and the time lost through detours made by the vehicle. Consider the example of a user who returns home after attending a certain activity. Without an extra waiting constraint, the scheduling procedure may assign (1) waiting time before the departure and (2) additional ride time through detours, as long as the predefined time window width and maximum user ride time are respected. With an extra waiting constraint, waiting time before the departure may no longer be allowed. In such case, a solution in which the user reaches his home location at exactly the same time as in the first scenario may violate the maximum ride time and therefore be infeasible. This example illustrates why it seems advisable to rethink the current approach of time-related constraints and create corresponding scheduling procedures. From a technical point of view, it should be irrelevant whether a user loses a certain amount of time waiting for the departure or making detours. From a quality-related point of view, users may even prefer a longer detour while being seated in the vehicle to a longer waiting time outside. This assumption may be confirmed by means of user surveys, which are currently quite uncommon in the literature.

Such surveys may also reveal other time-related aspects to which users attach importance. For example, the current approach only imposes maximum losses of time that users may incur while waiting and while being transported in the vehicle. Every schedule that falls within these bounds is assumed to be feasible from a technical point of view. However, different schedules may also cause different service experiences among users. Even two schedules with an equal total user ride time may considerably differ with respect to their perceived fairness, e.g. the distribution of detours among the users. The skewness of this distribution may impact the acceptability of a schedule among the users receiving a lesser service level, although no research on this issue has been performed yet.

The comparison of the scheduling heuristics discussed in this chapter offers even more opportunities for future research. For example, it would be useful to analyze the results of the different procedures at the level of individual routes, particularly to discover whether similarities could be found in the routes for which incorrect infeasibility declarations are made. This would be the first step in the creation of a scheduling procedure that minimizes the total user ride time with no risk of incorrect infeasibility declarations. Furthermore, the deviations from the optimal schedules may still be reduced, although the proposed heuristic already made a strong contribution in this respect. Finally, this chapter performs computational tests on feasible routes, but it may also be interesting to assess the behavior of the different scheduling procedures on infeasible routes. All of them will produce an infeasibility declaration, but the required time to reach this conclusion is also a relevant

assessment criterion, since both feasible and infeasible routing proposals may be encountered by a solution algorithm for dial-a-ride problems.

Chapter 6

Operational effects of joint route planning

— *Short summary* —

This chapter investigates the operational effects of centralized decision making through joint route planning among dial-a-ride providers. The current practice is that users choose a particular service provider to submit their requests. Providers operating in the same area create their vehicle routes independently of each other, given their own set of user requests. As will be explained in Section 6.1, joint route planning implies that providers cooperate by exchanging requests in order to minimize their overall operating costs. Section 6.2 presents an adaptation of a well-known large neighborhood search algorithm, which can be used to compare the operational costs in scenarios with and without joint route planning. Section 6.3 quantifies the resulting benefits on artificial and real-life data. Operational characteristics influencing the magnitude of the savings are identified. Statistical analyses confirm that the potential of joint route planning is larger for variants with smaller time windows, more long-distance trips, weaker clustering of users around their closest depot, more overlap between service areas or fewer users. In addition, the reasons why service providers benefit from certain types of request exchanges are investigated, as well as the extent to which exchanges are predictable. This reveals that sharing complete information on user requests is a prerequisite for service providers to exploit the benefits of joint route planning. Based on these results, Section 6.4 draws conclusions in view of a practical implementation and suggests ideas for future work.

6.1 Introduction

When a certain area is served by multiple dial-a-ride providers with a comparable quality policy, most users living in that area will intuitively submit their requests to the provider located closest to their homes. The different providers create their vehicle routes and schedules independently of each other, based on the requests they receive. Whereas strategies to share information or resources have been cost-effective practices among logistic service providers for many years (Verdonck et al., 2013), they are completely unexplored in the domain of dial-a-ride services. Also the academic literature lacks insights into the effects of cooperation in the specific context of demand-responsive passenger transportation.

In general, Cruijssen et al. (2007b) describe *horizontal cooperation* in logistics as the identification and exploitation of win-win situations by companies operating at the same level of the supply chain. The participating parties may apply various strategies to meet their expectations of such a cooperation, which are often related to an increased productivity for core activities (Cruijssen et al., 2007c). Verdonck et al. (2013) present a detailed classification of cooperation techniques in freight transportation. This chapter translates one of these techniques, being *joint route planning*, to the related domain of demand-responsive passenger transportation. Joint route planning is a particular type of request sharing, which means that service providers may decide to exchange their users in order to align their available resources with the demand of users. Providers retain their own set of vehicles and depots, but routes are designed in an overall efficient manner, which implies that users may be served by any given provider. Such a centralized decision making enables a better capacity utilization and savings in transportation costs, e.g. thanks to reductions in unproductive distance traveled by unloaded vehicles (Cruijssen et al., 2007a). In logistics, joint route planning has been applied to a pickup and delivery problem with time windows (PDPTW) by Krajewska et al. (2008). Their computational tests on both artificial and real-life data reveal cost savings between 16% and 31% obtained by the participating carriers. Although the PDPTW and the DARP are interrelated, the latter is more tightly constrained because of the maximum user ride time. The impact of this additional constraint is difficult to predict. On the one hand, it reduces the operational flexibility of service providers and therefore increases the benefits of scaling through cooperation. On the other hand, it may impede efficient user combinations when cooperating.

The problem studied in this chapter involves multiple service providers which may or may not cooperate through joint route planning. Since each participating provider disposes of at least one depot, a multi-depot variant of the DARP (Braekers et al., 2014) needs to be solved. In the scenarios with joint route planning, requests may be served by a vehicle originating from any depot. Without

cooperation, requests are preassigned to a specific provider, based on the user's geographical situation. All other problem characteristics correspond to the standard DARP of Cordeau and Laporte (2003) that was introduced in Chapter 2. Routing solutions with and without joint route planning are obtained using an adjusted large neighborhood search (LNS) metaheuristic whose performance is demonstrated on common benchmark data from the literature.

The contribution of this chapter is threefold. First, the potential of centralized decision making through joint route planning among dial-a-ride providers is analyzed. Second, the operational characteristics that influence these benefits are identified using artificially constructed data in which the operational setting is varied. Third, a real-life case study is performed to discover the reasons why providers benefit from exchanging certain requests. The predictability of these exchanges influences the extent to which information must be disclosed in order to enable a successful cooperation. Note that all analyses focus on the joint operational benefits incurred by the overall cooperation. To allocate these benefits among the providers, which falls outside the scope of this chapter, gain sharing techniques from existing literature on horizontal cooperation (e.g. Shapley value, alternative cost avoided method, equal profit method) may be used (Verdonck et al., 2016).

6.2 Algorithm

The algorithm used in this chapter is based on an efficient implementation of a large neighborhood search (LNS) framework by Pisinger and Røpke (2010), extended with an additional periodic diversification phase. It combines well-known operators from the literature and specific components that are useful to solve the problem at hand. The algorithmic structure is summarized in Algorithm 2 and the design choices are explained below.

The algorithm requires an initial solution s_{init}, constructed by a random order insertion heuristic. This solution is copied to initialize s_{cur}, s_{best} and s_{ovr}. In each iteration, s_{cur} represents the solution on which operations are performed and s_{best} is the best solution found during the current search phase, which ends at the next periodic diversification. By contrast, s_{ovr} represents the best solution found throughout the entire procedure. Each of the n_{iter} iterations starts with the choice of a destroy operator. The four operators introduced in Section 6.2.1 may be selected with equal[1] probabilities. The selected operator is executed until a random percentage of requests, situated in the interval $[0.01, des_{max}]$, has been removed from s_{cur}. Then, a repair operator is chosen in order to gradually recomplete s_{cur}. The three operators discussed in Section 6.2.2 may be selected with

[1] Using a basic adaptive large neighborhood search, in which the probability of selecting a certain operator depends on its past performance, did not improve the results.

equal probabilities. In order to focus on promising parts of the search area, the newfound objective value is not allowed to deteriorate excessively. Specifically, it cannot exceed det_{max} times the objective value of s_{best}. This approach corresponds to the record-to-record strategy of Dueck (1993). If the solution cannot be completed without violating this threshold or any feasibility constraint, s_{cur} is replaced with s_{best} and the current iteration is terminated. Otherwise, the complete s_{cur} is intensified using two local search operators which never deteriorate the objective value, as explained in Section 6.2.3. Afterwards, s_{best} and s_{ovr} are updated if necessary. After f_{div} iterations without improvement of s_{best}, a periodic diversification is applied with the intention of escaping the local optimum, as described in Section 6.2.4. The resulting s_{cur} serves as the first s_{best} of the next search phase.

Even though the LNS algorithm in this chapter is embedded in a different metaheuristic framework than the VND applied in Chapter 4, both implementations show substantive similarities as they include similar operators and algorithmic components. For example, they rely on an interaction between (1) a diversification phase in which a considerable part of the solution is destroyed and reconstructed and (2) an improvement phase in which the solution is intensified using tailored local search operations on a small scale (e.g. relocate, exchange natural sequences). The fact that both solution techniques are able to reach high-quality results supports the claim in Chapter 2, stating that the choice of effective local search operators is a more decisive performance determinant than the metaheuristic framework as such.

6.2.1 Destroy operators

In every iteration, one of the following destroy operators is executed until the required destroy percentage is reached. The first three operators are based on existing strategies from the literature (Røpke and Pisinger, 2006), whereas the fourth operator is a new contribution.

- The *random removal* operator removes random requests from the solution. All requests have equal probabilities to be selected.

- The *worst removal* operator involves a biased random selection procedure that considers the saving obtained by removing a request. Removing undesirably positioned requests may cause considerable improvements of the objective value. Following the roulette wheel principle, the probability of removing request $i \in R$ equals $a_i^2 / \sum_{r \in R} a_r^2$, with a_i denoting the corresponding distance saving and R the set of requests. A probalistic selection procedure is preferred to a deterministic variant in order to encourage diversification.

Algorithm 2 - Structure of the large neighborhood search algorithm

```
set parameters: n_iter, des_max, det_max, f_div

construct s_init
initialize s_cur, s_best, s_ovr ← s_init
set i_impr = 0   (iteration of last improvement s_best)

for i = 1 → n_iter do

  if i = i_impr + f_div then
    s_cur, s_best ← periodic diversification on s_ovr
    i_impr = i
  end if

  select p = rand_(0.01,des_max)
  randomly select D_i ∈ {D_rand, D_worst, D_rel, D_prox}
  randomly select R_i ∈ {R_rand, R_gree, R_regr}
  s_cur ← R_i(D_i(s_cur, p))

  if s_cur complete and f(s_cur) ≤ f(s_best) * det_max then
    s_cur ← relocate on s_cur
    s_cur ← exchange natural sequences on s_cur

    if f(s_cur) < f(s_best) then
      s_best ← s_cur
      i_impr = i
      if f(s_cur) < f(s_ovr) then
        s_ovr ← s_cur
      end if
    end if

  else
    s_cur ← s_best
  end if

end for

return s_ovr
```

- The *related removal* operator involves a biased random selection procedure that takes into account spatial and temporal similarities between requests. Removing interchangeable requests increases the probability of finding a different solution. After removing a first randomly chosen request i, the similarity between this request and each other request j is computed as:

  $$b_{i,j} = \left(\frac{d_{i,j} + d_{i+n,j+n}}{d_{max}} + \frac{|T_i - T_j| + |T_{i+n} - T_{j+n}|}{t_{max}} \right)^{-1}$$

$d_{a,b}$	Distance between nodes a and b
d_{max}	Length of the longest arc in the network
T_a	Center of the time window of node a
t_{max}	Difference between the largest upper time window bound and the smallest lower time window bound over all nodes
$i, i+n$	Origin and destination nodes of request i

  Consequently, for all of the following request removals within the same iteration, the probability of removing request $j \in R$ equals $b_{i,j}^2 / \sum_{r \in R} b_{i,r}^2$, with R denoting the set of remaining requests. Thanks to the small time windows involved with DARPs, the actual service time in a node can accurately be approximated by the center of the corresponding time window. As a result, the similarities between requests are static and can be precomputed, which is computationally more efficient than the classical related removal strategy and makes the similarities independent of the scheduling procedure used.

- The *proximity removal* operator is based on a biased random selection procedure that favors request removals in areas where routes overlap in time and space, since enhancing the interaction among routes may be interesting to facilitate joint route planning in the context of this chapter. Thus, the probability of removing request $i \in R$ equals $c_i^2 / \sum_{r \in R} c_r^2$, with c_i denoting the similarity between request i and the most similar request in a *different* route. Although similarity is measured as for the related removal operator, both strategies are very different. Related removal starts from a single random request and intends to remove requests that are interchangeable with this first request. Proximity removal does not require relatedness between the selected requests, but focuses on requests which are each individually related to a request served in another route.

The aforementioned operators were selected after testing their contribution to the solution quality of the algorithm, as will be illustrated in Section 6.3.1. These tests also reveal that a route removal operator is rather unappropriate within the current framework, despite its common application in the literature. It tends to result in new solutions whose objective value exceed the allowed threshold.

6.2.2 Repair operators

In each iteration, one of the following repair operators is executed until either a complete solution is reached or no insertion can be performed without exceeding the maximum deterioration threshold or violating a feasibility constraint. The operators are based on common repair strategies from the literature (Røpke and Pisinger, 2006).

- The *random order insertion* operator repeatedly performs the optimal insertion of a randomly selected request, checking all insertion positions in any route. Requests have equal probabilities to be selected.
- The *greedy insertion* operator repeatedly performs the best insertion over all requests, considering all insertion positions in any route.
- The *2-regret insertion* operator repeatedly performs the optimal insertion of the request having the largest regret value, considering all insertion positions over all routes. This regret value is defined as the difference between the insertion cost at its second-best insertion position and the insertion cost at its best insertion position. This strategy recognizes that postponing the insertion of requests having a large regret value may deteriorate the eventual solution quality. The second-best insertion position is assumed to be in a different route than the best one, such that both positions cannot simultaneously be eliminated by the insertion of another request.

A random noise factor in the interval [0.95, 1.05] is applied to encourage variation in the choice of the insertion positions and routes. The scheduling procedure of Cordeau and Laporte (2003) will be used to perform time-related feasibility checks on the artificial data in Sections 6.3.1 and 6.3.2, which ensures the comparability with other solution approaches. The real-life case in Section 6.3.3 invokes the scheduling procedure introduced in Chapter 5, which is more in line with the provider's quality policy.

6.2.3 Additional local search

Two additional local search operators are applied to repaired solutions whose objective value does not exceed the maximum deterioration threshold. The *relocate* operator considers all requests in a random order and moves them to the best position in any route. In the worst case, a request is reinserted at its original position, which implies that this operation cannot deteriorate the solution quality. Then, the *exchange natural sequences* operator (see Chapter 4) performs the best swap of two natural sequences, being node sequences before and after which the vehicle is empty. They can be exchanged without violating the pairing and precedence constraints. Any two natural sequences

in the solution may be used, provided that they belong to different routes and at most one of them is an empty sequence. If no improving exchange can be found, the original solution is maintained.

6.2.4 Periodic diversification

After f_{div} iterations without improvement of s_{best}, a periodic diversification is applied in order to guide the search towards another part of the solution space. This periodic diversification phase affects s_{cur} and s_{best}, but not s_{ovr}, unless the diversification would directly result in a new overall best solution. The random removal and random order reinsertion operators presented in Sections 6.2.1 and 6.2.2 destroy and repair des_{max} percent of all requests in a copy of s_{ovr}. No threshold applies to the newfound objective value, such that a complete solution is always accepted, provided that all feasibility constraints are respected. If not, the entire diversification phase is reapplied until a feasible solution is found. It is stored into s_{cur} and s_{best}, after which the normal LNS procedure resumes from these solutions.

6.3 Analysis of joint route planning

The LNS algorithm has been implemented in Python 3.4.3. Note that Cython 0.22, a module that runs Python code in C and allows variables to be declared as C types, is invoked to accelerate the numerous executions of feasibility checks. The computational resources and services were provided by the VSC (Vlaams Supercomputer Centrum), funded by the Research Foundation Flanders (FWO) and the Flemish Government, department EWI. All experiments are performed using Xeon E5-2680v2 CPUs at 2.8 GHz with 64 GB of RAM.

Section 6.3.1 uses existing artificial data to obtain a first impression of the performance of the solution method and the operational savings that may be obtained thanks to joint route planning. However, this data set is rather small and ignores real-life characteristics that may influence the findings. Therefore, Section 6.3.2 introduces a new artificial data set to analyze the effect of different operational settings on the benefits of joint route planning. Finally, Section 6.3.3 conducts a real-life case study to observe a pattern in the exchanges of requests among providers. This may allow them to predict which requests contribute in the cooperation.

6.3.1 Existing artificial data

A first series of experiments is performed on the relatively small artificial benchmark instances of Røpke et al. (2007), discussed in Chapter 2. Their original data cover the widely studied single-depot case, with all vehicles stationed in a single depot that is located in the center of the square area at

coordinates (0, 0). Braekers et al. (2014) extended this data set to a multi-depot variant, creating four depots located at coordinates (5, 5), (-5, -5), (5, -5) and (-5, 5). This variant will be used to represent the presence of multiple service providers. If they cooperate, all requests may be handled by vehicles originating from any depot. Otherwise, each request is assigned to one specific depot, which is the depot located closest to their inbound (home) location. Note that this chapter assumes that the different providers implement a comparable quality policy, such that the average quality experience of users is equal among all providers.

6.3.1.1 Parameter tuning

First, these artificial instances are used for tuning the three main parameters of the metaheuristic. These are (1) the maximum destroy percentage des_{max}, (2) the maximum deterioration factor det_{max} and (3) the diversification frequency f_{div}. Prior to performing detailed parameter tuning tests, a range of appropriate values is identified for each parameter using the automatic iterated racing algorithm of López-Ibáñez et al. (2016)[2]. The algorithm is initiated with a very broad range of possible settings for each of the three parameters: (1) [0.20, 0.50], (2) [1.01, 1.08] and (3) [100, 5000]. In each race, a sample of candidate configurations is drawn and repeatedly tested on the single-depot and multi-depot variants of the instances a8-64, a8-80 and a8-96[3]. Candidate configurations are discarded as soon as they perform statistically worse than others. The next race starts with a new sample of candidate configurations that is drawn such that promising parameter settings are favored. In this test, a large tuning budget of 1000 experiments is provided, which implies that the algorithm can perform three subsequent races of approximately 333 experiments. The number of candidate configurations decreases in subsequent races, such that more evaluations per configuration can be executed. Table 6.1 summarizes the results of the automatic tuning. After three races, the algorithm identifies three elite configurations, but it is not able to discard any of the 34 other configurations investigated during the last race. This implies that the average results found for all 37 candidate configurations in the last race are extremely close. For example, the differences in solution quality found for the three elite candidates do not exceed 0.04%. Therefore, a more in-depth tuning is required within the promising ranges identified by the tuning algorithm. Based on the surviving candidates throughout the entire procedure, three well-performing settings were identified for all three parameters: (1) 0.30, 0.35 and 0.40, (2) 1.04, 1.05, 1.06 and (3) 500, 1000 and 1500.

For each of the 27 resulting combinations, the algorithm is now run 20 times (20000 iterations per run) on the selected instances. Table 6.2 summarizes the average objective value and average

[2]The experiments in previous chapters were performed before this tool became commonly established.
[3]Tests on smaller instances do not result in any optimality gaps at all.

Race	Number surviving	Elite candidates		
1	24/47	0.44	1.05	925
		0.20	1.06	581
		0.26	1.07	152
2	11/41	0.41	1.04	1417
		0.40	1.05	1482
		0.37	1.04	480
3	37/37	**0.40**	**1.04**	**1468**
		0.39	1.05	1456
		0.37	1.04	480

Table 6.1: Parameter tuning on selected artificial instances, using the Irace package of López-Ibáñez et al. (2016).

	Single-depot		Multi-depot		Average	
	dist	time (m)	dist	time (m)	dist	time (m)
Maximum destroy percentage des_{max}						
30%	975.99	13.60	933.08	12.73	954.53	13.16
35%	**975.93**	14.85	**932.81**	14.00	**954.37**	14.42
40%	975.95	16.05	932.84	15.20	954.39	15.62
Maximum deterioration factor det_{max}						
1.04	975.97	13.96	932.90	13.05	954.44	13.51
1.05	**975.88**	14.89	**932.88**	14.03	**954.38**	14.46
1.06	976.01	15.64	932.94	14.84	954.47	15.24
Diversification frequency f_{div}						
500	**975.75**	15.52	**932.69**	14.55	**954.22**	15.03
1000	975.96	14.75	932.96	13.91	954.46	14.33
1500	976.15	14.22	933.07	13.47	954.61	13.85

Table 6.2: Manual parameter tuning on selected artificial instances.

computation time (in minutes) for each parameter separately. This analysis confirms that the performance of the LNS algorithm remains noticeably stable and reveals $des_{max} = 0.35$, $det_{max} = 1.05$ and $f_{div} = 500$ as the preferred parameter setting. This combination also generates the best average result of all 27 combinations, with objective values 975.22 and 932.26. The fact that the resulting parameter values are identical for the single-depot and multi-depot instances illustrates the robustness of the parameter tuning. As expected, the solution quality is not very sensitive to small changes in des_{max} and det_{max}, as long as these parameters remain within acceptable ranges. That is, large enough to escape the current solution, but also small enough to preserve an efficient route structure. However, rather frequent diversifications through a small value of f_{div} are highly beneficial, which was not apparent from the automatic tuning. All three parameters clearly influence the computation time. In other words, small sacrifices of solution quality may reduce the computation time if needed.

Also note that the automatic tuning does not consider the spread of the solutions for a particular parameter setting. Figure 6.1 reveals a more detailed analysis in which information on the average solution quality for each parameter separately (based on Table 6.2) is supplemented with the worst and the best solution found. This illustrates that applying a larger destroy percentage and a larger maximum deterioration may be useful to avoid extreme solutions. The worst solution is - contrary to the average solution - considerably better than for the preferred setting, whereas the best-found solution tends to remain stable. This is also confirmed by an analysis of the corresponding standard deviations, which is not shown here.

Even though all experiments performed on the artificial instances are based on 20000 iterations, it is interesting to point out that the LNS metaheuristic tends to converge towards a good solution range rather quickly. Table 6.3 shows the evolution of the average solution quality and computation time for the single-depot and multi-depot variants of instances a8-64, a8-80 and a8-96 after every 5000 iterations. Note that the solutions for multi-depot instances tend to converge somewhat more slowly, which confirms the observation of Braekers et al. (2014) that these instances are more difficult to solve. In other words, the larger search space offers more degrees of freedom. Nevertheless, after only 5000 iterations, the average objective value already approximates its eventual value. Beyond 20000 iterations, the limited improvements in solution quality do no longer justify the linear increase in computation time.

6.3.1.2 Design choices

As discussed in Section 6.2.1, the set of destroy operators used slightly deviates from the common practice in the literature. For example, no route removal is included and the additional proximity

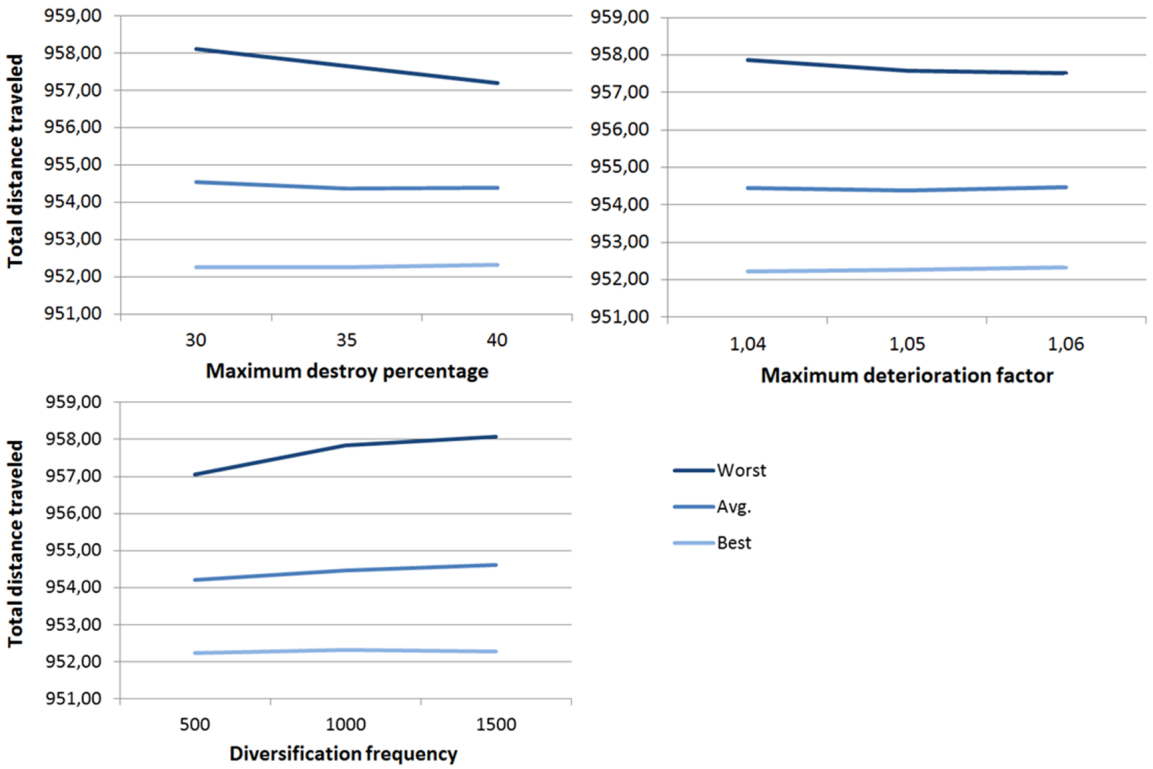

Figure 6.1: Influence of parameter settings on the best, average and worst solutions for selected artificial instances.

Iter.	**Single-depot**		**Multi-depot**	
	dist	time (m)	dist	time (m)
5000	977.52	3.95	935.09	3.71
10000	976.44	7.87	933.28	7.39
15000	975.67	11.75	932.61	11.01
20000	**975.22**	**15.66**	**932.26**	**14.65**
25000	975.17	19.61	931.99	18.29
30000	975.09	23.50	931.87	21.91
35000	975.03	27.39	931.71	25.55
40000	974.95	31.29	931.68	29.21

Table 6.3: Evolution of solution quality and computation times throughout the iterations for selected artificial instances.

removal operator is proposed. This choice was based on preliminary tests of different designs. The upper part of Table 6.4 summarizes the effect of adding a route removal operator or deleting one of the removal operators currently included. For every design, 20 runs consisting of 20000 iterations were performed on the single-depot and multi-depot variants of instances a8-64, a8-80 and a8-96. Adding a route removal operator, which continues removing all requests in random routes until the destroy percentage is reached, causes the largest deterioration of the objective value. This operator tends to destroy the solution structure to such an extent that the repaired solution often violates the maximum deterioration threshold. On the other hand, it reduces the total computation time, which indicates that this operator requires little computational effort. Deleting the related removal causes the second largest deterioration of the objective function. As explained in Section 6.2.1, it is beneficial to remove requests that are easily interchangeable in order to obtain a different repaired solution. The related removal is computationally rather inexpensive thanks to the fact that the similarities between requests can be computed in advance. The other operators - including the newly proposed proximity removal - also have a positive, though more limited effect on the solution quality.

Generally, the effects of deleting any operator remain very limited thanks to the additional local search phase, which contributes considerably to the performance of the algorithm. As illustrated in the lower part of Table 6.4, leaving out these local search operators causes a considerable deterioration of the solution quality, even when the corresponding savings in computation time are spent on additional LNS iterations. In other words, the advantages caused by including the additional local search phase justify its computational requirements amounting up to half of the total computation time.

	dist	time (m)
Current design	**953.74**	**15.15**
... with route removal	954.78	14.47
... without random removal	953.87	15.41
... without worst removal	953.83	15.18
... without related removal	954.32	15.36
... without proximity removal	953.89	14.74
... without additional LS (20000 LNS iter.)	968.40	7.59
... without additional LS (40000 LNS iter.)	965.30	15.15

Table 6.4: Average solution quality and computation times for different designs of the removal phase for selected artificial instances.

6.3.1.3 Computational tests

Tables 6.5 and 6.6 present detailed average, best and worst results obtained by the LNS metaheuristic for all single-depot instances of Røpke et al. (2007) and multi-depot instances of Braekers et al. (2014) in 20 runs, which consist of 20000 iterations each. For the single-depot instances, the solution quality comes close to the results of the hybrid genetic algorithm of Masmoudi et al. (2017). They published the best-known heuristic solutions on these instances, performing 5 runs consisting of 50000 iterations. The proposed LNS metaheuristic leaves no optimality gap for the smallest half of the instances, whereas the average gap for the largest half of the instances equals a mere 0.05% (varying between 0.00% and 0.26%). The best solutions match the optimal results in Gschwind and Irnich (2014) for all instances. For the multi-depot instances, a comparison is made with the results of Braekers et al. (2014), performing 5 runs consisting of 350,000 deterministic annealing iterations. The LNS metaheuristic improves 5 of their average solutions, although its overall performance is slightly worse. The average optimality gap equals 0.00% for the smallest half of the instances and 0.19% for the largest half (varying between 0.00% and 0.45%). The best solutions match the optimal results in Braekers et al. (2014) for all but two instances. The optimal results for instances a8-80 and a8-96 are still unknown, but new best heuristic results are found. The corresponding gap for these instances is expressed relative to the best-known upper bound. Finally, the worst results over 20 runs of the LNS metaheuristic are mentioned for the information of the reader. They show that it performs very consistently in the sense that it never produces undesirably large optimality gaps. As other authors did not report their worst results, no comparison can be made in this respect.

The total distance traveled is consistently lower in the presence of multiple depots than in the single-depot case, although the number of requests and the total size of the fleet remains unchanged. This indicates that from a cost perspective, service providers may take advantage of having multiple depots spread across their service area. These operational savings should of course be weighed against the fixed costs of exploiting multiple depots and a potential loss of operational flexibility. For example, a central depot often accommodates a backup driver to deal with unexpected sickness of a driver, whereas it may be impracticable to provide such a backup in each of the multiple depots.

The computation times exceed the ones reported by both Masmoudi et al. (2017) and Braekers et al. (2014), even though a relatively small number of iterations is performed. In this respect, it should be noted that these authors implemented their algorithms using the C++ programming language, rather than Python. The Python language allows for a rapid development of object-oriented programs, especially suitable to examine algorithm potential. Unfortunately, these programs tend to execute relatively slowly. A fair comparison between the proposed LNS and other metaheuristics

	LNS algorithm							*Masmoudi et al. (2017)*				
Inst.	**Avg.**	**Gap**	**T(m)**	**Best**	**Gap**	**Worst**	**Gap**	**Avg.**	**Gap**	**T(m)**	**Best**	**Gap**
a2-16	294.25*	0.00%	0.57	294.25*	0.00%	294.25*	0.00%	294.25*	0.00%	0.21	294.25*	0.00%
a2-20	344.83*	0.00%	0.87	344.83*	0.00%	344.83*	0.00%	344.83*	0.00%	0.48	344.83*	0.00%
a2-24	431.12*	0.00%	1.14	431.12*	0.00%	431.12*	0.00%	431.12*	0.00%	0.42	431.12*	0.00%
a3-18	300.48*	0.00%	0.98	300.48*	0.00%	300.48*	0.00%	300.48*	0.00%	0.22	300.48*	0.00%
a3-24	344.83*	0.00%	1.32	344.83*	0.00%	344.83*	0.00%	344.83*	0.00%	0.40	344.83*	0.00%
a3-30	494.85*	0.00%	1.94	494.85*	0.00%	494.85*	0.00%	494.85*	0.00%	0.45	494.85*	0.00%
a3-36	583.19*	0.00%	2.54	583.19*	0.00%	583.19*	0.00%	583.19*	0.00%	0.67	583.19*	0.00%
a4-16	282.68*	0.00%	0.97	282.68*	0.00%	282.68*	0.00%	282.68*	0.00%	0.23	282.68*	0.00%
a4-24	375.02*	0.00%	1.49	375.02*	0.00%	375.02*	0.00%	375.02*	0.00%	0.31	375.02*	0.00%
a4-32	485.50*	0.00%	2.59	485.50*	0.00%	485.50*	0.00%	485.50*	0.00%	0.61	485.50*	0.00%
a4-40	557.69*	0.00%	3.10	557.69*	0.00%	557.69*	0.00%	557.69*	0.00%	0.63	557.69*	0.00%
a4-48	668.82*	0.00%	5.53	668.82*	0.00%	668.82*	0.00%	668.82*	0.00%	0.85	668.82*	0.00%
Avg.	**430.27**	**0.00%**	**1.92**	**430.27**	**0.00%**	**430.27**	**0.00%**	**430.27**	**0.00%**	**0.46**	**430.27**	**0.00%**
a5-40	498.41*	0.00%	3.89	498.41*	0.00%	498.41*	0.00%	498.41*	0.00%	0.52	498.41*	0.00%
a5-50	686.62*	0.00%	6.35	686.62*	0.00%	686.62*	0.00%	686.62*	0.00%	0.73	686.62*	0.00%
a5-60	808.42*	0.00%	8.16	808.42*	0.00%	808.42*	0.00%	808.42*	0.00%	1.05	808.42*	0.00%
a6-48	604.12*	0.00%	6.06	604.12*	0.00%	604.12*	0.00%	604.12*	0.00%	0.73	604.12*	0.00%
a6-60	819.30	0.01%	8.74	819.25*	0.00%	820.22	0.12%	819.25*	0.00%	0.88	819.25*	0.00%
a6-72	916.46	0.05%	11.71	916.05*	0.00%	918.08	0.22%	916.05*	0.00%	1.28	916.05*	0.00%
a7-56	724.04*	0.00%	8.00	724.04*	0.00%	724.04*	0.00%	724.04*	0.00%	0.77	724.04*	0.00%
a7-70	889.44	0.04%	11.18	889.12*	0.00%	892.32	0.36%	889.12*	0.00%	1.04	889.12*	0.00%
a7-84	1036.08	0.26%	15.41	1033.37*	0.00%	1038.00	0.45%	1033.37*	0.00%	1.39	1033.37*	0.00%
a8-64	747.58	0.02%	10.85	747.46*	0.00%	749.29	0.24%	747.46*	0.00%	0.90	747.46*	0.00%
a8-80	946.37	0.07%	16.09	945.73*	0.00%	949.19	0.37%	945.97	0.02%	1.38	945.73*	0.00%
a8-96	1231.72	0.17%	20.02	1229.66*	0.00%	1234.68	0.41%	1231.04	0.11%	1.59	1229.66*	0.00%
Avg.	**825.71**	**0.05%**	**10.54**	**825.19**	**0.00%**	**826.95**	**0.18%**	**825.32**	**0.01%**	**1.02**	**825.19**	**0.00%**

$*$ = *optimal solution*

Table 6.5: Complete LNS results on the single-depot artificial data of Røpke et al. (2007) after 20000 iterations.

	LNS algorithm							*Braekers et al. (2014)*				
Inst.	**Avg.**	**Gap**	**T(m)**	**Best**	**Gap**	**Worst**	**Gap**	**Avg.**	**Gap**	**T(m)**	**Best**	**Gap**
a2-16	284.18*	0.00%	0.57	284.18*	0.00%	284.18*	0.00%	284.18*	0.00%	0.09	284.18*	0.00%
a2-20	343.43*	0.00%	0.86	343.43*	0.00%	343.43*	0.00%	343.43*	0.00%	0.27	343.43*	0.00%
a2-24	427.17*	0.00%	1.13	427.17*	0.00%	427.17*	0.00%	427.17*	0.00%	0.26	427.17*	0.00%
a3-18	289.67*	0.00%	0.83	289.67*	0.00%	289.67*	0.00%	289.67*	0.00%	0.13	289.67*	0.00%
a3-24	348.30*	0.00%	1.38	348.30*	0.00%	348.30*	0.00%	348.30*	0.00%	0.24	348.30*	0.00%
a3-30	469.16*	0.00%	1.74	469.16*	0.00%	469.16*	0.00%	469.16*	0.00%	0.24	469.16*	0.00%
a3-36	592.42*	0.00%	2.57	592.42*	0.00%	592.42*	0.00%	592.42*	0.00%	0.41	592.42*	0.00%
a4-16	262.44*	0.00%	0.85	262.44*	0.00%	262.44*	0.00%	262.44*	0.00%	0.12	262.44*	0.00%
a4-24	355.72*	0.00%	1.49	355.72*	0.00%	355.72*	0.00%	355.72*	0.00%	0.18	355.72*	0.00%
a4-32	461.65*	0.00%	2.48	461.65*	0.00%	461.65*	0.00%	461.65*	0.00%	0.30	461.65*	0.00%
a4-40	540.42	0.01%	3.20	540.34*	0.00%	540.72	0.07%	540.34*	0.00%	0.33	540.34*	0.00%
a4-48	631.75*	0.00%	5.20	631.75*	0.00%	631.75*	0.00%	632.31	0.09%	0.43	631.75*	0.00%
Avg.	**417.19**	**0.00%**	**1.85**	**417.19**	**0.00%**	**417.22**	**0.01%**	**417.23**	**0.01%**	**0.25**	**417.19**	**0.00%**
a5-40	482.20	0.00%	4.12	482.19*	0.00%	482.30	0.02%	482.19*	0.00%	0.33	482.19*	0.00%
a5-50	665.08	0.08%	6.02	664.54*	0.00%	665.17	0.09%	665.17	0.09%	0.45	664.54*	0.00%
a5-60	790.13	0.03%	7.45	789.87*	0.00%	793.32	0.44%	789.87*	0.00%	0.58	789.87*	0.00%
a6-48	586.85	0.13%	5.70	586.08*	0.00%	589.91	0.65%	586.08*	0.00%	0.40	586.08*	0.00%
a6-60	777.99	0.17%	8.07	776.63*	0.00%	780.94	0.55%	776.65	0.00%	0.57	776.63*	0.00%
a6-72	884.38	0.07%	11.29	883.77*	0.00%	886.23	0.28%	883.77*	0.00%	0.83	883.77*	0.00%
a7-56	680.44	0.05%	7.08	680.08*	0.00%	681.86	0.26%	682.14	0.30%	0.47	680.08*	0.00%
a7-70	857.09	0.34%	11.20	854.22*	0.00%	860.18	0.70%	857.67	0.40%	0.70	855.76	0.18%
a7-84	1011.82	0.45%	14.82	1007.33*	0.00%	1014.68	0.73%	1009.92	0.26%	0.86	1007.33*	0.00%
a8-64	715.04	0.36%	9.83	713.11*	0.00%	717.50	0.62%	713.11*	0.00%	0.73	713.11*	0.00%
a8-80	891.82	0.28%	14.84	889.29	0.00%	894.15	0.55%	892.79	0.39%	1.19	890.69	0.16%
a8-96	1189.91	0.38%	19.28	1186.63	0.10%	1195.24	0.83%	1189.74	0.36%	1.57	1187.26	0.15%
Avg.	**794.39**	**0.19%**	**9.98**	**792.81**	**0.01%**	**796.79**	**0.48%**	**794.09**	**0.15%**	**0.72**	**793.11**	**0.04%**

$*$ = *optimal solution*

Table 6.6: Complete LNS results on the multi-depot artificial data of Braekers et al. (2014) after 20000 iterations.

can only be based on results obtained upon meeting a criterion that is independent of the processor speed or the coding efficiency (Leyman and De Causmaecker, 2017), such as the number of routes investigated during the procedure (i.c. the number of times a feasibility check has been performed). Anyway, this contribution has no intention to be competitive in terms of computation time, particularly since the analysis of centralized decision making through joint route planning is situated at a strategic level, rather than at a time-critical operational level. The aforementioned results indicate that in terms of solution quality, the LNS metaheuristic can be competitive with the current state of the art within reasonable computation times. Hence, it is an appropriate instrument for assessing the potential of joint route planning in dial-a-ride services.

6.3.1.4 Joint route planning

The presence of multiple depots in the data of Braekers et al. (2014) allows to analyze the impact of joint route planning for these artificial instances. In fact, the solution strategy adopted thus far assumes a cooperation between four service providers, i.c. the four depots. Every request may be handled by a vehicle originating from any depot, which can be seen as service providers that allow the exchange of any request among each other. A scenario without joint route planning is created by assigning each request to one specific service provider. To this end, it is assumed that a request can only be handled by a vehicle that originates from the depot closest to the user's home location. This represents the provider that the user would intuitively choose, assuming that all providers offer an equal quality of service on average. Each depot disposes of one fourth of the total fleet. If the number of vehicles is not sufficient to serve all users, an additional vehicle is added. The results are now obtained by solving the problem for each depot separately, given the assignment of requests to depots. The resulting difference in solution quality and fleet requirements is presented in Table 6.7. The second column indicates the relative saving in the joint distance traveled obtained through joint route planning. The next four columns present detailed results for the scenarios with and without cooperation. Although no exact solutions are known for the scenario without cooperation, the LNS metaheuristic is likely to generate optimal solutions for such small-sized problems. The results show that for this data set, the decrease in joint distance traveled through joint route planning ranges between 10.28% and 29.57%. Fleet requirements are smaller as well, although fleet minimization is not an objective as such. Particularly small-sized service providers considerably reduce their fleet requirements by exchanging requests that are inconvenient to include in their own routes.

6.3.2 New artificial data

Real-life dial-a-ride systems are most often characterized by additional features not included in the data used in Section 6.3.1. First, requests are usually submitted in pairs. A user traveling from

Inst.	**Benefits**	**Cooperation**		**No cooperation**	
	dist	dist	fleet	dist	fleet
a2-16	21.62%	284.18	2	362.56	5
a2-20	10.28%	343.43	2	382.79	4
a2-24	12.75%	427.17	2	489.57	5
a3-18	18.78%	289.67	3	356.65	4
a3-24	24.43%	348.30	3	460.87	4
a3-30	19.75%	469.16	3	584.59	5
a3-36	15.60%	592.42	3	701.95	7
a4-16	18.61%	262.44	4	322.43	5
a4-24	20.34%	355.72	4	446.56	7
a4-32	23.18%	461.65	4	600.96	7
a4-40	24.69%	540.34	4	717.47	8
a4-48	18.74%	631.75	4	777.48	8
Avg.	**19.30%**	**417.19**	**3.17**	**516.99**	**5.75**
a5-40	26.34%	482.19	5	654.59	6
a5-50	29.57%	664.54	5	943.59	6
a5-60	22.58%	789.87	5	1020.23	7
a6-48	22.83%	586.08	6	759.46	8
a6-60	23.20%	776.63	6	1011.22	8
a6-72	25.70%	883.77	6	1189.43	9
a7-56	27.00%	680.08	7	931.57	9
a7-70	23.18%	854.22	7	1111.93	8
a7-84	26.41%	1007.33	7	1368.92	8
a8-64	28.91%	713.11	8	1003.07	9
a8-80	25.40%	889.29	8	1192.13	10
a8-96	24.49%	1185.45	8	1569.84	10
Avg.	**25.43%**	**792.71**	**6.50**	**1063.00**	**8.17**

Table 6.7: Effect of cooperation through joint route planning on the multi-depot instances of Braekers et al. (2014).

his home location to a certain event (e.g. a medical appointment) is likely to make the reverse trip later that day. Even though the existing artificial instances consist of an equal number of outbound and inbound requests, their geographical and time-related pairing is completely ignored. Second, requests tend to be clustered in space. Most origins and destinations of users are located in densely populated areas or at specific attraction poles, such as hospitals. The existing artificial instances do not take into account this observation either. Finally, most real-life systems serve a larger number of users. Particularly in the multi-depot case, even the largest instance reaches an average of merely 24 requests per depot (= 96 requests over four depots). Therefore, a more extensive artificial data set with additional real-life characteristics is introduced in Section 6.3.2.1. Section 6.3.2.2 presents six hypotheses regarding the effect of the operational setting on the benefits of joint route planning, which are then tested in Section 6.3.2.3 using the new artificial instances.

6.3.2.1 Data set

To analyze the effect of joint route planning in the presence of somewhat more realistic features, 10 new artificial multi-depot instances have been created. All instances include a total number of 400 requests, originating from 200 users submitting corresponding outbound and inbound requests. Random time windows of 15 minutes are imposed for the arrival of outbound trips and the departure of inbound trips. The time between the latest arrival for an outbound trip and the earliest departure for the corresponding inbound trip ranges between one and four hours. The service area $[-10, 10]^2$ is assumed to consist of four densely populated cities or clustering poles. Each clustering pole has a vehicle depot (i.c. four service providers) possessing 8 vehicles, which results in a total number of 32 vehicles. The depots are located at coordinates (-5, -5), (5, -5), (-5, 5) and (5, 5) and an equal number of 50 users (100 requests) is assigned to each depot. This is reflected by the fact that their home location, being the origin of their outbound trip and the destination of their inbound trip, is clustered around a particular depot using the procedure of Cordeau et al. (1997), with clustering parameter $\phi = 0.75$. Based on real-life experiences, it is assumed that 25% of the users request a long-distance trip, in which case the destination of their outbound trip (the origin of their inbound trip) is clustered around a different depot than their home location. For other users, both locations are clustered around the same depot. The maximum user ride time and maximum route duration equal 30 and 720 minutes, respectively. The service at each stop takes three minutes and the vehicle capacity is limited to three users.

6.3.2.2 Effect of operational setting on cooperation benefits

The impact of the operational setting on the benefits of joint route planning can be analyzed by applying minor changes to the instances introduced in the previous section. Six groups of alternative

operational settings are considered based on the following hypotheses:

1. The relative benefits of joint route planning increase as the *width of the users' time windows decreases*. This hypothesis is based on the fact that a reduction of feasible user combinations decreases the operational flexibility of providers (see Chapter 3), such that the effect of joint route planning may become more important. Reductions of the time window width from 15 minutes to (a) 10, (b) 5 and (c) 0 minutes are investigated.

2. The relative benefits of joint route planning increase as *more users request long-distance trips*. This hypothesis assumes that a frequent presence of service providers in each other's service areas enables more request exchanges and increases the potential of joint route planning. The percentage of long-distance requests is increased from 25% to (a) 50%, (b) 75% and (c) 100%.

3. The relative benefits of joint route planning increase as the *strength of the clustering around the clustering poles weakens*. This hypothesis assumes that overlapping service areas of different providers enable additional potential request exchanges and increase the benefits of joint route planning. The clustering parameter ϕ is reduced from 0.75 to (a) 0.60, (b) 0.45 and (c) 0.30[4].

4. The relative benefits of joint route planning increase as the *clustering poles are located closer to each other*. This hypothesis may again be explained by an increased overlap of the providers' service areas. The coordinates of the depots are adapted from -5.00/5.00 to (a) -3.33/3.33, (b) -1.66/1.66 and (c) a single central depot at (0, 0).

5. The relative benefits of joint route planning increase as the *size of the problem reduces*. This hypothesis assumes that small-sized service providers have few possibilities to efficiently combine their own users, such that they gain considerable operational flexibility by means of joint route planning. The instance sizes are reduced from 400 requests and 32 vehicles to (a) 300 requests and 24 vehicles, (b) 200 requests and 16 vehicles and (c) 100 requests and 8 vehicles.

6. The relative benefits of joint route planning either increase or decrease as *the providers' sizes are less balanced*. It is hard to predict whether the relatively large benefits obtained by small providers or the relatively small benefits obtained by large providers will prevail. Starting from equal providers handling 100 requests each, the balance is changed to (a) 125-75-75-125, (b) 150-50-50-150 and (c) 175-25-25-175. The large-sized providers exploit depots (5, 5) and (-5, -5) and the fleet is allocated proportionally: (a) 10-6-6-10, (b) 12-4-4-12 and (c) 14-2-2-14.

[4]Scenarios with lower values of ϕ have not been considered, since they are not realistic. In the most extreme scenario ($\phi = 0.00$), user locations would be spread in a completely random manner, not influenced by the distance to any depot. As a result, a user living near a particular service provider would have a 75% probability of contacting one of the three other providers, which does not correspond to realistic behavior.

Size	dist	time
4 routes	2191.60	15.72
6 routes	2154.08	26.69
8 routes	**2130.24**	**39.95**
10 routes	2114.63	57.10
12 routes	2107.81	75.77

Table 6.8: Average results and computation times for different sizes of the selected subset of routes in each iteration.

To investigate these hypotheses, adaptations are applied to the current operational setting for each hypothesis separately. The relative benefits of joint route planning are computed and compared to the baseline scenario. The results are obtained after 2 runs for each instance (20 runs in total), with each run consisting of 20000 iterations. In order to deal with the large scale of the instances, the operations in each iteration of the LNS algorithm are performed on a selection of eight related routes, rather than on the complete set of routes (e.g. similar to the principle of parallel computing (Taillard, 1993)). For large-scale instances with many vehicles, requests that are distant from each other are unlikely to be successfully interchangeable. The selection of related routes prior to each iteration requires the following steps. First, a random route is selected based on equal probabilities. Second, the relatedness between this route and all other routes is assessed. The distances between each node in the first route and its closest node in the second route (which may be the depot) are summed, resulting in a measure that is small for related routes. Third, the seven routes being most related to the first route are selected, provided that at least two routes in the total selection are not empty. This selection procedure is repeated prior to each iteration. The choice of the subset's size is based on computational tests, of which the results are shown in Table 6.8. This table shows the average total distance for different sizes of the subset, as well as the corresponding computation time in minutes. The solution quality improves at a declining rate as the size of the subset increases, whereas the computation times increase at a growing rate. Subsets consisting of 8 routes provide a reasonable equilibrium between solution quality and computation time.

6.3.2.3 Results

Table 6.9 presents the average results in the baseline scenario and the six alternative operational settings. The first column refers to the scenario under consideration. The second column gives the relative saving in joint distance traveled caused by joint route planning. Detailed information on the distance traveled and the required fleet with and without cooperation is displayed in the next four

columns. The seventh column shows whether the relative benefits of joint route planning in a given scenario are statistically different from the 'previous' scenario. For example, scenario *50% long trips* is compared with the baseline scenario, scenario *75% long trips* is compared with scenario *50% long trips*, etc. The corresponding p-values are indicated. For hypotheses (1) and (5), the assessment is based on the *Wilcoxon signed-ranks test*[5], since the locations of the user requests remain unchanged from the baseline scenario. The weaker, but less restricted *Mann-Whitney U-test* is used for the other hypotheses[6] (Anderson et al., 2010). Finally, the last column in Table 6.9 mentions the percentage of requests that are exchanged to another depot in the scenario with cooperation.

Table 6.9 shows that the benefits of joint route planning, ranging between 5.61% and 16.73%, depend on the operational setting. In the given experimental setting, hypotheses (1), (2), (3), (4) and (5) are essentially confirmed by the results. For example, reducing the width of the time windows in hypothesis (1) increases the total distance traveled, due to the fact that service providers have fewer possibilities to combine users. However, this impact is less noticeable when service providers take advantage of the operational flexibility created by joint route planning, which explains why the benefits of cooperating become more important as time windows become tighter. Note that for hypothesis (2), the benefits of joint route planning do no longer increase significantly once a level of 75% long trips is reached. For hypothesis (3), the benefits of joint route planning only increase significantly once the clustering becomes weak ($\phi = 0.45$ or lower). Hypothesis (6) is not confirmed, as there is no clear pattern across the scenarios and the differences are very small. Consequently, in the given experimental setting, only the global size of the cooperation is determinative of the operational benefits, not the size differences between the participating service providers. However, the reader should be aware that this experimental design does not consider any interaction effects. For example, hypothesis (6) may be confirmed for a smaller number of requests in the data set.

Hypotheses (2), (3) and (4) assumed that the benefits of joint route planning are affected by the potential of request exchanges, e.g. because of overlapping service areas. This is confirmed by the

[5]The *Wilcoxon signed-ranks* test is a non-parametric variant of the *paired t-test*. The assumption of paired data stems from the fact that the location of the user requests remains unchanged from the baseline scenario. The choice of a non-parametric variant is explained by the fact that the population of paired differences does not approximate a normal distribution. The test computes the difference in the relative advantage of joint route planning for each pair of instances. These differences are ranked and each rank receives the sign of the original difference. Assuming that the sum of the ranks follows a symmetric distribution around 0, it is assessed whether the encountered deviation is plausible.

[6]The *Mann-Whitney U-test* does not require paired data, which is appropriate given that the physical location of the users is different from the baseline scenario. The test makes a joint ranking of all relative advantages of joint route planning. Assuming that the population from which an observation originates does not influence its position in the ranking, it is assessed whether the encountered ranking is plausible.

Scenario	**Benefits**	**Cooperation**		**No cooperation**		**Sign.**	**Exch.**
	dist	dist	fleet	dist	fleet	p-val	pct.
Baseline	5.61%	2130.24	30.20	2256.96	32.00	-	43.34%
Hypothesis 1							
- time window width 10	6.70%	2195.39	31.45	2353.15	32.00	0.005*	41.61%
- time window width 5	7.85%	2302.46	31.95	2498.56	32.00	0.005*	42.19%
- time window width 0	10.33%	2489.76	32.00	2776.47	32.00	0.005*	42.63%
Hypothesis 2							
- 50% long trips	8.01%	2563.82	29.80	2786.95	32.00	0.000*	53.38%
- 75% long trips	10.62%	3001.90	29.80	3358.43	32.00	0.000*	57.69%
- 100% long trips	11.56%	3349.86	29.95	3787.91	32.00	0.063	57.44%
Hypothesis 3							
- clustering $\phi = 0.60$	6.27%	2337.91	29.95	2494.23	32.00	0.436	47.06%
- clustering $\phi = 0.45$	9.00%	2601.31	30.50	2858.70	32.00	0.001*	51.23%
- clustering $\phi = 0.30$	12.67%	2925.11	30.55	3349.34	32.00	0.001*	57.61%
Hypothesis 4							
- depots -3.33/3.33	7.17%	1856.75	29.15	2000.23	31.95	0.003*	48.75%
- depots -1.66/1.66	11.53%	1559.06	26.40	1762.28	31.80	0.000*	63.60%
- central depot	16.73%	1390.84	21.00	1670.31	31.35	0.000*	74.33%
Hypothesis 5							
- 300 requests, 24 vehicles	8.42%	1673.09	23.55	1826.84	24.00	0.005*	43.58%
- 200 requests, 16 vehicles	11.10%	1181.27	15.90	1328.83	16.00	0.007*	42.90%
- 100 requests, 8 vehicles	14.41%	669.27	8.00	781.92	8.00	0.005*	38.30%
Hypothesis 6							
- providers 125-75-75-125	5.83%	2164.61	29.45	2298.59	30.40	-	-
- providers 150-50-50-150	7.02%	2095.32	29.50	2253.55	28.15	-	-
- providers 175-25-25-175	6.29%	2074.89	28.35	2214.25	25.30	-	-

Table 6.9: Effect of different operational scenarios on benefits of cooperation through joint route planning.

percentage of request exchanges, particularly for hypothesis (4). Although this percentage is not an accurate indicator of the benefits of joint route planning (e.g. it remains unchanged for hypothesis (1)), this finding confirms the underlying mechanism assumed for hypothesis (2), (3) and (4).

As a final remark, note that the solution approach may slightly underestimate the operational savings generated by joint route planning. The average total distance traveled without cooperation, obtained by solving four separate small-sized routing problems, is presumably closer to optimality than a solution with joint route planning. From this perspective, the benefits of joint route planning may be slightly larger than computed. The optimality gaps cannot be displayed, since the optimal solutions for these large instances are unknown.

6.3.3 Real-life case study

The artificial data sets used in Sections 6.3.1 and 6.3.2 are suitable for comparing algorithmic performances or demonstrating the relative effects of different operational settings. However, they ignore real-life complexities which may influence the absolute size of the benefits obtained by joint route planning. First, real-life systems operate on a road network, such that the distances or travel times may not be directly proportional to Euclidean distances. Second, case-specific problem characteristics may be applicable, giving rise to additional constraints. Third, both from a cost-related and a quality-related perspective, service providers usually pursue more complicated objectives (measurable or not) than a minimization of the total distance traveled. The objective of this section is to quantify the benefits of joint route planning in the context of a real-life case study. In addition, the solutions with and without joint route planning are analyzed to identify specific conditions under which request exchanges among participating providers may be beneficial.

6.3.3.1 Data set

The data of this real-life case study is provided by a service provider in Belgium, having branches in Wilrijk and Melsbroek. Currently, these branches operate independently, meaning that there is no exchange of information or resources. Both branches dispose of a single depot which is strategically situated near a large rehabilitation center or hospital, notwithstanding the fact that users may request rides to any destination of choice. The current practice is that the users are assigned to the branch of which the depot is closest to their destination location, regardless of their home location. In Figure 6.2, the home locations of the users assigned to the branches in Wilrijk and Melsbroek (which are approximately 35 km apart) are represented by red dots and blue diamonds, respectively. It is visually clear that the service areas of both branches overlap to a certain extent, such that a cooperation between these branches may be worth investigating.

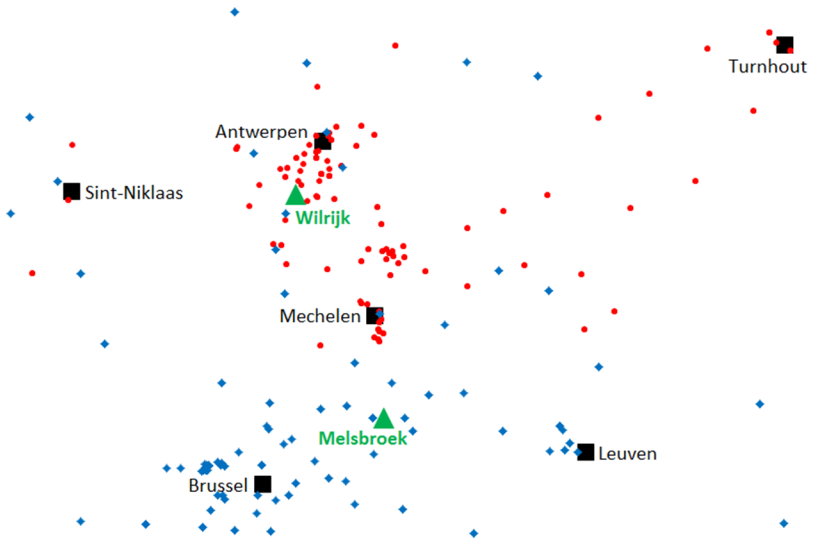

Figure 6.2: Home locations of the users in the real-life case study.

The data set covers two consecutive operating days, being Monday May 18 and Tuesday May 19, 2015. On the first day, the data consists of 296 private requests served in both branches. Without cooperation, 134 and 162 requests are assigned to Wilrijk and Melsbroek, respectively. On the second day, a total number of 241 (124 + 117) private requests should be satisfied. Note that a single request may involve multiple users traveling together, such that the total number of individual private requests equals 444 on Monday, which is typically a rather busy day, and 319 on Tuesday. The vast majority of the requests come in pairs. Users are heterogeneous with respect to their mobility, since approximately 70% of them are wheelchair users. Requests tend to be clustered around peak hours (8-10 am, 3-5 pm) and mainly represent trips to/from a limited set of rehabilitation centers or hospitals around Antwerp and Brussels.

The fleet consists of 69 vehicles, which offer a configurable ratio of normal seats and wheelchair spaces. In one of the extreme configurations, a maximum of four wheelchair users can be transported, leaving three normal seats available. The other extreme configuration provides a maximum of seven normal seats, leaving space for one wheelchair user. Several intermediate combinations are eligible as well. Routes investigated by the LNS algorithm satisfy the load constraints if at least one feasible configuration exists for each of its arcs separately. Currently, 23 and 46 vehicles are assigned to the depot in Wilrijk and Melsbroek, respectively. At first sight, the fleet size seems more than sufficient

to serve all requests. However, apart from the private user requests mentioned before, the provider also performs fixed pickup and delivery tours, typically at the request of daycare centers. They pick up users and deliver them at the daycare center in the morning, and bring them back home in the afternoon. For contractual reasons, private requests cannot be served in a fixed pickup and delivery tour. As a result, the vehicles performing such a tour are only available for private users in the time span between their arrival and their departure at the daycare center, which is usually the off-peak period between the morning and the afternoon peak. Therefore, the optimization problem addressed in this chapter, i.c. related to private requests, is quite tightly constrained during peak hours and rather easy during the off-peak. More specifically, a total number of 41 vehicles (12 in Wilrijk, 29 in Melsbroek) is not available during the peak hours. Finally, a vehicle is not necessarily physically stationed at the depot to which it is assigned. Drivers living relatively far from the depot are asked to take their vehicles home during the night if this allows for a cost-efficient service of users in that region during the early morning or late evening.

6.3.3.2 Problem characteristics

In consultation with the provider, the impact of three different objective functions is analyzed:

1. The first objective function corresponds to the typical objective used for artificial data, which is a minimization of the total traveling cost. This objective assumes that the costs incurred by a service provider are exclusively related to the total distance traveled by its vehicles, which is a substantial simplification of the service provider's real cost structure. Based on the service provider's information, a variable vehicle cost of €0.30 is incurred per kilometer traveled. Only the costs for serving private users are taken into account, since the fixed pickup and delivery tours are invariable.

2. The second objective function is based on a more comprehensive computation of costs, considering both the distance traveled and the wage payed to the driver. According to the service provider, the hourly driver wage amounts to €25 including taxes. A driver's wage cost is computed based on the time spent in the vehicle. This includes potential waiting time faced at any location, be it between two private requests, after a fixed pickup tour or before a fixed delivery tour. To avoid excessive driver waiting time during the off-peak, a shift can be split into two short shifts at most, provided that the intermediate break (starting and ending at the vehicle's depot) lasts for at least 3 hours. Since the driver may freely dispose of his time during such a break, the service provider does not incur wage costs meanwhile. This operational flexibility is useful to cope with demand differences during peak and off-peak hours.

3. In order to support tactical decisions, the fleet size may be included in the optimization process.

Even though the available number of vehicles is fixed in the short term, the service provider is particularly interested in knowing the minimum fleet size required to meet all user demand. Given certain fixed costs for acquiring, insuring and maintaining vehicles, this allows him to find a balance between a medium-term reduction in the fixed vehicle costs and the associated increase in operating costs due to a reduced flexibility in serving users. More specifically, the third objective function consists of a hierarchical minimization of (1) the number of activated vehicles and (2) the operating costs as defined in the second objective function. The first part only relates to those vehicles not performing a fixed pickup and delivery tour. When a vehicle is activated for a fixed pickup and delivery tour anyway, it would not make sense not to use it for private requests during the intermediate time span.

Furthermore, the heterogeneity of the users and the configurability of the fleet were already discussed in Section 6.3.3.1. Other problem characteristics largely correspond to the standard problem (Cordeau and Laporte, 2003). A time window is imposed for every request, though always on the time of pickup. The maximum user ride time is expressed as 1.5 times the direct ride time between the user's pickup and delivery location. A fixed service duration of five minutes is taken into account for wheelchair users only. The maximum route duration is limited to 10 hours, even for split shifts. An exception ensures that the same driver performs corresponding fixed pickup and delivery tours.

6.3.3.3 Results

Solving the real-life case requires the following conceptual adjustments. First, the *Graphhopper* API is invoked to geocode all nodes and precompute distances and travel times on the road network, using *OpenStreetMap* data. Based on realistic estimates, these travel times are multiplied by a factor 1.25 to provide sufficient margin in the schedules. Second, the scheduling procedure of Cordeau and Laporte (2003) is replaced by the one discussed in Chapter 5. The latter scheduling procedure delivers schedules in which the total user ride time is minimized, which is fully in line with the service policy of the provider. Last, additional depots disposing of a single vehicle are added to represent the addresses of the drivers who may take their vehicles home. An additional constraint prevents the total number of vehicles from exceeding the actual fleet size. Without cooperation, the total of all vehicles stationed at the depot in Wilrijk (resp. Melsbroek) or at the home address of a driver related to this depot cannot exceed 23 (resp. 46). With joint route planning, only the overall fleet size is limited to 69 vehicles. No constraint is imposed on the distribution of these vehicles among both depots.

Table 6.10 shows the average results with and without joint route planning for both operating days, optimizing the different objectives. A total number of 20 runs was performed, each consisting of 20000 iterations. This is a relatively high number of iterations, since the real-life case study turns

out to be more easily solvable than the artificial instances. Due to its large service area, many user combinations are anyhow impossible without violating the time-related constraints. The first column indicates the operating day and the applicable scenario (with/without joint route planning). The second column shows the total cost for a given scenario, which consists of the variable vehicle cost (€0.30 per km traveled) and the wage cost (€25 per hour, if applicable) displayed in the next two columns. The fifth column shows the total required fleet size, summing the active vehicles in both depots as mentioned in the next two columns. The eighth and ninth column indicate the number of requests served by both branches. The last two columns indicate the number of requests exchanged from Wilrijk to Melsbroek and from Melsbroek to Wilrijk, respectively.

Optimizing the first objective, the benefits of joint route planning are comparable to the baseline scenario for the artificial instances in Section 6.3.2. A 5.64% (May 18) and 3.88% (May 19) average reduction in the variable vehicle cost is found. This is a plausible outcome, given the rather limited degree of overlap between both service areas. In the scenario with joint route planning, only very small net transfers of vehicles[7] and users are suggested. However, the absolute number of request exchanges is very different on both operating days, even though the degree of overlap between the service areas is similar. This reconfirms the finding from Section 6.3.2.3 that the number of requests exchanged is not a reliable indicator of the benefits of joint route planning.

As mentioned before, a mere minimization of the total traveling cost does not fully comprise the cost-related approach envisaged by the service provider. It is therefore important to investigate how the benefits of joint route planning change when a more realistic objective is introduced. The results for the second objective show that, due to high labor costs in Belgium, the wage payed to drivers considerably outweighs the variable vehicle cost. The relative importance of the latter becomes so small that the efficiency of the vehicle routing (i.e. combining the users as efficiently as possible) is partly sacrificed whenever another routing plan shortens the overall work duration of the drivers. Joint route planning turns out to contribute relatively little in shortening the drivers' shifts. The average reduction in total cost is limited to 3.17% (May 18) and 1.69% (May 19) when the second objective is investigated. There is a net transfer of users towards Melsbroek, particularly on May 18. This is explained by the computation method of the wage cost. In Melsbroek, many drivers perform a fixed pickup and delivery tour at the start and end of their shift. Unproductive (but costly) waiting time after the end of a fixed pickup tour or before the start of a fixed delivery tour may be

[7] On May 19, the available fleet of 23 vehicles is insufficient to find a feasible solution for the branch of Wilrijk in the scenario without joint route planning, such that an additional vehicle was added during the solution procedure. In reality, the service provider would allow slight violations of time-related constraints to find a feasible solution requiring only 23 vehicles.

filled with requests transferred from Wilrijk. This is particularly beneficial for requests that would be a driver's first or last job of the day, thanks to the direct effect on the route duration and wage cost.

Optimizing the third objective, solutions involving an average of 64.90 (May 18) and 68.00 (May 19) instead of 69 vehicles can be obtained in the scenario without joint route planning. Joint route planning further reduces the average fleet size to 63.75 (May 18) and 66.40 (May 19). This reduction mainly concerns the depot in Melsbroek, in combination with a net transfer of users from Melsbroek to Wilrijk. The corresponding increase in the average total cost relative to objective 2 is only 1.63% (May 18) and 1.30% (May 19), which is a particularly interesting finding from the service provider's point of view. Of course, it should be stressed that medium-term decisions on the fleet size cannot be based on results originating from two operating days. Ideally, this decision is based on data over a longer period of time, such that the service provider can determine his requirement of vehicles in normal circumstances. Unfortunately, since the data collection requires lots of manual interventions by the provider, it was not possible to provide additional data.

From a quality-related perspective, it is important to verify whether joint route planning causes any increase in the average ride time of the users, since a larger number of users may be transported jointly. Even though a minimal service level is guaranteed through an upper bound on each user's ride time, the average ride time remains a very relevant quality measure. Analyzing the minimal-cost solution on each operating day, joint route planning is found to increase the average user ride time from 21.30 to 21.58 minutes (May 18) and from 24.55 to 24.89 minutes (May 19), which is negligible. This implies that the greatest merit of joint route planning lies in a reduction of empty rides. For example, on May 18, the total empty distance traveled reduces from 4179.38 to 3886.29 km (-7.01%). Nevertheless, empty rides still account for almost half of the total distance traveled, since drawing efficient vehicle routes is complicated due to the tight quality constraints that characterize DARPs.

6.3.3.4 Reasons for request exchanges

This case study deals with different branches belonging to the same service provider, facilitating the implementation of a centralized decision making strategy. When two separate service providers decide to cooperate, they may not be willing to share all information on their user requests, both for privacy reasons and from a strategic perspective. Therefore, it would be interesting for them to know which requests are most suitable to be exchanged. It is unclear whether this assessment can be made in advance and if so, whether similar operational benefits can be obtained by sharing only those requests. The remainder of this section aims to identify reasons why requests are exchanged, analyzing the detailed route structures in the solutions obtained before.

Scenario		**Total**	**Veh**	**Wage**	F_t	F_W	F_M	**W**	**M**	**W→M**	**M→W**
Objective 1: minimization of traveling cost											
18.05	coop.	2136.47	2136.47	-	68.95	24.70	44.25	126.05	169.95	60.10	52.15
	no coop.	2264.21	2264.21	-	68.95	22.95	46.00	134.00	162.00		
19.05	coop.	2114.19	2114.19	-	69.00	23.60	45.40	118.65	122.35	26.75	21.40
	no coop.	2199.63	2199.63	-	69.85	24.00	45.85	124.00	117.00		
Objective 2: minimization of distance and wage cost											
18.05	coop.	8556.61	2512.85	6043.76	69.00	22.15	46.85	105.45	190.55	76.60	48.05
	no coop.	8836.29	2532.01	6304.28	69.00	23.00	46.00	134.00	162.00		
19.05	coop.	8391.43	2521.63	5869.80	69.00	25.05	43.95	121.95	119.05	26.90	24.85
	no coop.	8535.94	2509.08	6026.86	70.00	24.00	46.00	124.00	117.00		
Objective 3: hierarchical minimization of (1) fleet size and (2) distance and wage costs											
18.05	coop.	8695.80	2460.48	6235.33	63.75	25.95	37.80	163.60	132.40	54.50	84.10
	no coop.	8924.59	2473.68	6450.91	64.90	21.00	43.90	134.00	162.00		
19.05	coop.	8500.42	2497.40	6003.02	66.40	25.20	41.20	137.65	103.35	19.95	33.60
	no coop.	8575.29	2491.58	6083.71	68.00	24.00	44.00	124.00	117.00		

Table 6.10: Effect of cooperation through joint route planning in the real-life case study, considering different objectives.

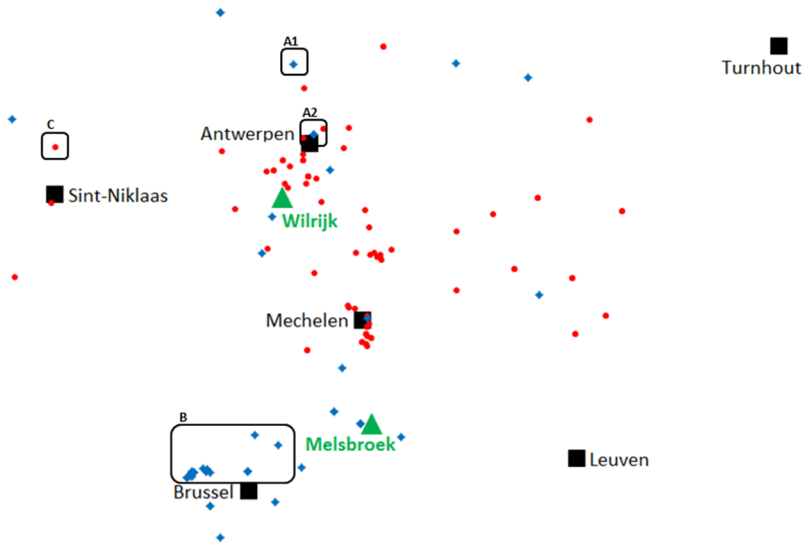

Figure 6.3: Home locations of the users whose request has been exchanged in the real-life case study.

Figure 6.3 is similar to Figure 6.2, but only shows the users being exchanged in the cost-minimal solution (objective 2). Consequently, the red dots (resp. blue diamonds) represent the home locations of users who were previously assigned to the branch in Wilrijk (resp. Melsbroek), but are now served by a vehicle originating from the depot in Melsbroek (resp. Wilrijk). Three specific groups of users are highlighted, since they are examples of the main reasons for exchanges that are identified. They are ranked in descending order of predictability:

1. A first group of exchanges is explained directly by the geographical situation of the users' home addresses. For example, consider users A1 and A2, who are living close to Antwerp and both request a trip to the rehabilitation center in Melsbroek early in the morning. Thanks to their coinciding time windows, it is immediately obvious why these users will be combined in a single ride. However, in this case, it is more efficient to deviate from the current assignment rules and serve these users with a vehicle that originates from the depot in Wilrijk, which covers less empty distance to reach the users' home addresses. This type of exchange is easily predictable in advance. Unfortunately, relatively few requests belong to this first category because of the limited overlap between both service areas. Nevertheless, in a setting where service providers prefer to have their own set of users, one might think of an agreement to preassign them based on their geographical characteristics. Note that this corresponds to the current practice of the real-life service provider considered in this study, who - in the scenario without cooperation -

Scenario		Total	Veh	Wage
18.05	joint route planning	8556.61	2512.85	6043.76
	assign by destination	8836.29	2532.01	6304.28
	assign by home	8764.87	2505.98	6258.89
	assign by average	8933.25	2586.28	6346.97
19.05	joint route planning	8391.43	2521.63	5869.80
	assign by destination	8535.94	2509.08	6026.86
	assign by home	8732.30	2563.97	6168.34
	assign by average	8862.98	2549.22	6313.77

Table 6.11: Effect of different request assignment strategies in the real-life case study, assuming a total cost minimization objective.

assigns users to the branch that is located closest to their outbound location. In this respect, it is interesting to investigate whether it is more efficient to systematically assign users based on (1) their *home* location or (2) the *geographical center* between their home and outbound location. Table 6.11 compares the three different assignment strategies for the second objective function. Similar results are obtained for the other objectives as well. On both operating days, the assignment based on the geographical center is strongly outperformed, but it is not clear which of the two other strategies is preferable. None of them matches the results of the fully centralized decision making strategy, which indicates that joint route planning is a prerequisite for obtaining substantial operational benefits.

2. A second group of exchanges can be explained indirectly, based on the fact that user requests come in pairs. For example, consider the group of users with label B in Figure 6.3. Based on their home location near Brussels, it might seem strange that in the scenario with joint route planning, these users are picked up by a vehicle from the branch in Wilrijk. This is explained by the fact that the vehicles from the depot in Wilrijk deliver some users at the rehabilitation center in Melsbroek (such as A1 and A2 in the previous example). While these users undergo their treatment, it would not be efficient for these vehicles to idle at the rehabilitation center, nor to return to their own service area being unloaded. Hence, they serve one or several local requests in the area around Melsbroek, i.c. with label B. However, since multiple request(s) are eligible, this type of exchanges is difficult to predict, despite being easy to explain in retrospect.

3. A last group of exchanges are impossible to predict, since they are only convenient relative to the actual structure of the routes. For example, consider user C, living north of Sint-Niklaas

(close to Wilrijk and Antwerp, but far from Melsbroek). This user is picked up at a hospital in Antwerp by an empty vehicle from the depot in Melsbroek. The reason for this unexpected finding is that after the delivery of this user, the vehicle picks up another user living south of Sint-Niklaas and requesting a ride to the rehabilitation center in Melsbroek. The exchange of user C cannot be explained taking an isolated view on his request. It can only be interpreted in a broad context, optimizing the entire set of routes in the solution. Therefore, it seems an impossible task for providers to predict this type of exchanges in advance.

6.4 Conclusions and future research

Even though centralized decision making through joint route planning has been a cost-effective practice among logistic service providers for many years, this strategy is unexplored in the domain of demand-responsive passenger transportation. This chapter quantifies the potential of joint route planning among dial-a-ride providers and the operational characteristics that influence the benefits. In addition, a real-life case study reveals the reasons why providers benefit from exchanging particular user requests among each other. The extent to which these exchanges are predictable determines how much information must be disclosed to make the cooperation succeed. From a technical point of view, the analysis of joint route planning is approached as a multi-depot variant of the dial-a-ride problem, in which each service provider exploits one or several depots. Joint route planning implies that requests may be served by vehicles originating from any depot. Without cooperation, users are preassigned to a particular service provider based on their geographical location. A large neighborhood search metaheuristic with additional periodic diversification is applied to solve this problem. Its excellent performance on single-depot and multi-depot instances is illustrated through tests on existing artificial data.

To analyze the effect of joint route planning, a new artificial data set with realistic characteristics is introduced. Comparing scenarios with and without joint route planning reveals a total distance saving between 5.61% and 16.73%, which is more modest than reported in freight transportation. The operational setting has a substantial impact on the magnitude of the benefits. Statistical analyses confirm that the potential of joint route planning increases for problem variants with smaller time windows, more long-distance trips, weaker clustering of users around their closest depot, more overlap between service areas or fewer users. Further research may focus on analyzing interactions between these effects. In addition, the influence of other operational characteristics may be identified, such as heterogeneity of users or combination constraints.

The concept of joint route planning is also applied to a real-life case study, involving a service

provider considering a cooperation between two of his branches. Several conclusions can be drawn which go beyond the particular characteristics of this case study. For example, the benefits of joint route planning are mainly explained by a reduction in empty distance, without affecting the service level offered to users. Unfortunately, the variable costs related to the distance traveled only account for a small part of the total cost incurred by the provider. Taking driver wages into account, rather than a mere minimization of the total distance traveled, the impact of joint route planning becomes smaller. In addition, real-life cooperations between separate service providers may be hampered by their reluctance to share full information for several reasons. The solutions with and without joint route planning are analyzed in order to discover the underlying reasons why requests are exchanged among the participating parties. Few exchanges turn out to be easily predictable in advance, which implies that the potential of joint route planning can only be fully exploited if service providers share complete information on their requests. However, the observed exchange patterns may provide useful insights to develop new local search operators tailored to the context of joint route planning.

This chapter mainly focuses on the operational effects of joint route planning. In reality, other factors may impede the implementation of such a centralized decision making strategy (Cruijssen et al., 2007c). The initialization phase often requires considerable investments to align information systems and policies. For example, the proposed form of cooperation has little chance of success if the participating providers offer a different quality of service. Moreover, trust and commitment are required among all participating parties. In this respect, joint route planning would be facilitated by the creation of an independent body that collects all user requests and divides them in a globally optimal manner among the participating service providers.

Finally, note that the analysis in this chapter is executed in terms of the joint benefits obtained by the participating service providers. This disregards the fact that all parties have to agree on a strategy to divide these benefits, which is necessary to guarantee the stability of the cooperation in the long term. Considerable research has already been conducted into profit sharing techniques in logistics, which may also be applied to demand-responsive passenger transportation. For example, Verdonck et al. (2016) compare three common sharing techniques (Shapley value, alternative cost avoided method, equal profit method) and find that the participating providers may prefer different techniques, e.g. depending on the size of their contribution. Moreover, the first two techniques do not guarantee to deliver a stable solution in which no subcoalition would benefit from leaving the grand coalition, whereas the latter technique may only be acceptable in the early phases of a growing cooperation. In summary, it is not straightforward for the participating providers to agree on the choice of a particular profit sharing technique.

Chapter 7

Conclusions and future outlook

The objective of this thesis was to investigate how service providers may obtain a more efficient balance between the service level offered to their customers and the corresponding operational costs, taking into account the influence of real-life problem characteristics. First, Section 7.1 summarizes the main findings of this thesis with respect to the research questions formulated in Chapter 1 and the aims of the different chapters visualized in Figure 7.1. Then, Section 7.2 analyzes these findings in order to formulate recommendations to the three types of stakeholders influencing the usability of dial-a-ride systems: the academic community, the service providers and the government. In other words, their decisions and actions determine the extent to which a system satisfies the needs of the users, who represent the fourth main stakeholder in the context of this thesis.

7.1 Conclusions

Dial-a-ride systems are typically invoked to provide collective demand-responsive transportation to people with reduced mobility, such as elderly and disabled. Chapter 1 explains that the demand for these services will increase in the coming years, due to demographic evolutions and developments in the healthcare sector. In addition, modern policy visions integrate them into the regular public transportation network. Providers of dial-a-ride services require efficient vehicle routing algorithms, taking into account all relevant real-life problem characteristics, to support their operational activities. Moreover, solving the dial-a-ride problem entails a considerable tradeoff of operational costs and service quality. From the provider's perspective, the most efficient routing solutions are created by combining multiple user requests into a single vehicle ride. However, the users face inconvenience because of the resulting detours and potential deviations from their preference times. Rather than unilaterally minimizing the operational costs subject to minimum quality requirements, this PhD

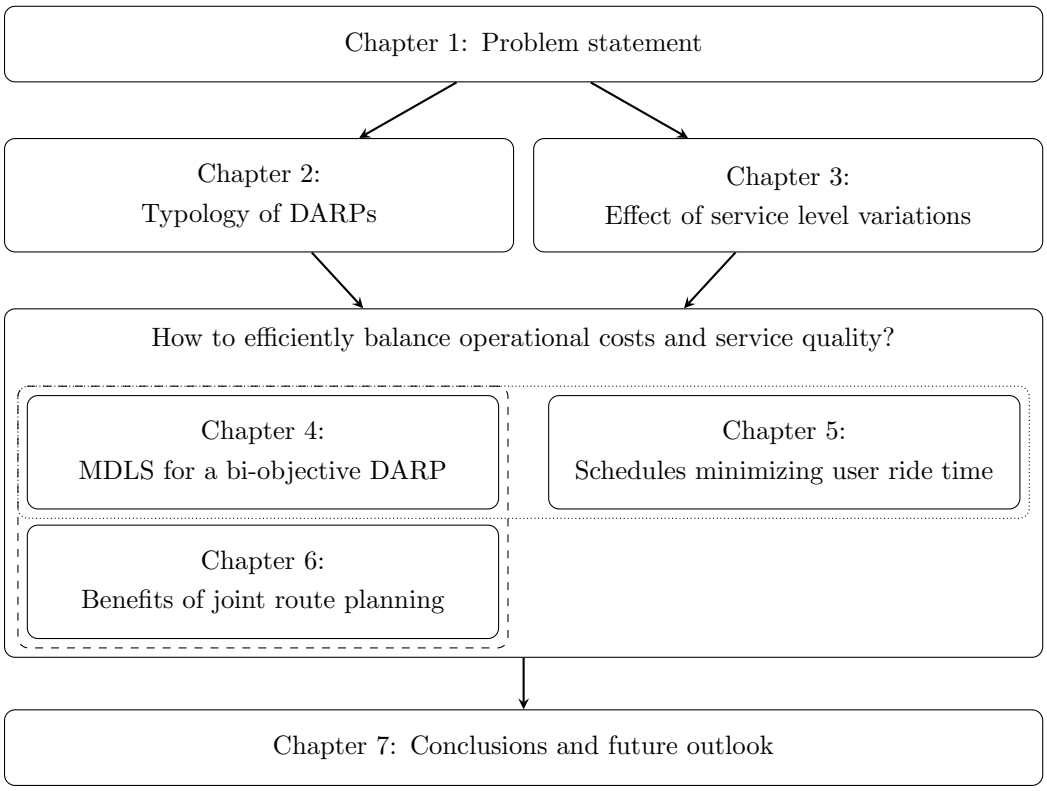

Figure 7.1: Thesis overview

thesis aims at providing insight into the tradeoff of operational costs and service quality, allowing service providers to implement a strategic quality vision and to make operational decisions in line with this policy. To this end, the following research questions are answered in this thesis:

(1) What is the state of the art in solving dial-a-ride problems with real-life characteristics?
(2) How important is the effect of service level variations on the operational costs?
(3) How can optimization techniques reveal the tradeoff of the service level and operational costs?
(4) How can operating strategies of service providers balance the service level and operational costs?

The first research question is answered by means of a literature review in Chapter 2. First, this review results in a typology of the problem features that have been included in solution algorithms. In recent years, authors have increasingly devoted attention to the incorporation of real-life characteristics, which ensures the practical applicability of their solution methods. These extensions were mainly related to heterogeneity (e.g. different types of users, configurable vehicles), advanced service designs (e.g. transfers, breaks) and limited availability of information (e.g. requests submitted in real time). However, the existing research fails to combine the advancements achieved in different categories and should to a larger extent focus on analyzing the impact of these additional real-life characteristics on the solution. Second, the review also compares different solution techniques that have been applied in the literature. Due to the complexity of the DARP, exact approaches are generally unable to solve real-life problems within reasonable computation times. Authors have mainly focused on the development of fast and efficient approximate approaches, such as metaheuristics and matheuristics. Particularly algorithms based on local search operations, possibly extended with a population strategy, have obtained outstanding results in recent years. However, most existing routing algorithms deliver a single solution that is optimized with respect to an operational objective. Even though a minimum service level is imposed by means of constraints, these algorithms do not optimize the service quality experienced by users, nor do they provide insight into the effect of changes in this quality level on the operational costs incurred by the provider. Consequently, this output does not deliver sufficient information for service providers to implement a strategic quality policy, being aware of the operational consequences.

Yet, as an answer to the second research question, the sensitivity analysis in Chapter 3 reveals that the operational costs incurred by a service provider are considerably affected by the settings of the two most common service level parameters, which are the maximum deviation from a user's preference time and the maximum detour a user may face. This is explained by the fact that stricter quality constraints reduce opportunities to efficiently combine multiple requests into the same ride. As a result, the vehicles need to cover additional distance to satisfy all requests, thereby increasing

the operational costs. Particularly for private service providers, this finding has important strategic implications due to the elasticity of the demand. Implementing a policy of excellent service quality may force providers to charge high rates, making their services unaffordable to people with limited budgets, whereas reductions in service quality may result in a loss of unsatisfied users. The size of this effect depends on the real-life characteristics applying to the operating context of the provider. For example, larger providers are relatively sensitive to service level variations, as they are better able to combine different user trips when the parameter choices become less stringent. Urban traffic conditions and heterogeneous user types limit the relative effect of variations in the service level.

The third research question gives rise to the development of a routing algorithm that allows service providers to analyze the tradeoff of operational costs and service quality. The multi-directional local search metaheuristic developed in Chapter 4 emphasizes this fundamental nature of the DARP. It delivers a set of non-dominating solutions with respect to two objectives. The first one corresponds to the frequently used operational objective of minimizing the total distance traveled, whereas the second objective represents the service quality by means of the total time users spend in the vehicles. The local search operators with respect to each objective are embedded in a variable neighborhood descent framework. This is guided by a new candidate list principle that reduces the computation time without sacrificing considerable solution quality. The multi-directional local search is supplemented with path relinking to obtain an extensive and diversified solution set. Even though only a single solution can be executed, this provides practitioners with valuable insights. For example, the pattern of the solution set reveals that a small deviation from the minimum-distance solution may cause a substantial improvement of the service level experienced by the users. Supporting decisions by means of such a tradeoff allows service providers to pursue a well-considered, consistent service policy. Moreover, Chapter 4 studies the effect of combination constraints, based on an application in patient transportation where drivers have different qualification levels and certain combinations of users are not allowed for medical reasons. Such combination constraints substantially affect the cost structure of a service provider, reducing the efficiency of the routing solutions. Since the proposed algorithm includes the minimization of the total user ride time as an objective, the results depend on the computation of the time schedule for the routes in a solution. To ensure consistency, Chapter 5 presents a scheduling heuristic that aims at minimizing the total ride time of all users in a given route. It outperforms a comparable procedure in the academic literature in terms of both computation time and solution quality. Particularly important is the strong reduction of incorrect infeasiblity declarations. Imposing additional constraints on the allocation of waiting time results in easier and more realistic schedules, avoiding that pickups of available users can be postponed (i.c. that excessive ride time is converted into unnecessary waiting time) if this increases other users' ride times. The fact that this realistic scheduling approach proves more restrictive in terms of feasibil-

ity illustrates why it seems advisable to rethink the current definition of the time-related constraints.

Whereas the third research question relates to algorithmic advancements in revealing the tradeoff of operational costs and service quality, the fourth research question focuses on the fact that strategic behavior of service providers - in addition to the aforementioned decisions on their service policy - may also affect this tradeoff. Chapter 6 shows that providers with a comparable quality vision and overlapping service areas may reduce their operational costs without deteriorating the service quality by implementing centralized decision making through joint route planning. Requests of competing service providers are allocated in such a manner that the joint routing costs are minimized. A large neighborhood search algorithm is introduced to compare the operational costs with and without cooperation. This reveals a total distance saving between 5.61% and 16.73%, which is more modest than reported in freight transportation. In practical applications, the benefits of joint route planning may strongly depend on the operating area (e.g. locations of the depots), the requests (e.g. clustered demand, number of requests, length of the trips) and the operational characteristics (e.g. allowed deviation from the user's preference time). The savings are principally explained by a reduction in the empty distance traveled, without affecting the service level offered to the users. In other words, joint route planning may indeed be a suitable strategy for providers to improve their operational efficiency. However, note that the variable costs related to the distance traveled may only account for a small part of the total cost incurred by the provider. The operational benefits of cooperation should also be weighed against its overhead cost and potential trust issues among the participating providers should be addressed.

7.2 Recommendations

Based on the findings of this thesis, a wide range of recommendations can be formulated to the different stakeholders influencing the usability of dial-a-ride systems. First, opportunities for future work are suggested to the research community, both with respect to real-life problem variants and technical aspects to facilitate the development of solution algorithms. Second, dial-a-ride providers are addressed by means of strategic advises to improve the balance of their operational efficiency and the service quality offered to users. Third, the role that policymakers have defined for dial-a-ride systems is assessed and the required supporting measures are identified.

7.2.1 Recommendations to the scientific community

The typology in Chapter 2 reveals that research on dial-a-ride problems has focused on various types of real-life problem characteristics, ensuring the practical applicability of solution approaches.

Unfortunately, this also explains why the research domain suffers from a lack of unification. Authors consider different generalizations of the standard problem, which causes difficulties in comparing the performances of solution approaches. However, it is unclear whether future research would benefit from establishing a new, enriched standard definition of the DARP, including an unambiguous formulation of some frequent characteristics in real-life applications (e.g. heterogeneous users, driver breaks). On the one hand, this would create a common basis for researchers to continue developing and solving extended problem variants. As a result, the contributions of different researchers would be more easily comparable and testable on new benchmark data, created according to the enriched standard problem definition. On the other hand, establishing such an enriched problem requires a rather arbitrary selection of additional problem characteristics. It is not clear which criteria could be used to decide whether particular characteristics may belong to the enriched standard definition or not. In addition, this entails the risk that authors will unilaterally study a single problem variant or tailor their algorithms to the corresponding benchmark data set. This would be contrary to the aim of this thesis, focusing on the incorporation of real-life problem features.

When analyzing any type of additional problem characteristic that applies to a real-life context, the role of the academic research is at least twofold. On the one hand, efficient solution approaches should be developed that take into account the additional feature. On the other hand, the effect of including this feature on the operational costs and the quality of solutions should be assessed under various operating circumstances. The latter type of analysis currently receives little attention, whereas this is essential to provide dial-a-ride companies with general managerial insights as a first step to support strategic or tactical decisions. For example, the results in Chapter 4 may be used to determine the required number of medically qualified drivers in the presence of combination constraints. This also applies to the new phenomenon of integrated dial-a-ride problems, which is likely to emerge in the coming years. The integration of dial-a-ride systems with public transportation is already developing in many countries, whereas innovative contributions in the literature advocate an integration with various related problems, ranging from healthcare scheduling to parcel delivery. Research on the potential benefits of such integrated approaches, taking into account the overhead costs (e.g. investments in new software), may convince service providers to exploit such innovations.

Given their main field of application, dial-a-ride systems are typically characterized by a strong customer-oriented focus. Consequently, relevant academic research ought to include the needs and expectations of current and potential users. For example, the motivation of this thesis can be found in the surveys on service quality discussed in Chapter 1. Noticeably, the literature on expectations and needs of users is scarce in comparison with contributions on solution algorithms. Furthermore, since the last analyses were performed several years ago, they provide no insight into the impact of

recent rich solution algorithms on the satisfaction of the users, nor do they reveal the expectations of users with regard to the innovations that will impact dial-a-ride systems in the near future. An update would be desirable to ensure that the feedback of users in real-life applications remains an important source of inspiration for future research. On the one hand, knowing the users' needs may result in more real-life constraints taken into account by solution approaches. On the other hand, the service-related goals in multi-objective approaches may be expressed in a more comprehensive manner, taking into account the relative importance of all quality expectations that users may have.

From an algorithmic perspective, the solution approaches developed in Chapters 4 and 6 confirm that metaheuristics based on local search are a powerful tool to obtain fast and efficient solutions for large-scale dial-a-ride instances. The efficiency of a particular type of metaheuristic is no intrinsic feature of the framework, but depends on the specific implementations of operators and other algorithmic components. Chapter 2 shows that the state-of-the-art algorithms of the past decade included several types of metaheuristics, whereas different implementations of the same metaheuristic may not be equally efficient. Researchers should gain more insight into the reasons why particular combinations of operators and other algorithmic components are preferable under certain operating circumstances. Assessment techniques, e.g. based on multilevel regression analysis, were developed for this purpose. The recent emergence of hybrid metaheuristics fits into this discussion. Whereas standalone population-based metaheuristics could not reach outstanding solutions in the past, e.g. since it is difficult to define strategic crossover operators to combine desirable characteristics of two encoded solutions, recent work shows that they are useful to provide an overarching framework in which local search approaches can be integrated. For rich problem variants, commonly used local search strategies may be supplemented with tailored operators to exploit the specific problem structure. Finally, although most metaheuristics in the current state of the art are sufficiently fast to serve in a static problem context, it remains important to ensure faster response times in dynamic settings. For example, the algorithms developed in this thesis benefit from the accelerating candidate list principle explained in Chapter 4 and the faster scheduling heuristic presented in Chapter 5. Still, many other opportunities to improve the balance of computation times and solution efficiency may be identified. In this respect, matheuristics constitute a promising research direction, as they exploit the combination of time-efficient metaheuristics and exact mathematical programming techniques, such that high-quality solutions tend to be obtained early in the optimization process.

7.2.2 Recommendations to service providers

Service providers tend to devote little attention to the methodology and the objectives of their vehicle routing software, if they have it at all. In addition, manual interventions into the solutions

are often required to repair violated real-life constraints (e.g. availability of drivers, preferences of users, traffic circumstances) not taken into account by standard software, which harms the eventual efficiency of the solution. Therefore, service providers should be advised to invest in vehicle routing software that is tailored to their specific needs and allows them to obtain a competitive advantage. The clearest example is presented in Chapter 5, which proves that the objective of the scheduling heuristic in a solution software may heavily influence the service quality experienced by users, often without affecting the operational costs incurred by the service provider. The results in Chapters 3 and 4 suggest that providers can only develop a strategic quality policy based on routing software which helps them understand the tradeoff of operational costs and service quality in their specific environment. In that regard, private service providers should be aware of the elasticity of demand, which has not been considered in this thesis. They may partially recover the additional costs of an increased service level through an extension of their customer base, whereas a reduced service level may trigger a downward spiral.

More generally, this thesis illustrates that the operational efficiency of dial-a-ride systems heavily depends on the extent to which different user trips can be combined in the same vehicle. Some combinations are impossible due to factors that are beyond the control of the provider, such as the combination restrictions discussed in Chapter 4. Stringent service level constraints also reduce the potential of request combinations, but may be considered as a tool to ensure customer satisfaction and an asset to improve long-term customer loyalty. Most interesting, however, is strategic behavior of the service provider that creates operational benefits without affecting the service quality experienced by users. Exploiting the benefits of multiple depots or a joint route planning, as proposed in Chapter 6, is an interesting strategy to reduce the empty distance traveled without any harmful side effects on customer satisfaction. However, such opportunities are numerous. For example, the topology in Chapter 2 identifies the integration of dial-a-ride services and related problems, such as the scheduling of treatments in a healthcare center, as a promising, yet rarely explored strategy to improve the joint operational efficiency of both parties. Whenever the operational benefits of either form of cooperation or integration are demonstrated, it will be vital for service providers to embrace such innovations in order to remain a competitive player on the rapidly changing dial-a-ride market. Obviously, clear agreements should be made whenever this implies that part of their operational autonomy (e.g. route planning, allocation of requests) needs to be entrusted to another party.

Finally, remark that many aspects of this thesis do not exclusively apply to dial-a-ride services. Also providers of related types of transportation, even not necessarily involving people with reduced mobility, may take advantage of the proposed algorithms and build upon the conclusions formulated in this thesis. For example, the centralized planning approach suggested in Chapter 6 may also be

applied for solving vehicle routing problems in modern ride sharing systems (e.g. Uber). In these systems, on-demand transportation requests of users may be executed by private drivers using their own car. Hence, these systems can be modeled as a dynamic cooperation of providers which dispose of a single vehicle each. In addition, the origin and destination depots of the vehicles may be located at different physical locations if the private drivers are making a trip themselves. However, the basic principle of assigning requests to vehicles in a distance-efficient and time-feasible manner through centralized decision making remains the same.

7.2.3 Recommendations to policy makers

As discussed in Chapter 1, modern policy visions aim at the integration of dial-a-ride services and public transportation. Regular public transportation will be restricted to (sub)urban and interurban services, whereas dial-a-ride systems will provide on-demand transportation for all people (with and without mobility restrictions) in rural areas, as well as for people with reduced mobility in urban areas. Users will submit a single request for their entire trip. Their itineraries and potential transfer locations will need to be determined by a routing algorithm that optimizes the routes and schedules of the dial-a-ride vehicles, given all requests and the timetables of public transportation services. However, the typology in Chapter 2 reveals that scientific research has barely focused on this complicated variant. In fact, the only contribution on the integrated dial-a-ride problem avoids a synchronization of schedules and neglects waiting times during transfers, assuming that regular buses have an infinite frequency. It also ignores that some users cannot use public transportation due to their mobility restriction. To ensure a successful implementation of their integrated mobility vision, governments need to invest in the development of an appropriate routing software and make it available to the body coordinating the routing task. Attention must also be devoted to a dynamic handling of requests submitted shortly before the desired departure time, taking into account that many users are unable to specify the time of their (inbound) trip in advance. This unavailability of information may considerably affect the efficiency of routing solutions and the required government subsidies. By all means, the current approach of many service providers, requiring users to submit their requests one day in advance, is no longer tenable when dial-a-ride services replace the regular public transportation in rural areas.

Finally, this thesis gives rise to several suggestions with respect to the compensations granted to dial-a-ride providers for the transportation of people with reduced mobility. For example, consider the *compensation decree* issued by the Flemish government, discussed in Chapter 1. First, the decree assumes a single subsidized provider in each of the 27 transportation areas, even if several companies are prepared to comply with its requirements. Based on the findings in Chapter 6, it may be more

cost-efficient to allow multiple compensated providers in each transportation area and establish a coordinating body that assigns requests in a globally optimal manner. Particularly if the depots of these providers are well spread, the coverage of the transportation area may be improved and long empty rides may be avoided. Second, the rules according to which compensation is granted to the providers may cause undesirable consequences. Although the decree imposes stringent constraints related to the availability of the providers' services, the design of their vehicles and the education of drivers, some measurable aspects of service quality in the routing solutions receive little attention. For example, the maximum detour due to the collective nature of the services may not exceed the fixed duration of 30 minutes, which is unreasonably large for short trips. Moreover, the decree even encourages service providers to exploit such excesses due to the fact that only the distances traveled by a non-empty vehicle are compensated. This exposes the need for a revised compensation decree which also accounts for the interests of the users, rewarding efficient tradeoffs of operational costs and service quality by dial-a-ride providers.

Appendix A

Literature overview table

The overview table in this appendix provides full details on the problem characteristics and the solution methods applied in each of the papers that were discussed in Chapter 2. Column 1 contains the references to these contributions in alphabetic order. Columns 2-24 correspond to the problem characteristics discussed in Section 2.2. They indicate whether a certain problem characteristic is studied in the corresponding paper. More specifically, columns 2-6 refer to the characteristics of the standard DARP (Section 2.2.1), columns 7-13 to heterogeneity (Section 2.2.2.1), columns 14-18 to more complex routing properties (Section 2.2.2.2) and columns 19-24 to stochastic or dynamic information (Section 2.2.4). Column 25 indicates whether the corresponding paper applies a direct bi/multi-objective solution approach (Section 2.2.3.2). Column 26 gives more details on the solution method applied (Section 2.3). Column 27 enumerates all aspects included in the objective function. Columns 28-31 mention the data used to test the corresponding solution method (Section 2.3.3). Finally, an explanation of the abbreviations used can be found at the bottom of the table.

Reference	User time preference	Max. user ride time	Service duration	Max. route duration	Max. vehicle capacity	Heterogeneous users	Upgrading conditions	Accompanying person	Configurable vehicles	Priority requests	Isolated transportation	Driver qualifications	Transfers	Integration	Split requests	Breaks	Multiple depots	Time-dep. travel times	Stochastic travel times	Dynamic context	Stochastic requests	Probalistic requests	Stochastic user behavior	Multi-objective	Solution method	Objective function	Data based on C&L (2003)	Data based on C (2006)	Other artificial data	Real-life data
Aldaihani and Dessouky (2003)	x				x								x	x											TS	Distance, user travel time				x
Atahran et al. (2014)	x		x		x																			x	GA	Routing cost, time window violations, user waiting time, CO2 emission	x		x	
Attanasio et al. (2004)	x	x	x	x	x															x					TS	Users served	x			
Baugh et al. (1998)	x				x																			x	SA	Routing costs, time window violations, fleet size				x
Beaudry et al. (2010)	x	x	x		x	x		x	x	x	x					x	x			x					TS	Vehicle travel time, deviation from user preference time				x
Berbeglia et al. (2012)	x	x	x		x															x					H-TS	Users served	x	x		
Borndörfer et al. (1999)	x		x	x	x	x		x									x								BC	Distance				x
Braekers et al. (2014)	x	x	x	x	x	x	x	x									x								BC, DA	Routing costs		x		
Braekers and Kovacs (2016)	x	x	x	x	x	x	x	x																	BC, LNS	Routing costs				x
Chassaing et al. (2016)	x	x	x	x	x																				ELS	Routing costs	x	x		
Chevrier et al. (2012)	x	x			x																			x	GA	Fleet size, route duration, user delays			x	
Coppi et al. (2013)	x	x		x	x	x											x								CG	Routing costs				x
Cordeau and Laporte (2003)	x	x	x	x	x																				TS	Routing costs	x			x
Cordeau (2006)	x	x	x	x	x																				BC	Routing costs		x		
Coslovich et al. (2006)	x	x																		x					IH	Deviation from user preference time, user ride time			x	
Crawford et al. (2007a,b)	x	x	x	x	x																				H-ACO	Routing costs				
Cremers et al. (2009)	x	x			x	x														x					GA	Expected routing costs			x	
Cubillos et al. (2007)	x	x			x												x								GA	Fleet size, vehicle travel time, vehicle slack time, user waiting time, user ride time			x	
Cubillos et al. (2009)	x				x																				GA	Fleet size, vehicle travel time, vehicle slack time, user waiting time, user ride time	x			
Desrosiers et al. (1986)	x		x		x																				DP	Distance				x
Dessouky et al. (2003)	x	x	x	x	x	x																			IH	Routing costs, penalties soft constraints, life-cycle environmental costs				x
Detti et al. (2016)	x	x	x		x	x					x	x					x								TS, VNS	Fixed and variable routing costs, vehicle travel time, patients transported, vehicle waiting time, user waiting time			x	x

Continued on next page

Reference	User time preference	Max. user ride time	Service duration	Max. route duration	Max. vehicle capacity	Heterogeneous users	Upgrading conditions	Accompanying person	Configurable vehicles	Priority requests	Isolated transportation	Driver qualifications	Transfers	Integration	Split requests	Breaks	Multiple depots	Time-dep. travel times	Stochastic travel times	Dynamic context	Stochastic requests	Probalistic requests	Stochastic user behavior	Multi-objective	Solution method	Objective function	Data based on C&L (2003)	Data based on C (2006)	Other artificial data	Real-life data
Diana and Dessouky (2004)	x	x	x					x																	IH	Distance, user ride time, vehicle idle time				x
Garaix et al. (2011)	x	x	x		x												x								CG	Occupancy rate (= total user ride time over total vehicle travel time)	x	x	x	x
Gschwind and Irnich (2014)	x	x	x		x																				BCP	Routing costs		x		
Guerriero et al. (2013)	x	x	x	x	x																				H-GRASP	Routing costs	x		x	x
Häll et al. (2009)	x	x	x	x	x	x							x	x											BC	Routing costs			x	
Häll and Peterson (2013)	x	x	x		x	x	x										x			x					LNS	Routing costs, vehicle waiting time, user ride time, user waiting time				x
Häme (2011)	x	x			x															x					DP	Route duration, user waiting time, user ride time			x	
Häme and Hakula (2013)	x	x	x		x																				DP	Users served		x		
Häme and Hakula (2014)	x	x	x		x																				DP	Users served		x		
Hanne et al. (2009)	x	x			x	x		x	x	x	x					x	x			x					IH, GA	Total lateness, total earliness, vehicle travel time, total user ride time				x
Heilporn et al. (2011)	x	x	x	x	x																		x		BC	Routing costs, expected delays			x	
Ho and Haugland (2011)	x	x	x	x	x																	x			TS	Expected routing costs	x			
Hu and Chang (2015)	x	x	x		x													x							BP	Routing costs			x	
Ioachim et al. (1995)	x		x		x	x											x								CG	Distance				x
Jain and Van Hentenryck (2011)	x	x	x	x	x																				H-LNS	Routing costs	x			
Jaw et al. (1986)	x	x	x		x																				IH	Deviation from user preference time, user ride time, active vehicle time, vehicle slack time, system workload			x	x
Jørgensen et al. (2007)	x	x	x	x	x												x								GA	Vehicle travel time, user ride time, user waiting time, route duration, time window violations, maximum user ride time violations, maximum route duration violations	x			
Karabuk (2009)	x	x		x	x	x	x		x							x	x								CG	Users served, distance				x
Kirchler and Wolfler Calvo (2013)	x	x	x	x	x																				TS	Routing costs, user ride time, user waiting time aboard, route duration, early arrivals, users served	x			
Lehuédé et al. (2013)	x	x	x	x	x												x							x	LNS	Maximum user ride time, user ride time, user waiting time, vehicle travel time, fleet size				x

Continued on next page

Reference	User time preference	Max. user ride time	Service duration	Max. route duration	Max. vehicle capacity	Heterogeneous users	Upgrading conditions	Accompanying person	Configurable vehicles	Priority requests	Isolated transportation	Driver qualifications	Transfers	Integration	Split requests	Breaks	Multiple depots	Time-dep. travel times	Stochastic travel times	Dynamic context	Stochastic requests	Probalistic requests	Stochastic user behavior	Multi-objective	Solution method	Objective function	Data based on C&L (2003)	Data based on C (2006)	Other artificial data	Real-life data
Li et al. (2016)	x		x	x	x																				ALNS	Profit from parcels, profit from user rides, routing costs, discount for detours	x		x	
Liu et al. (2015)	x	x	x	x		x		x	x							x									BC	Vehicle travel time				x
Luo and Schonfeld (2007)	x	x	x		x																				IH	Vehicle travel time, user ride time			x	x
Luo and Schonfeld (2011)	x	x			x															x					IH	Vehicle travel time, user ride time			x	
Maalouf et al. (2014)	x	x			x															x				x	FL	Distance, user ride time			x	
Masmoudi et al. (2017)	x	x	x	x	x	x	x	x																	H-GA	Routing costs	x	x		
Masmoudi et al. (2016)	x	x	x	x	x	x	x	x								x	x								ALNS, H-BA	Routing costs	x	x		
Masson et al. (2014)	x	x	x	x	x								x				x								ALNS	Distance	x			x
Mauri and Lorena (2006)	x	x	x	x	x												x								SA	Distance, fleet size, route duration, user ride time, waiting time, maximum route duration violations, maximum user ride time violations, maximum waiting time violations, vehicle capacity violations, time window violations	x			
Melachrinoudis et al. (2007)	x				x												x								TS	Routing costs, time window violations, user ride time				x
Muelas et al. (2013)	x	x			x																				VNS	Routing costs			x	
Muelas et al. (2015)	x	x			x																				VNS	Routing costs			x	
Oberscheider and Hirsch (2016)	x	x	x	x	x	x			x		x					x	x								TS	Vehicle travel time, overtime				x
Paquette et al. (2013)	x	x	x	x	x	x																		x	TS	Routing costs, user waiting time, user ride time	x	x		x
Parragh et al. (2009)	x	x	x		x																			x	IVNS	Routing costs, user ride time		x		
Parragh et al. (2010b)	x	x	x	x	x																				VNS	Routing costs	x			
Parragh (2011)	x	x	x	x	x	x	x	x																	BC	Routing costs, vehicle waiting time		x		
Parragh et al. (2012)	x		x	x	x	x	x	x								x									CG	Routing costs		x		x
Parragh et al. (2014)	x		x	x											x										BP, VNS	Profits		x	x	
Parragh and Schmid (2013)	x	x	x	x	x																				H-CG, LNS	Routing costs	x	x		
Psaraftis (1980)					x															x					DP	Vehicle travel time, user waiting time, user ride time			x	
Psaraftis (1983)	x																								DP	Vehicle travel time			x	
Qu and Bard (2013)	x				x	x	x		x																ALNS	Fleet size, distance, user ride time		x	x	x

Continued on next page

Reference	User time preference	Max. user ride time	Service duration	Max. route duration	Max. vehicle capacity	Heterogeneous users	Upgrading conditions	Accompanying person	Configurable vehicles	Priority requests	Isolated transportation	Driver qualifications	Transfers	Integration	Split requests	Breaks	Multiple depots	Time-dep. travel times	Stochastic travel times	Dynamic context	Stochastic requests	Probalistic requests	Stochastic user behavior	Multi-objective	Solution method	Objective function	Data based on C&L (2003)	Data based on C (2006)	Other artificial data	Real-life data
Rekiek et al. (2006)	x	x	x	x	x	x			x								x								GA	Vehicle efficiency (= exploitation of vehicle capacity)				x
Ritzinger et al. (2016)	x	x	x	x	x																				H-LNS	Vehicle travel time	x	x		
Røpke et al. (2007)	x	x	x		x																				BC	Routing costs	x	x		
Rubinstein et al. (2012)	x	x			x	x												x							IH	Time window violations			x	x
Santos and Xavier (2015)	x				x															x					GRASP	Users served, cost payed by all users			x	
Schilde et al. (2011)	x	x		x	x															x	x				SVNS	Time window violations, fleet size, route duration			x	
Schilde et al. (2014)	x	x		x	x													x	x	x					SVNS	Time window violations, user ride time violations, fleet size, route duration			x	
Teodorovic and Radivojevic (2000)	x	x			x															x				x	FL	Distance, vehicle waiting time; total time, detour of previously assigned users			x	
Toth and Vigo (1997)	x	x	x		x	x		x																	TT	Routing costs, deviation from user preference time				x
Wolfler Calvo and Touati-Moungla (2011)	x				x													x							TS	Users served, fleet size, user ride time			x	
Wong and Bell (2006)	x	x	x		x	x		x									x								IH	Vehicle travel time, user ride time, taxi costs			x	
Xiang et al. (2006)	x	x	x		x	x	x					x													SO	Routing costs, driver wage (first); empty distance, wage for driving an empty vehicle or waiting, user ride time (second)			x	
Xiang et al. (2008)	x	x	x	x	x	x	x					x						x	x	x					SO	Routing costs, driver wage (first); empty distance, wage for driving an empty vehicle or waiting, user ride time (second)			x	
Zhang et al. (2015)	x	x	x	x				x								x									H-GA	Requests served (first), total vehicle travel time (second)	x	x		x
Ziliaskopoulos and Kozanidis (2006)	x		x		x																				DP	Vehicle travel time	x			x

Solution methods:
ACO = ant colony optimization, (A)LNS = (adaptive) large neighborhood search, BA = bee algorithm, BC = branch-and-cut, BCP = branch-and-cut-and-price, BP = branch-and-price, CG = column generation, CP = constraint programming, DA = deterministic annealing, DP = dynamic programming, ELS = evolutionary local search, FL = fuzzy logic, GA = genetic algorithm, H = hybrid, IH = insertion heuristic, (I/S)VNS = (iterated/stochastic) variable neighborhood search, SA = simulated annealing, SOF = secondary objective function, TS = tabu search, TT = tabu thresholding

List of Symbols

P	Set of pickup nodes $= \{1, 2, ..., n\}$
D	Set of corresponding delivery nodes $= \{n+1, n+2, ..., 2n\}$
N	Set of all nodes, including the start depot 0 and end depot $2n+1$
K	Set of all vehicles
x_{ij}^k	Binary decision variable: equals 1 only if vehicle $k \in K$ traverses the arc between nodes $i \in N$ and $j \in N$
B_i	Real decision variable: start time of the service in node $i \in N$
L_i	Real decision variable: ride time of user $i \in P$, excluding his service duration
Q_i	Integer decision variable: load upon leaving node $i \in N$
c_{ij}	Cost associated with the arc between nodes $i \in N$ and $j \in N$
t_{ij}	Direct travel time associated with the arc between nodes $i \in N$ and $j \in N$
e_i	Lower time window bound imposed on the start time of the service in node $i \in N$
l_i	Upper time window bound imposed on the start time of the service in node $i \in N$
d_i	Service duration in node $i \in N$
L_i	Maximum user ride time for user $i \in P$, excluding his service duration
T_k	Maximum route duration for vehicle $k \in K$
q_i	Net number of users boarding in node $i \in N$

List of Tables

List of Figures

Bibliography

Aldaihani, M., Dessouky, M. M., 2003. Hybrid scheduling methods for paratransit operations. Computers & Industrial Engineering 45 (1), 75–96.

Anderson, D., Sweeney, D., Williams, T., Freeman, J., Shoesmith, E., 2010. Statistics for Business and Economics. CENGAGE Learning Business Press.

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

Ascheuer, N., Krumke, S. O., Rambau, J., 2000. Online dial-a-ride problems: Minimizing the completion time. Lecture Notes in Computer Science 1770, 639–650.

Atahran, A., Lenté, C., T'kindt, V., 2014. A multicriteria dial-a-ride problem with an ecological measure and heterogeneous vehicles. Journal of Multi-Criteria Decision Analysis 21 (5-6), 279–298.

Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing 30 (3), 377–387.

Baugh, J. W., Kakivaya, G. K. R., Stone, J. R., 1998. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. Engineering Optimization 30 (2), 91–123.

Beaudry, A., Laporte, G., Melo, T., Nickel, S., 2010. Dynamic transportation of patients in hospitals. OR Spectrum 32 (1), 77–107.

Ben-Akiva, M., Benjamin, J., Lauprete, G. J., Polydoropoulou, A., 1996. Impact of advanced public transportation systems on travel by dial-a-ride. Transportation Research Record 1557, 72–79.

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

Berbeglia, G., Cordeau, J.-F., Laporte, G., 2012. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. INFORMS Journal on Computing 24 (3), 343–355.

Berbeglia, G., Pesant, G., Rousseau, L.-M., 2011. Checking the feasibility of dial-a-ride instances using constraint programming. Transportation Science 45 (3), 399–412.

Borndörfer, R., Grötschel, M., Klostermeier, F., Küttner, C., 1999. Telebus Berlin: Vehicle scheduling in a dial-a-ride system. Lecture Notes in Economics and Mathematical Systems 471, 391–422.

Braekers, K., Caris, A., Janssens, G. K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transportation Research Part B: Methodological 67, 166–186.

Braekers, K., Kovacs, A., 2016. A multi-period dial-a-ride problem with driver consistency. Transportation Research Part B: Methodological 94, 355–377.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A. A., 2014. Rich vehicle routing problem: Survey. ACM Computing Surveys (CSUR) 47 (2), 32.

Chassaing, M., Duhamel, C., Lacomme, P., 2016. An ELS-based approach with dynamic probabilities management in local search for the dial-a-ride problem. Engineering Applications of Artificial Intelligence 48, 119–133.

Chevrier, R., Liefooghe, A., Jourdan, L., Dhaenens, C., 2012. Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. Applied Soft Computing 12 (4), 1247–1258.

Clímaco, J. C. N., Craveirinha, J. M. F., Pascoal, M. M. B., 2006. An automated reference point-like approach for multicriteria shortest path problems. Journal of Systems Science and Systems Engineering 15 (3), 314–329.

Coppi, A., Detti, P., Raffaelli, J., 2013. A planning and routing model for patient transportation in health care. Electronic Notes in Discrete Mathematics 41, 125–132.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Operations Research 54 (3), 573–586.

Cordeau, J.-F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30 (2), 105–119.

Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological 37 (6), 579–594.

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

Cortés, C. E., Matamala, M., 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.

Coslovich, L., Pesenti, R., Ukovich, W., 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European Journal of Operational Research 175 (3), 1605–1615.

Crawford, B., Castro, C., Monfroy, E., 2007a. Solving dial-a-ride problems with a low-level hybridization of ants and constraint programming. Lecture Notes in Computer Science 4528, 317–327.

Crawford, B., Castro, C., Monfroy, E., Cubillos, C., 2007b. Decomposition approach to solve dial-a-ride problems using ant computing and constraint programming. Lecture Notes in Computer Science 4729, 448–457.

Cremers, M. L. A. G., Klein Haneveld, W. K., van der Vlerk, M. H., 2009. A two-stage model for a day-ahead paratransit planning problem. Mathematical Methods of Operations Research 69 (2), 323–341.

Cruijssen, F., Bräysy, O., Dullaert, W., Fleuren, H., Salomon, M., 2007a. Joint route planning under varying market conditions. International Journal of Physical Distribution & Logistics Management 37 (4), 287–304.

Cruijssen, F., Cools, M., Dullaert, W., 2007b. Horizontal cooperation in logistics: opportunities and impediments. Transportation Research Part E: Logistics and Transportation Review 43 (2), 129–142.

Cruijssen, F., Cools, M., Dullaert, W., 2007c. Horizontal cooperation in logistics: opportunities and impediments. Transportation Research Part E: Logistics and Transportation Review 43 (2), 129–142.

Cubillos, C., Rodríguez, N., Crawford, B., 2007. A study on genetic algorithms for the DARP problem. Lecture Notes in Computer Science 4527, 498–507.

Cubillos, C., Urra, E., Rodríguez, N., 2009. Application of genetic algorithms for the DARPTW problem. International Journal of Computers Communications & Control 4 (2), 127–136.

De Lijn, 2016. Tarievenfolder 2016 (in Dutch).
Retrieved from `delijn.be/nl/overdelijn/organisatie/publicaties`.

Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences 6 (3-4), 301–325.

Dessouky, M., Rahimi, M., Weidner, M., 2003. Jointly optimizing cost, service, and environmental performance in demand-responsive transit scheduling. Transportation Research Part D: Transport and Environment 8 (6), 433–465.

Detti, P., Papalini, F., de Lara, G. Z. M., 2016. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. Omega (forthcoming).

Diana, M., Dessouky, M. M., 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research Part B: Methodological 38 (6), 539–557.

Diana, M., Dessouky, M. M., Xia, N., 2006. A model for the fleet sizing of demand responsive transportation services with time windows. Transportation Research Part B: Methodological 40 (8), 651–666.

Doerner, K. F., Salazar-González, J.-J., 2014. Pickup-and-delivery problems for people transportation. In: Toth, P., Vigo, D. (Eds.), Vehicle Routing: Problems, Methods, and Applications. pp. 193–212.

Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational physics 104 (1), 86–92.

European Commission, 2016. Public health - ECHI data tool.
Retrieved from `ec.europa.eu/health/dyna/echi/datatool/index.cfm?indlist=40a`.

Feuerstein, E., Stougie, L., 2001. On-line single-server dial-a-ride problems. Theoretical Computer Science 268 (1), 91–105.

Firat, M., Woeginger, G. J., 2011. Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh. Operations Research Letters 39 (1), 32–35.

FOD Economie, 2016. Bevolkings- en huishoudensvooruitzichten (in Dutch).
Retrieved from `statbel.fgov.be/nl/statistieken/cijfers/bevolking/vooruitzichten`.

Fu, L., 2002. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. Transportation Research Part B: Methodological 36 (6), 485–506.

Fu, L., Teply, S., 1999. On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. Computer-Aided Civil and Infrastructure Engineering 14 (5), 309–319.

Garaix, T., Artigues, C., Feillet, D., Josselin, D., 2010. Vehicle routing problems with alternative paths: An application to on-demand transportation. European Journal of Operational Research 204 (1), 62–75.

Garaix, T., Artigues, C., Feillet, D., Josselin, D., 2011. Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation. Computers & Operations Research 38 (10), 1435–1442.

Gschwind, T., 2015. Route feasibility testing and forward time slack for the synchronized pickup and delivery problem. Tech. rep.

Gschwind, T., Irnich, S., 2014. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. Transportation Science 49 (2), 335–354.

Guerriero, F., Bruni, M. E., Greco, F., 2013. A hybrid greedy randomized adaptive search heuristic to solve the dial-a-ride problem. Asia-Pacific Journal of Operational Research 30 (1), 1–17.

Häll, C. H., Andersson, H., Lundgren, J. T., Värbrand, P., 2009. The integrated dial-a-ride problem. Public Transport 1 (1), 39–54.

Häll, C. H., Peterson, A., 2013. Improving paratransit scheduling using ruin and recreate methods. Transportation Planning and Technology 36 (4), 377–393.

Häme, L., 2011. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. European Journal of Operational Research 209 (1), 11–22.

Häme, L., Hakula, H., 2013. Routing by ranking: A link analysis method for the constrained dial-a-ride problem. Operations Research Letters 41 (6), 664–669.

Häme, L., Hakula, H., 2014. A maximum cluster algorithm for checking the feasibility of dial-a-ride instances. Transportation Science 49 (2), 295–310.

Hanne, T., Melo, T., Nickel, S., 2009. Bringing robustness to patient flow management through optimized patient transports in hospitals. Interfaces 39 (3), 241–255.

Hansen, P., Mladenović, N., Pérez, J. A. M., 2010. Variable neighbourhood search: methods and applications. Annals of Operations Research 175 (1), 367–407.

Haugland, D., Ho, S. C., 2010. Feasibility testing for dial-a-ride problems. Lecture Notes in Computer Science 6124, 170–179.

Heilporn, G., Cordeau, J.-F., Laporte, G., 2011. An integer L-shaped algorithm for the dial-a-ride problem with stochastic customer delays. Discrete Applied Mathematics 159 (9), 883–895.

Hickman, M. D., Blume, K. L., 2001. An investigation of integrated transit service. Tech. rep.

Ho, S. C., Haugland, D., 2011. Local search heuristics for the probabilistic dial-a-ride problem. OR Spectrum 33 (4), 961–988.

Hu, T.-Y., Chang, C.-P., 2015. A revised branch-and-price algorithm for dial-a-ride problems with the consideration of time-dependent travel cost. Journal of Advanced Transportation 49 (6), 700–723.

Hunsaker, B., Savelsbergh, M., 2002. Efficient feasibility testing for dial-a-ride problems. Operations Research Letters 30 (3), 169–173.

Hyytiä, E., Penttinen, A., Sulonen, R., 2010. Congestive collapse and its avoidance in a dynamic dial-a-ride system with time windows. Lecture Notes in Computer Science 6148, 397–408.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M. M., Villeneuve, D., 1995. A request clustering algorithm for door-to-door handicapped transportation. Transportation Science 29 (1), 63–78.

Jain, S., Van Hentenryck, P., 2011. Large neighborhood search for dial-a-ride problems. Lecture Notes in Computer Science 6876, 400–413.

Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., Wilson, N. H. M., 1986. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. Transportation Research Part B: Methodological 20, 243–257.

Jørgensen, R. M., Larsen, J., Bergvinsdottir, K. B., 2007. Solving the dial-a-ride problem using genetic algorithms. Journal of the Operational Research Society 58 (10), 1321–1331.

Jozefowiez, N., Semet, F., Talbi, E.-G., 2008. Multi-objective vehicle routing problems. European journal of operational research 189 (2), 293–309.

Karabuk, S., 2009. A nested decomposition approach for solving the paratransit vehicle scheduling problem. Transportation Research Part B: Methodological 43 (4), 448–465.

Kirchler, D., Wolfler Calvo, R., 2013. A granular tabu search algorithm for the dial-a-ride problem. Transportation Research Part B: Methodological 56, 120–135.

Krajewska, M. A., Kopfer, H., Laporte, G., Ropke, S., Zaccour, G., 2008. Horizontal cooperation among freight carriers: request allocation and profit sharing. Journal of the Operational Research Society 59 (11), 1483–1491.

Krumke, S. O., de Paepe, W. E., Poensgen, D., Lipmann, M., Marchetti-Spaccamela, A., Stougie, L., 2006. On minimizing the maximum flow time in the online dial-a-ride problem. Lecture Notes in Computer Science 3879, 258–259.

Lehuédé, F., Masson, R., Parragh, S. N., Péton, O., Tricoire, F., 2013. A multi-criteria large neighbourhood search for the transportation of disabled people. Journal of the Operational Research Society 65 (7), 983–1000.

Leyman, P., De Causmaecker, P., 2017. Termination criteria for metaheuristics: Is computation time worth the time? In: Booklet of abstracts of the 31st annual conference of the Belgian Operational Research Society, Brussels. pp. 89–90.

Li, B., Krushinsky, D., Van Woensel, T., Reijers, H. A., 2016. An adaptive large neighborhood search heuristic for the share-a-ride problem. Computers & Operations Research 66, 170–180.

Lin, S., Kernighan, B. W., 1973. An effective heuristic algorithm for the traveling salesman problem. Operations Research 21 (2), 498–516.

Lipmann, M., Lu, X., de Paepe, W. E., Sitters, R. A., Stougie, L., 2004. On-line dial-a-ride problems under a restricted information model. Algorithmica 40 (4), 319–329.

Liu, M., Luo, Z., Lim, A., 2015. A branch-and-cut algorithm for a realistic dial-a-ride problem. Transportation Research Part B: Methodological 81 (1), 267–288.

Lois, A., Ziliaskopoulos, A., Aifantopoulou, G., 2007. A very large scale neighborhood heuristic algorithm for the multivehicle dial a ride with time windows. In: Proceedings of 87th Transportation Research Board Annual Meeting, Washington DC, USA.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives 3, 43–58.

Luo, Y., Schonfeld, P., 2007. A rejected-reinsertion heuristic for the static dial-a-ride problem. Transportation Research Part B: Methodological 41 (7), 736–755.

Luo, Y., Schonfeld, P., 2011. Online rejected-reinsertion heuristics for dynamic multivehicle dial-a-ride problem. Transportation Research Record: Journal of the Transportation Research Board 2218 (1), 59–67.

Maalouf, M., MacKenzie, C. A., Radakrishnan, S., Court, M., 2014. A new fuzzy logic approach to capacitated dynamic dial-a-ride problem. Fuzzy Sets and Systems 255, 30–40.

Masmoudi, M. A., Braekers, K., Masmoudi, M., Dammak, A., 2017. A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. Computers & Operations Research 81, 1–13.

Masmoudi, M. A., Hosny, M., Braekers, K., Dammak, A., 2016. Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. Transportation Research Part E: Logistics and Transportation Review 96, 60–80.

Masson, R., Lehuédé, F., Péton, O., 2012. Simple temporal problems in route scheduling for the dial-a-ride problem with transfers. Lecture Notes in Computer Science 7298, 275–291.

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

Mauri, G., Lorena, L., 2006. A multiobjective model and simulated annealing approach for a dial-a-ride problem. In: Workshop dos Cursos de Computação.

Meer Mobiel, 2016. Aangepast vervoer.
Retrieved from `meermobiel.be/info-aangepast-vervoer`.

Melachrinoudis, E., Ilhan, A. B., Min, H., 2007. A dial-a-ride problem for client transportation in a health-care organization. Computers & Operations Research 34 (3), 742–759.

Minder Mobielen Centrale, 2016. Wat is een minder mobielen centrale? (in Dutch)
Retrieved from `mindermobielencentrale.be/nl/node/16`.

Molenbruch, Y., Braekers, K., Caris, A., 2017a. Operational effects of service level variations for the dial-a-ride problem. Central European Journal of Operations Research 25 (1), 71–90.

Molenbruch, Y., Braekers, K., Caris, A., 2017b. Typology and literature review for dial-a-ride problems. Annals of Operations Research (forthcoming).

Molenbruch, Y., Braekers, K., Caris, A., Vanden Berghe, G., 2017c. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. Computers & Operations Research 77, 58–71.

Muelas, S., LaTorre, A., Peña, J.-M., 2013. A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city. Expert Systems with Applications 40 (14), 5516–5531.

Muelas, S., LaTorre, A., Peña, J.-M., 2015. A distributed vns algorithm for optimizing dial-a-ride problems in large-scale scenarios. Transportation Research Part C: Emerging Technologies 54, 110–130.

Nowak, M., Ergun, Ö., White, C. C., 2008. Pickup and delivery with split loads. Transportation Science 42 (1), 32–43.

Oberscheider, M., Hirsch, P., 2016. Analysis of the impact of different service levels on the workload of an ambulance service provider. BMC Health Services Research 16 (1), 487.

Paquette, J., Bellavance, F., Cordeau, J.-F., Laporte, G., 2012. Measuring quality of service in dial-a-ride operations: the case of a Canadian city. Transportation 39 (3), 539–564.

Paquette, J., Cordeau, J.-F., Laporte, G., 2009. Quality of service in dial-a-ride operations. Computers & Industrial Engineering 56 (4), 1721–1734.

Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M. M. B., 2013. Combining multicriteria analysis and tabu search for dial-a-ride problems. Transportation Research Part B: Methodological 52, 1–16.

Parragh, S. N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transportation Research Part C: Emerging Technologies 19 (5), 912–930.

Parragh, S. N., Cordeau, J.-F., Doerner, K. F., Hartl, R. F., 2012. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. OR Spectrum 34 (3), 593633.

Parragh, S. N., Doerner, K. F., Hartl, R. F., 2008. A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. Journal für Betriebswirtschaft 58 (2), 81–117.

Parragh, S. N., Doerner, K. F., Hartl, R. F., 2010a. Demand responsive transportation. Wiley Encyclopedia of Operations Research and Management Science, 1–9.

Parragh, S. N., Doerner, K. F., Hartl, R. F., 2010b. Variable neighborhood search for the dial-a-ride problem. Computers & Operations Research 37 (6), 1129–1138.

Parragh, S. N., Doerner, K. F., Hartl, R. F., Gandibleux, X., 2009. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. Networks 54 (4), 227–242.

Parragh, S. N., Pinho de Sousa, J., Almada-Lobo, B., 2014. The dial-a-ride problem with split requests and profits. Transportation Science 49 (2), 311–334.

Parragh, S. N., Schmid, V., 2013. Hybrid column generation and large neighborhood search for the dial-a-ride problem. Computers & Operations Research 40 (1), 490–497.

Pisinger, D., Røpke, S., 2010. Large neighborhood search. In: Handbook of metaheuristics. Springer, pp. 399–419.

Potvin, J.-Y., Rousseau, J.-M., 1995. An exchange heuristic for routing problems with time windows. Journal of the Operational Research Society 46 (12), 1433–1446.

Psaraftis, H. N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transportation Science 14 (2), 130–154.

Psaraftis, H. N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transportation Science 17 (3), 351–357.

Qu, Y., Bard, J. F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. Transportation Research Part C: Emerging Technologies 32, 1–20.

Qu, Y., Bard, J. F., 2014. A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. Transportation Science 49 (2), 254–270.

Quadrifoglio, L., Dessouky, M. M., Ordóñez, F., 2008. A simulation study of demand responsive transit system design. Transportation Research Part A: Policy and Practice 42 (4), 718–737.

Rekiek, B., Delchambre, A., Saleh, H. A., 2006. Handicapped person transportation: An application of the grouping genetic algorithm. Engineering Applications of Artificial Intelligence 19 (5), 511–520.

Ritzinger, U., Puchinger, J., Hartl, R. F., 2016. Dynamic programming based metaheuristics for the dial-a-ride problem. Annals of Operations Research 236 (2), 341–358.

Røpke, S., Cordeau, J.-F., 2009. Branch-and-cut-and-price for the pickup and delivery problem with time windows. Transportation Science 43 (3), 267–286.

Røpke, S., Cordeau, J.-F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 49 (4), 258–272.

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

Rubinstein, Z. B., Smith, S. F., Barbulescu, L., 2012. Incremental management of oversubscribed vehicle schedules in dynamic dial-a-ride problems. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. pp. 1809–1815.

Santos, D. O., Xavier, E. C., 2015. Taxi and ride sharing: a dynamic dial-a-ride problem with money as an incentive. Expert Systems with Applications 42 (19), 6728–6737.

Savelsbergh, M. W., 1992. The vehicle routing problem with time windows: Minimizing route duration. ORSA Journal on Computing 4 (2), 146–154.

Schilde, M., Doerner, K. F., Hartl, R. F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & Operations Research 38 (12), 1719–1730.

Schilde, M., Doerner, K. F., Hartl, R. F., 2014. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. European Journal of Operational Research 238 (1), 18–30.

Schneider, B., White, S. S., 2004. Service quality: Research perspectives. Vol. 107. Sage.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record breaking optimization results using the ruin and recreate principle. Journal of Computational Physics 159 (2), 139–171.

Shaw, P., 1997. A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. Lecture Notes in Computer Science 1520, 417–431.

Taillard, É., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23 (8), 661–673.

Tang, J., Kong, Y., Lau, H., Ip, A. W. H., 2010. A note on "Efficient feasibility testing for dial-a-ride problems". Operations Research Letters 38 (5), 405–407.

Teodorovic, D., Radivojevic, G., 2000. A fuzzy logic approach to dynamic dial-a-ride problem. Fuzzy Sets and Systems 116 (1), 23–33.

Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. Transportation Science 31 (1), 60–71.

Tricoire, F., 2012. Multi-directional local search. Computers & Operations Research 39 (12), 3089–3101.

United Nations, 2006. Convention on the rights of persons with disabilities and optional protocol. Retrieved from `www.un.org/disabilities/documents/convention/convoptprot-e.pdf`.

Urra, E., Cubillos, C., Cabrera-Paniagua, D., 2015. A hyperheuristic for the dial-a-ride problem with time windows. Mathematical Problems in Engineering 2015.

van Buggenhout, M., van Rompaey, B., Dierckxsens, S., 2012. De weg naar meer transparantie: wachtlijsten in Vlaamse woonzorgcentra. Ablecare.

Vandeurzen, J., 2015. 8413 nieuwe woongelegenheden in Vlaamse woonzorgcentra.
Retrieved from `www.jovandeurzen.be/nl/8413-nieuwe-woongelegenheden-vlaamse-woonzorgcentra`.

Verdonck, L., Beullens, P., Caris, A., Ramaekers, K., Janssens, G. K., 2016. Analysis of collaborative savings and cost allocation techniques for the cooperative carrier facility location problem. Journal of the Operational Research Society 67 (6), 853–871.

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

Verhetsel, A., Van Hecke, E., Thomas, I., Beelen, M., Halleux, J. M., Lambotte, J. M., Rixhon, G., Mérenne-Schoumaker, B., 2009. Sociaal-economische enquête 2001: monografieën - pendel in België (in Dutch). FOD Economie, K.M.O., Middenstand en Energie.

Vlaams Parlement, 2001. Decreet betreffende de organisatie van het personenvervoer over de weg en tot oprichting van de Mobiliteitsraad van Vlaanderen (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Vlaams Parlement, 2004. Decreet betreffende de Vlaamse Vervoermaatschappij - De Lijn, publiekrechtelijk vormgegeven extern verzelfstandigd agentschap, en houdende wijziging van het decreet van 31 juli 1990 tot oprichting van de Vlaamse Vervoermaatschappij (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Vlaams Parlement, 2008. Decreet houdende een kader voor het Vlaamse gelijkekansen- en gelijkebehandelingsbeleid (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Vlaams Parlement, 2009. Woonzorgdecreet (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Vlaams Parlement, 2013. Decreet tot compensatie van de openbaredienstverplichting tot het vervoer van personen met een handicap of een ernstig beperkte mobiliteit (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2009. Studieopdracht: Internationale benchmarkstudie openbaar vervoer (in Dutch).
Retrieved from `www.mobielvlaanderen.be/studies/ov-benchmarkstudie.php?a=18`.

Vlaamse Overheid, Departement Mobiliteit en Openbare Werken, 2013. Businessplan voor een gebiedsdekkend, complementair en gentegreerd Vlaams toegankelijk vervoersysteem in Vlaanderen en het Brussels Hoofdstedelijk Gewest (in Dutch).
Retrieved from `www.mobielvlaanderen.be/studies/toegankelijkvervoer.php?a=18`.

Vlaamse Regering, 2015. Conceptnota: met basisbereikbaarheid naar een efficiënt en aantrekkelijk vervoersmodel in Vlaanderen dat optimaal tegemoetkomt aan de globale en lokale vervoersvraag (in Dutch).
Retrieved from `codex.vlaanderen.be`.

Wolfler Calvo, R., Touati-Moungla, N., 2011. A matheuristic for the dial-a-ride problem. Lecture Notes in Computer Science 6101, 450–463.

Wong, K. I., Bell, M. G. H., 2006. Solution of the dial-a-ride problem with multi-dimensional capacity constraints. International Transactions in Operational Research 13 (3), 195–208.

Xiang, Z., Chu, C., Chen, H., 2006. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. European Journal of Operational Research 174 (2), 1117–1139.

Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. European Journal of Operational Research 185 (2), 534–551.

Yi, F., Tian, L., 2005. On the online dial-a-ride problem with time-windows. Lecture Notes in Computer Science 3521, 85–94.

Yi, F., Xu, Y., Xin, C., 2006. Online dial-a-ride problem with time-windows under a restricted information model. Lecture Notes in Computer Science 4041, 22–31.

Zhang, Z., Liu, M., Lim, A., 2015. A memetic algorithm for the patient transportation problem. Omega 54, 60–71.

Ziliaskopoulos, A., Kozanidis, G., 2006. Dynamic programming strategies for the dial a ride problem with time window constraints. In: TRB 2006: 85th Annual Meeting of the Transportation Research Board, Washington D.C.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3 (4), 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Grunert da Fonseca, V., 2003. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7 (2), 117–132.

Nederlandstalige samenvatting

Dial-a-ride systemen richten zich traditioneel op vraagafhankelijk, collectief vervoer van personen met mobiliteitsbeperkingen, zoals ouderen en mindervaliden. Omwille van demografische evoluties (bv. de vergrijzing) en ontwikkelingen in de gezondheidszorg (bv. de opkomst van dagopvangcentra) is een dergelijke dienstverlening op maat onontbeerlijk geworden in onze hedendaagse samenleving. Daarenboven incorporeren innovatieve mobiliteitsvisies, waaronder het concept *Basisbereikbaarheid* van de Vlaamse Regering, een vraagafhankelijke dienstverlening door een private aanbieder als kwalitatief en kostenefficiënt alternatief voor regulier openbaar vervoer in landelijk gebied. Hoofdstuk 1 verklaart waarom de aanbieders van dial-a-ride diensten, gegeven deze maatschappelijke en beleidsmatige context, steeds meer nood hebben aan rittenplanningsalgoritmes ter ondersteuning van hun operationele activiteiten. Deze doctoraatsthesis onderzoekt hoe aanbieders het kwaliteitsniveau en de operationele kosten van hun dienstverlening kunnen balanceren met het oog op de implementatie van een strategisch kwaliteitsbeleid. Hiertoe wordt een efficiënt rittenplanningsalgoritme ontwikkeld dat de aanbieders inzicht verschaft in bovenvermelde afweging, hetgeen hen vervolgens in staat stelt om een coherente kwaliteitsvisie te ontwikkelen en operationele keuzes hierin te kaderen. Bijzondere aandacht gaat uit naar de invloed van diverse operationele eigenschappen van het vervoerssysteem.

In hoofdstuk 2 wordt een overzicht geschetst van reeds bestaand academisch onderzoek naar rittenplanningsalgoritmes voor het dial-a-ride probleem. Tijdens het afgelopen decennium werd steeds meer aandacht besteed aan de incorporatie van geavanceerde operationele kenmerken, gebaseerd op realistische systemen. Hierbij werd o.a. rekening gehouden met het vervoer van heterogene klanten (bv. rolstoelgebruikers en mobiele personen), complexere routeringseigenschappen (bv. inlassen van overstaps), alsook het verwerken van onzekere of laattijdig beschikbare info (bv. vervoersaanvragen in real time). Aangezien exacte oplossingstechnieken, die de optimale rittenplanning garanderen te vinden, doorgaans een onrealistisch hoge rekentijd laten noteren, wordt het onderzoek voornamelijk toegespitst op de ontwikkeling van snelle benaderende (metaheuristische) rittenplanningsalgoritmes.

Bestaande algoritmes construeren doorgaans één rittenplanning die de totale afgelegde afstand van de voertuigenvloot minimaliseert. Het kwaliteitsniveau wordt desgevallend enkel gegarandeerd door middel van minimumvereisten. Enerzijds mag het tijdstip waarop een klant wordt opgehaald of afgeleverd slechts binnen opgelegde grenzen afwijken van zijn voorkeurtijdstip. Anderzijds mag zijn reistijd een vooraf bepaalde maximumduur niet overschrijden. Naarmate strengere kwaliteitsvereisten worden opgelegd, vermindert het aantal mogelijkheden om meerdere klanten in eenzelfde rit te combineren. Een sensitiviteitsanalyse in hoofdstuk 3 illustreert dat de opgelegde kwaliteitsvereisten daarom een aanzienlijke impact hebben op de werkingskosten van de aanbieder. Diverse operationele kenmerken (bv. de grootte van de aanbieder, heterogeneit van de gebruikers en verstedelijking van het werkingsgebied) beïnvloeden de precieze grootte van dit effect.

Bovenstaande werkwijze streeft geen optimalisatie van het kwaliteitsniveau na en veronderstelt bovendien dat een aanbieder de minimale kwaliteitsvereisten vooraf vastlegt, zonder de implicaties op de werkingskosten te kunnen inschatten. In hoofdstuk 4 wordt daarentegen een metaheuristisch rittenplanningsalgoritme ontworpen dat rekening houdt met de ware toedracht van een dial-a-ride probleem. Het algoritme genereert een set van oplossingen die elkaar niet domineren met betrekking tot twee doelstellingen: een minimalisatie van de totale afgelegde afstand van de dial-a-ride voertuigen (operationele doelfunctie) en een minimalisatie van de totale reistijd van de klanten (kwaliteitsgerelateerde doelfunctie). Deze werkwijze verschaft inzicht in de afweging tussen de werkingskosten en de kwaliteit, hetgeen aanbieders toelaat een onderbouwd kwaliteitsbeleid te implementeren. Uit experimenten blijkt dat de oplossingenset een convex patroon vertoont. Een lichte afwijking van de minimale operationele kost kan bijgevolg in een aanzienlijke kwaliteitsverbetering resulteren. Het algoritme wordt ten slotte toegepast op een probleemvariant met combinatiebeperkingen, geïnspireerd op een realistische case in het ziekenvervoer. Om medische redenen mogen bepaalde klanten hierbij niet gecombineerd vervoerd worden of dienen zij te worden toegewezen aan een bestuurder met een bijzonder kwalificatieniveau. Deze beperkingen oefenen een significante invloed uit op de situering van de oplossingenset.

Om de kwaliteitsgerelateerde doelfunctie van het voorgestelde algoritme correct te beoordelen, dient de vertrektijd op iedere locatie in een route zodanig bepaald te worden dat de totale reistijd van de betrokken klanten minimaal is. In de academische literatuur is geen heuristische procedure bekend die garandeert hierin te slagen. Voor een bepaald percentage van de routes wordt onterecht vastgesteld dat geen geldig tijdsschema bestaat of wordt een tijdsschema opgesteld waarin de totale reistijd afwijkt van de optimale waarde. In vergelijking met bestaand werk introduceert hoofdstuk 5 een procedure die het risico op dergelijke falingen sterk reduceert en daarenboven minder rekentijd vergt. Een variant op deze procedure vermijdt dat het ophalen van beschikbare klanten bewust kan

worden uitgesteld (i.c. dat overmatige reistijd wordt omgezet in onnodige wachttijd) indien andere klanten daardoor langer onderweg zouden zijn.

Ook andere operationele strategieën van aanbieders kunnen de afweging tussen werkingskosten en kwaliteit beïnvloeden. Indien aanbieders eenzelfde kwaliteitsvisie delen en in dezelfde regio actief zijn, kunnen hun werkingskosten gereduceerd worden door middel van een gezamenlijke rittenplanning. Dit impliceert dat vervoersaanvragen van klanten via een gecentraliseerde beslissingsstrategie worden uitgewisseld tussen de aanbieders. In hoofdstuk 6 wordt een metaheuristisch algoritme ontworpen waarin vervoersaanvragen zodanig worden toegewezen dat de gezamenlijk afgelegde afstand van alle dial-a-ride voertuigen geminimaliseerd wordt. Zodoende kan het aandeel ledige ritten gereduceerd worden, terwijl het kwaliteitsniveau van de dienstverlening onaangetast blijft. De precieze grootte van de hieruit voortvloeiende besparing wordt sterk beïnvloed door operationele kenmerken (bv. het aantal aanvragen, de locatie van de depots en de ruimtelijke clustering van de aanvragen). Daarenboven kan een gecentraliseerde beslissingsstrategie enkel standhouden indien alle betrokken aanbieders het eens raken over een verdeeltechniek om de verkregen besparing onderling te alloceren.

Op basis van de conclusies in de voorgaande hoofdstukken formuleert hoofdstuk 7 concrete aanbevelingen aan drie stakeholders die een rechtstreekse invloed uitoefenen op de balans tussen kosten en kwaliteit in dial-a-ride systemen, zijnde onderzoekers, aanbieders en beleidsmakers. Hun beslissingen bepalen in welke mate de kwaliteitsverwachtingen van de klant, de vierde belangrijke stakeholder in de context van deze thesis, worden ingelost. Wetenschappelijk onderzoek kan een contributie leveren via een ruimere focus op de operationele impact van realistische probleemkenmerken, alsook een verdere efficiëntieverbetering van de bestaande rittenplanningsalgoritmes. Diverse suggesties worden hiertoe aangereikt. De aanbieders zijn vooral gebaat bij de aanschaf van rittenplanningssoftware die hen voldoende informatie verschaft om een strategische kwaliteitsvisie op het operationele niveau te implementeren. Daarnaast wordt de aandacht gevestigd op enkele innovaties die hun competitiviteit kunnen verhogen. Ten slotte kunnen beleidsmakers de efficiëntie van dial-a-ride systemen faciliteren middels een volwaardige integratie in het regulier openbaar vervoer. Tegelijk dienen zij te waken over een kwaliteitsgericht compensatiesysteem voor het vervoer van personen met beperkte mobiliteit.

Publications and conference participations

Accepted journal publications

Molenbruch, Y., Braekers, K., Caris, A., Vanden Berghe, G., 2017. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. Computers & Operations Research, 77, 58-71.

Molenbruch, Y., Braekers, K., Caris, A., 2017. Operational effects of service level variations for the dial-a-ride problem. Central European Journal of Operations Research, 25(1), 71-90.

Molenbruch, Y., Braekers, K., Caris, A., 2017. Typology and literature review for dial-a-ride problems. Annals of Operations Research (forthcoming).

Submitted journal publications

Molenbruch, Y., Braekers, K., Caris, A., 2017. Benefits of horizontal cooperation in dial-a-ride services. Submitted to Transportation Research Part E.

Publications in conference proceedings

Molenbruch, Y., Braekers, K., Caris, A., 2015. A multi-directional local search metaheuristic for a bi-objective dial-a-ride problem. In: Brotcorne, L., Feillet, D., Jozefowiez, N., Quadri, D., Semet, F., Mouclier, C. (Ed.). 6th International Workshop on Freight Transportation and Logistics (Odysseus), 2015, 499-504.

Other conference participations

Molenbruch, Y., Braekers, K., Caris, A., 2014. Operational effects of variations in service level criteria for the dial-a-ride problem. In: 28th Annual Conference of the Belgian Operations Research Society, Mons, Belgium, 30-31 January, 2014.

Molenbruch, Y., Braekers, K., Caris, A., 2014. Operational effects of service level variations for the dial-a-ride problem. In: VeRoLog 2014, Oslo, Norway, 22-25 June, 2014.

Molenbruch, Y., Braekers, K., Caris, A., 2015. Solving a bi-objective dial-a-ride problem using multi-directional local search. In: 29th Annual Conference of the Belgian Operations Research Society, Antwerp, Belgium, 5-6 February, 2015.

Molenbruch, Y., Braekers, K., Caris, A., 2015. Solving a bi-objective dial-a-ride problem using multi-directional local search and an exact scheduling procedure. In: VeRoLog 2015, Vienna, Austria, 8-10 June, 2015.

Molenbruch, Y., Braekers, K., Caris, A., 2016. Horizontal cooperation in dial-a-ride services. In: 30th Annual Conference of the Belgian Operations Research Society, Louvain-la-Neuve, Belgium, 28-29 January, 2016.

Molenbruch, Y., Braekers, K., Caris, A., 2016. Horizontal Cooperation in Dial-a-Ride Services. In: VeRoLog 2016, Nantes, France, 6-8 June, 2016.

Molenbruch, Y., Braekers, K., Caris, A., 2016. Horizontal Cooperation in Dial-a-Ride Services. In: 28th European Conference on Operational Research, Poznan, Poland, 4-6 July, 2016.