Optimization method to integrate driver consistency and route balancing in a heterogeneous multi-period dial-a-ride problem

Auteur : Hubert, Margot
Promoteur(s) : Paquay, Célia
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en ingénieur de gestion, à finalité spécialisée en Supply Chain Management and Business Analytics
Année académique : 2022-2023
URI/URL : http://hdl.handle.net/2268.2/17204

OPTIMIZATION METHOD TO INTEGRATE DRIVER CONSISTENCY AND ROUTE BALANCING IN A HETEROGENEOUS MULTI-PERIOD DIAL-A-RIDE PROBLEM

Jury:
Supervisor:
Célia PAQUAY
Reader:
Marie BARATTO

Master thesis by
Margot HUBERT
For a Master in Business Engineering with a specialization in Supply Chain Management and Business Analytics
Academic Year 2022/2023

Acknowledgements

This thesis represents the culmination of a five-year educational journey which brought me increasingly enriching experiences. Furthermore, it enables me to obtain a Master's degree in Business Engineering, focusing on Supply Chain Management and Business Analytics.

I would like to express my particular gratitude to Ms Célia Paquay, my supervisor, for her continuous support, not only during the writing of this work but also throughout the year, for openly discussing my ideas and motivating me to push my limits. Her guidance was invaluable, and her supportive demeanour was a constant motivation that allowed me to produce this thesis under the best conditions.

Additionally, I would like to thank Ms Marie Baratto for consistently agreeing to address my questions, offering assistance whenever possible, and for her ongoing availability and adaptability.

I also extend a warm thanks to Mr Maxime André, a PhD student in the Faculty of Computer Science at the University of Namur, Mr Noé Schwaiger and Ms Madison Robert for proofreading this thesis before its submission. Their review was precious in ensuring that the work was of the highest possible quality.

Lastly, I would like to thank my family and loved ones for their unwavering support throughout my years of study and the development of this final achievement. I could not have produced this work at the level that I did without their encouragement.

Contents

List of abbreviations

ACO Ant Colony Optimization. 24

ConVRP Consistent Vehicle Routing Problem. 14, 21

CVRP Capacitated Vehicle Routing Problem. 13

DA Deterministic Annealing. 24, 26, 33

DARP Dial-a-Ride problem. 7–26, 33, 35, 44, 49, 67, 69, 70, 90

DC-DARP Driver Consistent Dial-a-Ride Problem. 10, 11

DCI Driver Consistency Indicator. 52–57, 59–62, 64, 68

DVRP Dynamic Vehicle Routing Problem. 14

E-ADARP Electric Autonomous Dial-a-Ride Problem. 17

FRDARP Fixed Route Dial-a-Ride Problem. 16

GA Genetic Algorithm. 24, 26, 33

GRASP Greedy Randomized Adaptive Search Procedure. 24, 26

GVRP Green Vehicle Routing Problem. 14

LNS Large Neighborhood Search. 24, 26, 33

MA Memetic Algorithm. 24, 26

MO-DARP Multi-Objective Dial-a-Ride Problem. 17, 18, 31

MP-DARP Multi-period Dial-a-Ride problem. 9, 13, 19, 25, 27, 67, 70, 90

PDVRP Pick-up and Delivery Vehicle Routing Problem. 9, 14

PL Path Relinking. 24, 26

RBI Route Balancing Indicator. 52–57, 59–64, 68

SA Simulated Annealing. 11, 12, 24–26, 33–35, 42, 43, 45, 51–53, 55, 56, 58, 59, 67, 68, 90

SDG Sustainable Development Goal. 7

SDVRP Split Delivery Vehicle Routing Problem. 14

TOD Transportation On Demand. 9, 14

TS Tabu Search. 24, 26, 33

VNS Variable Neighborhood Search. 24, 26, 33

VRP Vehicle Routing Problem. 9, 11–15, 19–23, 25, 37, 67

VRPRB Vehicle Routing Problem with Route Balancing. 22

VRPTW Vehicle Routing Problem with Time Windows. 9, 14

1 Introduction

The prevailing ecological issues confronting the globe, along with the growing traffic congestion and the exponential surge in fuel costs, as well as accessibility issues faced by people with limited mobility, have led to a mounting inclination towards shared transportation systems. Furthermore, contemporary advancements in technology are facilitating new transportation options and serve as a springboard towards a more eco-conscious future. Indeed, advanced computational technologies combined with mathematical models are exploited to optimize mobility plans. As stated in Mourad et al. (2019), there are various modes of shared mobility available, with some focusing on package delivery, others on transporting individuals, and some providing a combination of both. This analysis will specifically examine the Dial-a-Ride problem (DARP) within the realm of shared mobility for people transportation, which aims to develop the most efficient vehicle routes and schedules based on user-provided pick-up and delivery requests, as described in the earlier study by Cordeau and Laporte (2003).

Nevertheless, due to the continuous growth of the global population and increasingly intricate demands, the need to satisfy various requirements such as timing and comfort while also remaining cost-effective often takes precedence. The well-established pair of conflicting objectives remains ever-present: how to guarantee the fulfilment of as many needs as possible, while maintaining an adequate level of service quality? Despite public transportation serving a substantial number of passengers, it is not accessible in rural areas and installing it would be financially unfeasible.

Moreover, the population of individuals who are unwell, elderly, or experiencing restricted mobility is rising (Kovacs et al. (2014)), and such individuals may not always possess the capability to utilize these public transportation modes. Combined with the increased availability of home care services (Paquette et al. (2012)), there is a strong need for customized on-demand transportation services, even if they come at a high cost and have negative environmental impacts. There is thus a search for solutions that provide door-to-door transportation while prioritizing environmental sustainability at a reasonable expense (Ho et al. (2018)). All of these objectives are aligned with the 11th Sustainable Development Goal (SDG) established by the United Nations, which aims to *"make cities and human settlements inclusive, safe, resilient and sustainable"* (Nasri et al. (2022), p.1).

The primary objective of this thesis is to centre on people transportation. The tension between maximizing profitability and ensuring people's comfort is investigated. This analysis considers not only the comfort of patients but also the fairness in workload distribution among drivers, also called route balancing (Feng and Wei (2022)). It is important to ensure that the patients' comfort is not prioritized at the expense of inadequate workload distribution among drivers. Therefore, the study addresses the need to strike a balance between these factors to promote both passenger satisfaction and driver well-being.

The introduction of this thesis begins with a general overview of the DARP, followed by a detailed discussion of the specific elements that have been added to the problem formulation for the purpose of this study.

1.1 General problem description

Based on the existing literature (Cordeau and Laporte (2003); Cordeau and Laporte (2007); Cordeau et al. (2007); Molenbruch et al. (2017b)), the traditional DARP involves creating routes and schedules for a fleet of vehicles in order to satisfy patients'[1] requests over a single day. Typically, the patients are elderly or disabled individuals who require transportation for medical appointments, with outbound requests to reach their appointments and inbound requests to return home. The fleet of vehicles can either be homogeneous, with identical capacities and characteristics, or heterogeneous, with different types of vehicles. The vehicles' routes can either originate and end at a single depot or be split across multiple depots. Moreover, the vehicles are generally assumed to travel at a constant speed throughout the whole day. This shared transportation service enables multiple patients with different transportation requests to be accommodated in the same vehicle simultaneously. In most cases, patients are given the opportunity to specify their desired pick-up and delivery locations and time windows[2], indicate the number of accompanying individuals if there are any, and indicate the potential use of a wheelchair or stretcher. Some patients may also have a maximum travel time they can tolerate, such as those undergoing dialysis (Yazawa et al. (2020)). Drivers may also specify a maximum duration for their route according to their working schedules. Figure 1.1 depicts a basic instance of a DARP.

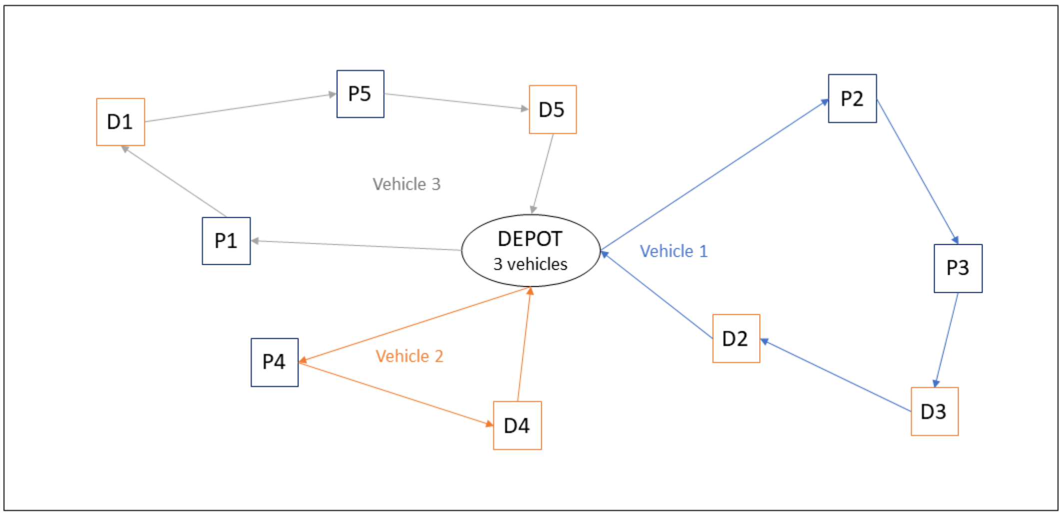

Figure 1.1: Example of DARP with 3 vehicles. The letter "P" stands for pick-up locations, where patients begin their travel, while "D" denotes delivery locations, where patients want to arrive. The central location represents the depot from which 3 vehicles leaves at the beginning of the working day. Each of them visits various locations to fulfil patients' requests. The numerical value indicates the patient associated with each location.

[1]Since the study focuses on the transportation of individuals, the term "customer" will be consistently replaced by "patient" throughout the remainder of the research.

[2]Earliest and latest times between which a patient wants to be loaded or unloaded at a specific location (Ho et al. (2018)).

The objectives of this transportation service can be categorized as either operational-oriented or user-oriented. In the former case, the primary objective is usually to minimize costs while satisfying all demands and side constraints or to maximize demand satisfaction subject to vehicle availability and side constraints. In the latter case, the aim may be to minimize the total ride time or the overall waiting time of patients, which could result in increased operating costs such as the requirement for additional vehicles. Indeed, these two objectives are always in conflict and cannot be simultaneously optimized. Thus, a conventional objective function can only prioritize one of these objectives, or both with a specific weighting factor (Ho et al. (2018)).

It should be noted that such DARPs have already been implemented in many major cities around the world, including Berlin, Hong Kong, London, New York, Paris, Stockholm, and many others (Paquette et al. (2009)). In Belgium, comparable transportation services are already in place through organizations such as the Red Cross, Public Social Welfare Centres, non-profit organizations, as well as some healthcare insurance companies.

The DARP is an extension of existing Vehicle Routing Problem (VRP)s, including the Pick-up and Delivery Vehicle Routing Problem (PDVRP) and the Vehicle Routing Problem with Time Windows (VRPTW) (Cordeau and Laporte (2003)), and is also included in the Transportation On Demand (TOD) problems' classification (Paquette et al. (2009)). Due to its NP-hard complexity (Molenbruch et al. (2017b)), it cannot be solved in polynomial time with exact methods. Thus, heuristics[3] are commonly used to provide solutions for problems of realistic sizes. This allows for the generation of high-quality solutions within reasonable time frames.

1.2 Choice of specifications to integrate into my thesis

Chapter 2 of this thesis will provide a more detailed analysis of the literature that has already been conducted in this area, primarily focusing on the classic DARP. However, the problem specifications that will be addressed in this study involve challenges that have not been deeply explored in the existing literature. The proposed approach in this thesis seeks to **integrate driver consistency into a Multi-period Dial-a-Ride problem (MP-DARP), with a significant emphasis on promoting fairness among drivers**[4]. These aspects have not been jointly considered before in previous problem definitions.

The DARP is a specific type of VRP that places a strong emphasis on the human aspect. This means that the well-being and comfort of patients are considered to be of utmost importance in this type of mobility service. The intensifying competition in the mobility services sector and the emergence of more demanding customers have made it imperative for companies to differentiate themselves based on the service quality they provide. Relying solely on competitive pricing is no longer sufficient. Furthermore, maintaining favourable working conditions for employees not only promotes high productivity but also leads to enhanced profitability (Kovacs et al. (2014)). Figure 1.2 is a possible representation of this service-profit chain.

[3] A heuristic is an algorithm allowing one to solve complex optimization problems without requiring formal modelling. The obtained solutions may be sub-optimal (Ho et al. (2018)).

[4] Fairness among drivers can be defined in various ways, which will be detailed in section 2.4

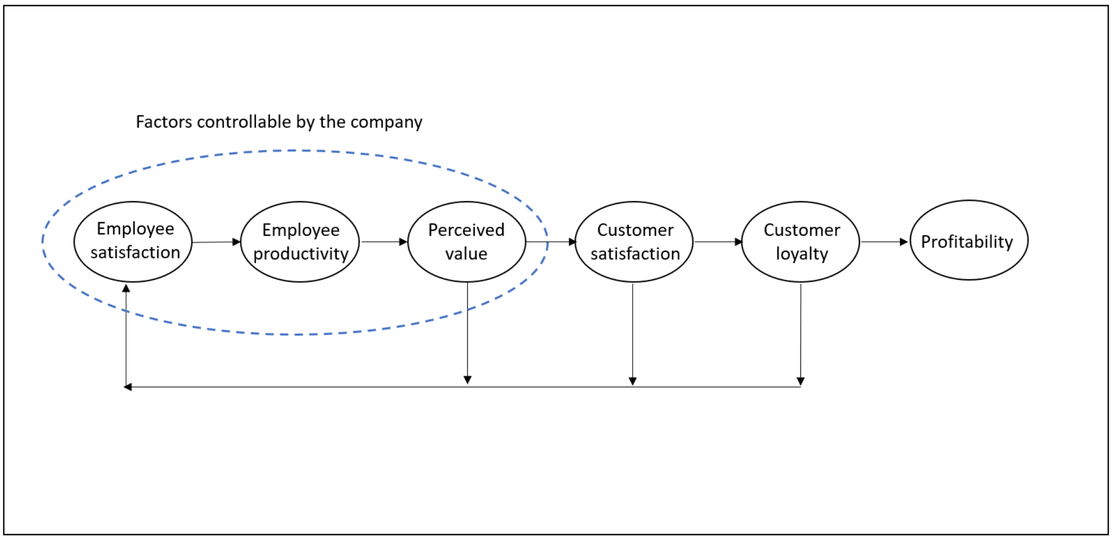

Figure 1.2: Service-profit chain: relationships between profitability, customer and employee satisfaction and productivity (reproduced from Kovacs et al. (2014)).

1.2.1 Integration of driver consistency

Studies and research conducted by Paquette et al. (2009) and Paquette et al. (2012) have shown that patients often associate the quality of their service with having a dedicated driver for each appointment, rather than seeing different drivers each time. This concept is referred to as **driver consistency** (Braekers and Kovacs (2016)), and it will be a key component of the problem formulation under study. The problem being studied involves analyzing the correlation between each patient and the overall number of drivers required to fulfill all their requests during a five-day period in a multi-period Driver Consistent Dial-a-Ride Problem (DC-DARP) (Braekers and Kovacs (2016)). Indeed, most studies conducted on DARP currently only focus on a single day. However, in this scenario, it is imperative to extend the analysis over multiple days, which will enable the examination of the relationships between patients and drivers over a more extended period.

In addition, it should be mentioned that driver consistency is advantageous for both patients and drivers. Patients can establish stronger relationships with their drivers and benefit from customized services, while drivers can achieve greater productivity by frequently visiting the same areas to serve the same patients. Moreover, it enhances driver satisfaction by eliminating the need for irregular routing plans (Kovacs et al. (2014)).

1.2.2 Integration of route balancing

In this thesis, the exploration of fairness among drivers, also called **route balancing**, will be undertaken. Given that the problem being studied spans multiple days, it is crucial that all drivers operate their vehicles in a fair and equitable manner. Fairness in this context can be defined in various ways:

- One approach involves ensuring that each driver has an equal number of working days over the entire week;
- Alternatively, fairness can be measured in terms of the total travel time for each driver, with

the requirement that all drivers drive the same number of hours over the course of the week (without regard to the distribution of those hours across days);

- Finally, another way to approach fairness is to assess the number of patient requests each driver fulfills throughout the week.

Having considered these different approaches, this thesis will concentrate on evaluating fairness among drivers by analyzing their **total travel time over an entire week**. This method provides a highly accurate and reliable measure that significantly reduces the likelihood of any unfair treatment among drivers. To achieve this, the total travel time for each driver will be treated as a variable, and any deviation from the ideal scenario in which all drivers have the same amount of work time will be minimized in the objective function.

1.2.3 Other specifications of the problem under study

The research problem will involve a **multi-period DC-DARP with a single depot**, where the fleet of vehicles is **heterogeneous** since each vehicle will have a dedicated capacity limit. The vehicle's speed will be assumed to stay constant throughout the day and the week. Patients will specify **time windows** for each request, along with a **maximum ride duration** and the option to **include accompanying individuals**. Furthermore, **patients requiring special assistance will be able to be transported in a wheelchair**.

It is essential to note that the problem formulation will remain **static and deterministic**. This implies respectively that all decisions will be made beforehand, without any modifications of the planning during the week, and that all information will be fully known at the time of decision-making, with no random or uncertain elements involved (Ho et al. (2018)).

The objective function of the problem aims to optimize driver consistency as the primary objective and route balancing as the secondary objective. The optimization process thus involves finding an improved solution regarding driver consistency, which will be set as an upper bound for the constraints. The aim is then to optimize the route balancing without worsening the best level obtained for driver consistency. This approach, where a high-level objective is optimized first, and a lower-level objective is optimized when possible, is known as a lexicographic objective function.

In the majority of DARP studies, the objective function typically prioritizes the minimization of total distance traveled. While the optimization process under study may not directly focus on minimizing distance, it inherently occurs as patient requests are inserted into routes in a way that minimizes the overall road distance.

The present static and deterministic DARP will be solved through a **Simulated Annealing (SA)**. The justification for this meta-heuristic choice and its features will be thoroughly explained in Section 3 of this thesis.

1.3 Thesis structure

To evaluate the existing literature on the topic, a literature review will be conducted in Chapter 2. This review will cover various aspects of VRPs in general, as well as static and deterministic DARPs. Additionally, it will analyze customer-oriented objectives that have been previously considered in both VRPs and DARPs. It will also focus on route balancing and the extent of work done in this area. Furthermore, the review will summarize the commonly used meta-heuristics for solving DARPs.

Chapter 3 will present a more precise formulation of the problem at stake, with a particular emphasis on the objective function and the metrics used to assess driver consistency and route balancing.

Subsequently, the choice of the SA will be discussed in Chapter 4 along with its constituent elements: the selection of the initial solution method, the choice of the neighbourhoods and the parameters identification.

Chapter 5 will encompass a description of the dataset used, implementation details, and a deep analysis of the results obtained.

Then, Chapter 6 will provide an answer to the research question, which is to evaluate the extent to which driver consistency and route balancing can be incorporated into a multi-period DARP. Finally, it will summarize the findings of the research as well as its limitations, and outline potential avenues for future research on this topic.

2 State-of-the-art

Given that the objective of this investigation is to solve a static and deterministic DARP using a meta-heuristic method, the state-of-the-art analysis will be confined to this specific research domain. Consequently, the review and comparison of existing techniques will be limited to those developed for comparable static and deterministic DARPs. Prior to that, a summary of the existing work related to the broader topic of VRPs will be conducted in Section 2.1.

Furthermore, the forthcoming literature review will be organized to clarify the previous research conducted on the main topics selected for this thesis, which are driver consistency in Section 2.3.2 and route balancing in Section 2.4. An initial analysis will be conducted on how consistency[1] more generally is currently defined and utilized in both VRPs and DARPs. The primary objective is to highlight the scarcity of research conducted on driver consistency in the context of a MP-DARP. Moreover, the aim is to establish that among the limited work carried out in this area, none has incorporated fairness among drivers, or route balancing, as an additional focus.

In Section 2.6, a comprehensive survey to investigate the various meta-heuristics that have been utilized to address DARPs is conducted.

2.1 VRPs in general

A complete literature review on the topic of VRPs at large has been conducted recently by Zhang et al. (2021). Their review involved categorizing VRPs based on their characteristics and potential applications, and providing an overview of the different resolution methods that are available for each type.

The first VRP problem was introduced by Dantzig and Ramser (1959) and used to assess a routing optimization problem of oil tankers. More generally, any kind of VRP always involves meeting the demands of a set of customers with a fleet of available vehicles while minimizing distance, cost, and time consumption.

According to the work conducted by Zhang et al. (2021), VRPs can be classified into five main different categories:

- **Capacitated Vehicle Routing Problem (CVRP)**, which incorporates capacity constraints for each vehicle in the fleet. These capacities can be uniform for all vehicles in the fleet, which is known as a homogeneous fleet, or different for each vehicle, referred to as a heterogeneous fleet.

[1] As it will be explained in the literature review, there are several types of consistency: time consistency, territory consistency, delivery consistency, and driver consistency (Kovacs et al. (2014)).

- **Vehicle Routing Problem with Time Windows (VRPTW)**, which adds a time window constraint to each customer point. It means that each delivery point has a specific time window during which it can be serviced, with both the earliest and latest start time for delivery.

- **Split Delivery Vehicle Routing Problem (SDVRP)**, which differs from traditional VRPs in that a single customer's demand can be completed by more than one vehicle.

- **Pick-up and Delivery Vehicle Routing Problem (PDVRP)**, which involves the pickup of goods at one location and the delivery of those goods to another location. It differs from traditional VRP in that it requires vehicles to make two distinct stops for each customer. It can also imply the consideration of both pickup and delivery time windows, adding an additional layer of complexity to the problem.

- **Dynamic Vehicle Routing Problem (DVRP)**, which takes into account dynamic factors such as traffic conditions, unpredictable changes in customers' demands, accidents, or weather conditions, which can impact previously established routes.

It has to be noted that combining these aspects in a single VRP can increase the complexity of the problem but also make it more realistic and practical, as it takes into account real-world factors such as capacity limitations, time windows, dynamic conditions, and pickup and delivery requirements.

In addition to the five main classes cited above, there are other derivatives that incorporate additional constraints or objectives. One example is the Green Vehicle Routing Problem (GVRP), which takes into account ecological factors such as reducing carbon emissions, fuel consumption, or other environmental impacts in the optimization process. Another is the Consistent Vehicle Routing Problem (ConVRP), which will be detailed in Section 2.3.2.

2.2 DARPs in general

This section of the literature review draws heavily upon the works of Ho et al. (2018) and Molenbruch et al. (2017b), which provide the latest state-of-the-art analyses of DARPs. These papers serve as a robust foundation for this investigation and offer valuable insights that can be further customized to the problem at hand. This research aims to build on the previous studies by including additional findings, which will help to advance the understanding of the driver-consistent DARP and route balancing.

The DARP is a variant of the VRP that focuses on the transportation of people rather than goods. It is a generalization of both the PDVRP and the VRPTW, as it requires the scheduling of pickups and drop-offs of passengers within specific time windows, and is part of the more general TOD problems (Paquette et al. (2009)). In DARPs, a fleet of vehicles serves a set of passengers who have specific pickup and drop-off locations, along with time windows in which they need to be picked up and dropped off. The objective is to minimize the total travel distance while adhering to various constraints such as vehicle capacity, time windows, and passenger preferences (Cordeau and Laporte (2003)).

The DARP has various real-life applications, including non-profit transportation services for elderly, disabled, or injured people. The main goal of DARP in this context is to minimize the cost of providing transportation services while respecting operational constraints, just like in the VRPs (Qu and Bard (2013), Qu and Bard (2015) and Karabuk (2009)). Other potential applications of DARP exist, such as airport transportation for the same type of patients, intra-hospital transportation of patients, supplies or equipment, and integration of dial-a-ride services into public transportation systems. This thesis will primarily focus on exploring the practical implementation of DARP within the context of non-profit transportation services for elderly, disabled, or injured individuals.

2.2.1 Problem characteristics

According to Cordeau and Laporte (2003), the classical definition of the DARP involves a directed graph containing nodes that represent pickup and delivery locations of patients, and arcs that connect nodes and are associated with a travel time and cost. Each route in DARP starts and ends at a depot, and the network of nodes can include one or several depots. The length of each route is limited by a maximum duration, either set by the driver or based on available working hours in a day. The vehicles' capacity cannot be surpassed either. In a single patient request, it is mandatory to visit the pickup location before the delivery location to ensure route feasibility, and both of these locations must be visited by the same vehicle. Additionally, each patient is associated with a specific service duration, which represents the time required to load or unload the patient at its pickup or delivery location. Each patient can specify time windows within which they prefer to be picked up or delivered to their desired location. Finally, each patient has a maximum ride time limit for their convenience, which cannot be exceeded. This last constraint, unique to the DARP, adds complexity to the problem compared to traditional VRPs. This complexity is further increased when considering smaller time windows in the DARP scenario.

2.2.1.1 Types of constraints

In the context of static and deterministic DARPs, most of the literature focuses on problems featuring a homogeneous fleet of vehicles, pickup and delivery time windows, maximum ride time for users, maximum route duration, and vehicle capacity. These constraints can be approached in two distinct ways:

1. As **hard constraints**, where any deviation from the constraints is not allowed. This is the approach taken in articles such as Chassaing et al. (2016) and Ritzinger et al. (2016).

2. As **soft constraints**, where violations are permitted, but a penalty is incurred in the objective function, proportional to the degree of the deviation, in order to discourage such occurrences. An example of this approach can be found in Urra et al. (2015).

2.2.1.2 Heterogeneity

Many problem definitions assume that patients and vehicles in DARP are homogeneous. This means that all patients have the same demand characteristics and cannot specify any additional requirements such as the need for wheelchair or stretcher transportation. Similarly, all vehicles are assumed to have the same properties and capacity. However, in real life, patients may have specific physical needs that require more customized transportation services (Parragh (2011)). To address these requirements, new **heterogeneous problem formulations** have been created that allow for customized patient requests and a fleet of vehicles with configurable options to become more demand-responsive.

With regards to the heterogeneity of users' demands, some patients may have specific requirements or expectations regarding the service provided, as discussed in Ilani et al. (2014). Other papers, such as Zhang et al. (2015) and Liu et al. (2015), consider the potential requirement of accompanying people for patients. Additionally, research has explored the combination of heterogeneous patients with heterogeneous drivers, as in Malheiros et al. (2021), Molenbruch et al. (2017) and Masmoudi et al. (2017).

The emergence of various research papers that concentrate on the heterogeneity of patients' demands has led to an increased focus on heterogeneous vehicles in research. Articles such as Parragh (2011), Parragh et al. (2012) and Detti et al. (2017) have investigated the challenge of ensuring compatibility between drivers and patients. There are some studies that take into account both regular and extra vehicles, with the objective of reducing the fixed cost associated with the additional vehicles required (Guerriero et al. (2013). These studies assume that the patients are similar and do not explore the compatibility between them. In an effort to reduce the environmental impact of shared-mobility systems, mixed fleets of vehicles are increasingly being considered, as discussed in a study by Masmoudi et al. (2020). These mixed fleets involve the use of two different types of vehicles, such as conventional vehicles with unlimited fuel supply and alternative fuel vehicles.

Other studies sequence the picking up and delivering of different types of patients in accordance with the layout of the vehicle and decisions on the vehicle configuration for each trip (Qu and Bard (2013), Qu and Bard (2015), Braekers et al. (2014) and Braekers and Kovacs (2016)), but at the cost of increasing complexity. In Tellez et al. (2018), the authors suggest a variant of the heterogeneous DARP that allows for en-route modifications of the vehicle's inner configuration. This means that during the vehicle's route, its capacity and other features can be adjusted to accommodate changing demands or operational requirements.

2.2.1.3 Routing properties

In the traditional DARP model, a patient is transported in a single vehicle throughout their entire trip. However, some more recent studies explore the possibility of transferring a patient from one vehicle to another during the same journey, with the aim of minimizing waiting times and ensuring synchronization (Reinhardt et al. (2013), Schönberger (2017)). While this may not always be feasible, such considerations can improve the overall efficiency and reduce the total distance traveled by the system, as demonstrated by Cortés et al. (2010) and Masson et al. (2014).

Instead of being transferred to another vehicle, users can also be transferred to public transit services during their journey, which can reduce the operating costs of the overall system. Molenbruch et al. (2021), Posada et al. (2017), Ronald et al. (2015) and Häll et al. (2009) have extensively studied this possibility, along with its benefits and drawbacks. This approach can also increase the productivity of the system but may come at the cost of decreased patient convenience due to increased waiting times. Moreover, Melis and Sörensen (2022) conducted a recent study that proved the effectiveness of a network of on-demand buses compared to traditional public transportation systems.

Some papers, like Parragh et al. (2015), have explored the concept of dividing multi-load requests into multiple rides, which has been shown to enhance efficiency from a wider operational standpoint.

The Fixed Route Dial-a-Ride Problem (FRDARP) is a variation of the DARP where patient requests are restricted to terminals located on a predetermined route. This variant enables the grouping of patients and the generation of more efficient schedules. Although this problem has not been extensively studied, recent work by Grinshpoun et al. (2022) has focused on simplifying the process of constructing optimal predetermined routes.

2.2.1.4 Workforce requirements

Since DARPs are designed to cater to people with restricted mobility, these patients may have specific requirements regarding their drivers. These requirements can be addressed in various ways:

- Patients may prefer to have the same driver over multiple time periods, which is known as **driver consistency** and is the primary focus of this thesis. This topic has been extensively studied by Braekers and Kovacs (2016)) and will be a major challenge in this research.

- Patients may also request to have **accompanying individuals** with them on the vehicle (studied by Parragh (2011) and Parragh et al. (2012)). This can lead to additional concerns regarding loads of these staff members, studied by Lim et al. (2017) and Zhang et al. (2015).

2.2.1.5 New insights for DARPs

The primary objective of shared-mobility systems is to reduce gas emissions and combat global warming, and as a result, these systems, including the DARP, are continually evolving. The development of new systems is aimed at achieving this objective by promoting sustainable and eco-friendly transportation solutions. One such emerging system is the Electric Autonomous Dial-a-Ride Problem (E-ADARP), which involves a fleet of autonomous electric vehicles. In addition to the typical DARP constraints, the E-ADARP also incorporates a partial recharging policy to account for the vehicles' limited battery life. A recent study by Su et al. (2022) delved further into this topic.

2.2.2 Objective functions

Most objective functions in DARPs contain only a single objective, which is cost minimization by minimizing the total distance traveled (D'Souza et al. (2012)). This is because transportation costs are often directly linked to the distance covered by a vehicle. However, some problems may consider other objectives, such as minimizing the number of vehicles used or maximizing their efficiency (Garaix et al. (2011)), or even minimizing staff workload (Lim et al. (2017)). In some cases, the goal may even be to maximize profit rather than simply minimize costs (Parragh et al. (2015), or to maximize system reliability (Pimenta et al. (2017)). Other problem formulations minimize users' inconvenience metrics, such as their total travel time and waiting times. Some papers also deal with ecological measures, such as vehicle emissions (Chen et al. (2022), Atahran et al. (2014) and Chevrier et al. (2012)).

Other applications deal with multiple objectives where different goals are taken into consideration simultaneously, which are called Multi-Objective Dial-a-Ride Problem (MO-DARP). Researchers working on MO-DARP have identified three major types of approaches to manage the multiple objectives:

1. A **weighted sum of different measures**, as proposed in Mauri and Lorena (2006), Mauri et al. (2009), Jorgensen et al. (2007), Kirchler and Calvo (2013) and Melachrinoudis et al. (2007). To ensure that the weights are comparable and add up to one, some authors normalize them between 0 and 1, as in Hu et al. (2019). This weighted sum has the advantage of reducing the post-solution efforts required to make a final decision. However, this method requires that

all objectives have common units and that the relative importance of each objective can be quantified.

2. A **lexicographic objective function** with objectives classified in the order of importance, as realized by Garaix et al. (2010). This approach involves optimizing a higher-level objective first and then, if possible, a lower-level objective. This method is particularly useful when objectives cannot be represented by the same unit and when one objective is significantly more important than the other. However, this hierarchical structure can prevent the final decision-maker from analyzing possible trade-offs between different objectives.

3. A **Pareto frontier** of the problem, which consists of a set of solutions that are not dominated by others, as realized in Zidi et al. (2012). The Pareto frontier represents the set of all feasible solutions that cannot be improved in one objective without worsening at least one other objective. In other words, it defines the trade-off between different conflicting objectives. Similar to the lexicographic approach, this method does not require finding weights or converting different measures into a common unit. In the end, the decision-maker can choose among several solutions that all provide the same level of objective function but favor one objective over the other (Parragh et al. (2009)). This approach is relevant when the relative importance of criteria is uncertain and helps to analyze trade-offs between opposing objectives (Paquette et al. (2013)).

All three approaches employ distinct metrics based on their respective objectives. However, except for Lehuédé et al. (2014), none of the authors have considered the interactions between them.

A study by Guerreiro et al. (2020) provides an overview of the various algorithmic techniques that can be specifically used to solve MO-DARPs, as well as the corresponding available literature.

2.2.3 Taxonomy

Before delving into the specific variant of the DARP being considered in this study, a brief explanation of the various existing categories of DARP will be provided.

According to Ho et al. (2018), DARPs can be classified according to two different aspects:

- Whether the decisions are made a priori and cannot be changed anymore, or if they can be made even after the start of operations when new information is received and can be changed through the entire process. The former is referred to as **static DARP**, while the latter is referred to as **dynamic DARP**.

- Whether the information received is known with certainty at the moment decisions are made, which is called **deterministic DARP**, or if the decisions have to be made with still undetermined information, which is called **stochastic DARP**.

These two classifications give rise to four possible combinations, as shown in Table 2.1.

It should be emphasized that in practice, the majority of real-world DARPs are dynamic and stochastic in nature. This is due to the fact that many factors can be unpredictable, such as traffic congestion, unexpected patient no-shows, vehicle breakdowns, and adverse weather conditions. As a result, drivers may need to modify their plans and adjust their initial schedules to cope with the coming changes.

	Information known with certainty (at time of decision)	Information not known with certainty (at time of decision)
Decisions cannot be modified after the start of operations	**Static and deterministic**	**Static and stochastic**
Decisions can be modified in response to new information received after the start of operations	**Dynamic and deterministic**	**Dynamic and stochastic**

Table 2.1: Taxonomy of Dial-a-Ride Problems, reproduced from Ho et al. (2018).

However, the majority of the research on DARPs has focused on static and deterministic scenarios. Similarly, for the purposes of this study, the problem being addressed will also be static and deterministic. This is because in order to optimize driver-patient relationships and ensure fairness among drivers in a MP-DARP, information about patients' requests over multiple periods must be analyzed in advance. This necessitates careful planning and precludes the decision-maker from accepting new information during the week. Due to the aforementioned factors, the upcoming section will exclusively concentrate on the research conducted in the field of static and deterministic DARPs. The existing body of work in this area is more applicable to the problem at hand.

2.3 Focus on patient satisfaction

In the field of DARPs and VRPs, there is a growing emphasis on taking customer satisfaction into consideration (Paquette et al. (2009), Paquette et al. (2012)). With rising customer expectations, problem definitions must now include service quality considerations in order to stay competitive. The paper by Nasri et al. (2021) provides a comprehensive review of existing research on service quality in DARPs. Based on the findings of a study by Paquette et al. (2009), there is no universally accepted definition of quality. Instead, quality is seen as a combination of various factors, including safety, comfort, and reliability. The study also found that driver characteristics were important to users when considering quality.

2.3.1 Time windows and maximum user ride time

In their study, Molenbruch et al. (2017a) investigate the relationship between service quality levels and operational costs in DARPs. Their objective is to minimize the maximum deviation from the patients' preferred time. The study also highlights two important specifications in DARP services that directly affect the service experience of patients. These are:

- **Time window width at pickup or delivery locations**: This specification measures the deviation from the patient's chosen departure or arrival time and affects the waiting time of the patient. Increasing the width of time windows in DARP services gives greater flexibility for the service provider, but at the same time, it results in lower service quality for patients. In order to address the trade-off between time window width and service quality in DARPs, some authors, such as Melachrinoudis et al. (2007) and Jorgensen et al. (2007) have suggested adjusting time windows through a relaxation process - if the time windows are violated, they are then penalized in the objective function.

- **Maximum user ride time**: This specification sets a maximum time limit that patients should not exceed and corresponds to the total time spent by a patient in the vehicle, riding or waiting. The maximum user ride time can be the same for all patients, as in the study of Cordeau and Laporte (2003), or differ among patients, as assumed in Molenbruch et al. (2017a), Chassaing et al. (2016) and Parragh et al. (2009), who argue that this information is highly specific to each of them. This maximum user ride time can also be enforced by imposing time windows for each patient, both for the pickup and the delivery, as in Wolfler Calvo and Touati-Moungla (2011), Parragh et al. (2015) and Atahran et al. (2014). In contrast to the approach of setting user ride time and waiting time as upper bounds in constraints, Pfeiffer and Schulz (2022a) and Pfeiffer and Schulz (2022b) take a different approach and focus on minimizing these factors, specifically user ride time and waiting time, as the primary objectives in their optimization models. Fahmy (2022) concentrates on minimizing extra ride times, among other objectives, allowing this constraint to be relaxed.

The model proposed by Nasri and Bouziri (2017) specifically addresses service quality in depth in DARPs. The problem addressed in this model involves minimizing travel costs and total travel time for all vehicles while penalizing time window violations. Additionally, the model provides more customized routes for patients with patient-dependent constraints. In Nasri et al. (2022), the goal is to minimize total travel costs, which includes penalties for waiting times.

The studies of Dong (2022) and Dong et al. (2020) focus on defining patients' utilities, which are directly related to the satisfaction of time windows and maximum ride times. The studies allow for violations of these constraints but aim to keep patients' utilities within predefined limits. The utility function is then used as a constraint to ensure that the solutions generated are acceptable based on patients' preferences.

2.3.2 Consistency in VRPs in general

Kovacs et al. (2014) demonstrate that providing a consistent service is also essential to reach customer satisfaction in VRPs. Customers prefer to be serviced at regular times of the day and by the same driver each time. While ensuring consistency may lead to higher operating costs, companies that can achieve a high level of consistency at a reasonable cost have a significant competitive advantage over their competitors. This finding is supported by Braekers and Kovacs (2016), who emphasize the importance of balancing consistency and cost-effectiveness in transportation services.

Kovacs et al. (2014) define three types of consistency in multi-period VRPs:

- **Arrival time consistency**, which is achieved when a driver visits each customer at similar times of the day in the long run. This ensures that customers can anticipate when the driver will arrive, which can improve customer satisfaction.

- **Person-oriented consistency**, also known as driver consistency from the customer's point of view. It aims to reduce the number of different drivers that serve a customer. From the driver's perspective, it is called territory consistency, which means that each driver is assigned to the same region repeatedly, allowing the driver to become familiar with the area and potentially improve efficiency.

- **Delivery consistency**, which is particularly relevant for goods delivery. It means that a customer is replenished at regular intervals with similar delivery quantities or maintains a stable inventory level. This can help ensure that the customer always has sufficient inventory and can plan their operations accordingly.

The problem formulation with a focus on consistency is referred to as the **Consistent Vehicle Routing Problem (ConVRP)** in the literature, as highlighted in the study of Lespay and Suchan (2021). It is important to note that consistency requirements must link different days in the planning horizon, and each day cannot be treated as an independent one-day period problem, as assumed by Lespay and Suchan (2021), Goeke et al. (2019), Kovacs et al. (2015) and Groër et al. (2009).

In the context of this study and focusing on DARP, the concept of **driver consistency** is of utmost relevance. This is because patients are picked up and delivered based on their scheduled appointments, making the arrival time consistency less significant in comparison. Additionally, since DARP primarily deals with the transportation of individuals, delivery consistency does not hold relevance in this context.

Paquette et al. (2012) demonstrate that driver consistency is one of the most important service quality criteria for elderly and disabled people. Woodward et al. (2004) emphasize the importance of having personnel with knowledge and skills regarding patients and services required. It is beneficial to have a staff who is familiar with the patients' preferences, culture, and routines, as this can make patients feel more comfortable and simplify communication between parties.

Recent research on driver consistency in the static DARP is limited, with the most recent work conducted by Braekers and Kovacs (2016). However, more recent studies have investigated driver consistency in the context of VRPs, such as those conducted by Yang et al. (2022) and Rodríguez-Martín and Yaman (2022).

There are various ways to model driver consistency:

- The common approach is to **assign each patient to a single driver** through a constraint. However, this approach can result in increased operational costs, as drivers may not be optimally assigned based on their availability and location. This approach can be seen in Groër et al. (2009), Lespay and Suchan (2021), Coelho et al. (2012), Coelho and Laporte (2013), Francis et al. (2006), Francis et al. (2007), Kovacs et al. (2014), Tarantilis et al. (2012) and Rodríguez-Martín and Yaman (2022).

- This constraint is relaxed, **allowing more drivers per patient, but bounding this number by a predefined parameter equal or larger than 1** (Braekers and Kovacs (2016), Kovacs et al. (2015), Luo et al. (2015), and Zhong et al. (2007)).

- Yang et al. (2022) define a clear **driver consistency measurement to be minimized in the objective function**. This is a measurement logarithmically linked to two factors: the number of days a patient requests service during the planning period and the number of drivers assigned to provide service to that patient.

Specifically, in Braekers and Kovacs (2016), the maximum number of different patients that could transport a driver over a multi-period planning horizon was limited, and different values were tested as upper bounds. The objective was to minimize the total routing costs. Specifically, upper bounds of 1, 2, and 3 were tested, and the cost of driver consistency was found to be more expensive when restricting the upper bound to 1. This resulted in cost increases ranging between 8,52% and 11,96% compared to the situation where driver consistency was not included as a constraint. On the other hand, allowing a maximum of 2 drivers per patient resulted in a cost increase of only 0,91% to 1,67%. Moreover, it was proven that allowing up to 3 drivers did not significantly reduce costs. This assumption is also proved in Kovacs et al. (2015).

Several authors have approached the issue of improving driver consistency by penalizing undesired driver-patient pairs in the objective function of the problem. Studies by Coelho et al. (2012), Zhong et al. (2007), Coelho and Laporte (2013), Eveborn et al. (2006), and Smilowitz et al. (2013) have implemented this approach to tackle the problem. Particularly, in Smilowitz et al. (2013), it has been demonstrated that giving slight importance to driver consistency in the objective function does not result in a significant increase in costs. However, it leads to a considerable improvement in the regularity of the routing plan, as demonstrated in Braekers and Kovacs (2016).

In the studies by Smilowitz et al. (2013) and Feillet et al. (2014), driver consistency is improved in a two-phase approach. In the first phase, initial routes are created without considering driver consistency. Then, in the second phase, driver consistency is improved, and each route is assigned to the driver who is the most familiar with the patients on the respective route or the region through which the route is going.

2.4 Focus on route balancing

The research conducted by Azad et al. (2016) sheds light on the fact that not a lot of VRP formulations take the issue of fairness into consideration. However, fairness considerations are critical for ensuring that the workload is distributed equitably among workers, and any transportation problem must take this dimension into account. Azad et al. (2016) also provides a review of the fairness issues that arise in VRPs.

For two decades, authors have started to deepen the issue of fairness among drivers, and some progress has been made in this area. However, these findings have only been incorporated into VRPs, and there has not been any study that has included route balancing in DARPs. A comprehensive literature review of such Vehicle Routing Problem with Route Balancing (VRPRB) can be found in Matl et al. (2018).

There are various ways to define the workload for a driver in the context of delivery or transportation services currently used in the literature. One possible approach is to consider the **total distance traveled** by each driver, which is often correlated with the **total route duration**. In their work, Feng and Wei (2022) draw attention to the difference between the total distance traveled and the total route duration. While the former only accounts for the physical distance covered, the latter includes the time spent on servicing each patient. Therefore, even if two vehicles cover the same total distance, their respective route duration can differ depending on the number of patients on board and the time needed to serve them. Another way is to quantify the workload based on the **number of units delivered** or the **number of satisfied patients**.

Here is a reformulation of the list of objective functions to minimize unfair treatments among drivers, based on Lozano et al. (2016) and Matl et al. (2018):

- **Range**: minimize the difference between the maximum and the minimum indicator value[2]

[2] This particular indicator value is selected among the various workload definitions mentioned above: total distance traveled, total route duration, number of units delivered or number of satisfied patients.

among all drivers;

- **Maximum value**: minimize the maximum indicator value among all drivers;
- **Variance** or **standard deviation**: minimize the variance or the standard deviation of indicator values among all drivers, representing the spread of the indicator values;
- **Relative deviation**: minimize the sum of the relative deviation of each indicator value for each driver from the maximum value of the same indicator among all drivers;
- **Cumulative difference**: minimize the cumulative differences overall drivers between their respective indicator values and the minimum indicator value among all drivers;
- **Mean absolute difference**: minimize the mean absolute difference between each indicator value and the mean of indicator values among all drivers;
- **Gini coefficient**: minimize the Gini coefficient of indicator values among all drivers, which represents the inequality of the indicator distribution;
- **Ratio between the maximum workload and minimum workload of drivers**: minimize the quotient obtained by dividing the indicator value of the driver with the highest indicator by the indicator value of the driver with the lowest indicator (introduced by Jingjing et al. (2022)).

According to experiments conducted by Lozano et al. (2016), the variance, standard deviation, mean absolute difference, and Gini coefficient objective functions are more effective than the others in minimizing unfair treatment among drivers. These indicators consider the distribution differences among all drivers, rather than solely comparing them with the maximum or minimum value. Therefore, they provide a more comprehensive view of the fairness issue in workload distribution. Individuals not only strive to improve their own welfare but also value being treated fairly in comparison to others. The importance of fairness in addition to personal gain is a recognized aspect of human behaviour, as highlighted by the findings of Sánchez et al. (2022) in their recent study. In this thesis, the **standard deviation** is chosen as the specific indicator for measuring route balancing.

2.5 Integration of driver consistency with route balancing

According to the explored literature, there is **only one research article that combines the concept of driver consistency with route balancing in a single problem formulation for the VRP** (Mancini et al. (2021)). However, there is currently no research that has attempted to combine these concepts in the context of the DARP. The authors of this study suggest that these objectives can be included in the same objective function without incurring significant additional costs. This finding was influential in the decision to pursue a similar approach in the current study.

2.6 Meta-heuristics used

The DARP is an NP-hard problem, as explained in Section 1.1. Although smaller instances of the DARP can be solved using exact methods such as the branch-and-cut algorithm or the column generation technique, these methods can be time-consuming and may not be adaptable to larger instances. Nevertheless, to counter these weaknesses, meta-heuristic algorithms are commonly used to solve DARPs. Therefore, this thesis will focus on reviewing the literature on meta-heuristic algorithms that are commonly used to solve DARP.

According to Sörensen and Glover (2013), meta-heuristic is an algorithmic framework that is problem-independent (as opposed to traditional heuristics that are problem-dependent) and operates at a high level to guide the development of heuristic optimization algorithms. The main goal of a meta-heuristic is to identify the best feasible solution among all possible solutions in an optimization problem. The authors also propose various classifications for meta-heuristics:

- **Local search algorithms** attempt to find relatively good solutions to an optimization problem by iteratively changing the current solution through a single move, known as a neighborhood search. This process generates a set of new solutions, called the neighborhood of the current solution, and the current solution is replaced iteratively by one of its neighbors based on a move strategy. Variable Neighborhood Search (VNS), Simulated Annealing (SA), Deterministic Annealing (DA), Tabu Search (TS) and Large Neighborhood Search (LNS) are examples of local search meta-heuristics.

- **Constructive algorithms** build solutions by incrementally adding elements to a partial solution. Once a complete solution is obtained, constructive algorithms are often combined with a local search method to improve the solution. Examples of constructive meta-heuristics include Greedy Randomized Adaptive Search Procedure (GRASP), Ant Colony Optimization (ACO), and the Pilot method.

- **Population-based algorithms** aim to identify good solutions by combining existing solutions from a set, known as a population. Genetic Algorithm (GA), Scatter Search, and Path Relinking (PL) are examples of population-based meta-heuristics.

- **Hybrid algorithms** combine two or three different types of meta-heuristics in an attempt to leverage the strengths of each component algorithm. One example of a hybrid meta-heuristic is the Memetic Algorithm (MA), which incorporates local search algorithms within a genetic algorithm framework.

As mentioned in Zidi et al. (2012), population-based algorithms can be computationally expensive and memory-intensive as they generate and evaluate a large number of solutions in the search process. This can result in longer execution times compared to local search algorithms. Given that the DARP often needs to be solved in real time, the current literature tends to prioritize the use of local search algorithms, which can quickly generate solutions and provide satisfactory results.

According to Braekers et al. (2014), meta-heuristics offer a more extensive exploration of the objective space than traditional heuristics. This is due in part to their ability to generate and consider deteriorating or infeasible intermediate solutions during the search process. As a result, meta-heuristics are less likely to become trapped in local optima, which is a common issue with traditional heuristics. **In this thesis, the SA algorithm, specifically a local search approach, will be employed**. The rationale behind this choice will be thoroughly explained in Section 4.1.

2.7 Summary and thesis contribution

Table 2.7 summarizes the literature on static and deterministic DARP from 2006 to the present. Each reference in the table indicates the characteristics of the problem formulation, including the number of depots (single, S or multiple, M), the fleet of vehicles homogeneity (homogeneous (HO) or heterogeneous (HE)), the inclusion of vehicle capacity constraints, patient time windows and maximum ride times, driver maximum route duration, and the presence of a single (S) or multiple (M) objective(s) in the objective function. Additionally, if the authors used a meta-heuristic or heuristic to solve the problem, the name of the method is mentioned in the last column. References marked with a "/" only discuss exact solution methods.

The main objective of this thesis is to **investigate the feasibility of incorporating driver consistency and route balancing into a single problem formulation for the MP-DARP**. This is a totally new research direction as no prior work has explored the integration of these aspects in the DARP context. Although driver consistency and route balancing have been studied to some extent in the VRP literature, there is very little research on these topics in the DARP domain, particularly any research regarding route balancing. Furthermore, only one article has dealt with the integration of both features in the VRP. To address this research gap, **a meta-heuristic approach will be adopted, and more specifically, the SA algorithm will be utilized**.

Reference	Depot(s)	Fleet	Cap.	TW	Ride time	Route duration	Obj.	Algorithm
Mauri and Lorena (2006)	M	HE	X	X	X	X	M	SA
Jorgensen et al. (2007)	S	HO	X	X	X	X	M	GA
Melachrinoudis et al. (2007)	M	HE	X	X			M	TS
Häll et al. (2009)	S	HO	X	X	X	X	S	/
Karabuk (2009)	M	HE	X	X	X	X	S	Heuristic
Mauri et al. (2009)	M	HE	X	X	X	X	M	SA
Parragh et al. (2009)	S	HO	X	X	X	X	M	VNS and PL
Cortés et al. (2010)	M	HO	X	X			S	/
Garaix et al. (2010)	M	HE	X	X	X		M	/
Garaix et al. (2011)	M	HE	X	X	X		S	/
Parragh (2011)	S	HE	X	X	X	X	S	/
Wolfler Calvo and Touati-Moungla (2011)	S	HO	X	X			M	TS
Chevrier et al. (2012)	S	HE	X	X	X	X	M	GA
D'Souza et al. (2012)	S	HO				X	M	SA and GA
Parragh et al. (2012)	S	HE	X	X	X	X	S	
Zidi et al. (2012)	S	HE	X	X			M	SA
Guerriero et al. (2013)	S	HE	X	X	X	X	S	TS and GRASP
Kirchler and Calvo (2013)	S	HO	X	X	X	X	M	TS
Paquette et al. (2013)	S	HE	X	X	X	X	M	TS
Qu and Bard (2013)	S	HE	X	X			S	LNS
Reinhardt et al. (2013)	M	HE	X	X		X	M	SA
Atahran et al. (2014)	S	HE	X	X	X		M	GA
Braekers et al. (2014)	M	HE	X	X	X	X	S	DA
Ilani et al. (2014)	M	HE	X	X	X		S	Shortest path
Masson et al. (2014)	M	HO	X	X	X	X	S	LNS
Lehuédé et al. (2014)	M	HO	X	X	X	X	M	LNS
Liu et al. (2015)	S	HE	X	X	X	X	S	/
Parragh et al. (2015)	S	HO	X	X	X	X	S	VNS
Qu and Bard (2015)	S	HE	X	X	X	X	M	/
Lehuédé et al. (2014)	M	HO	X	X	X	X	M	LNS
Urra et al. (2015)	S	HO	X	X	X	X	M	Hyper heuristic
Zhang et al. (2015)	S	HO	X	X	X	X	M	GA
Braekers and Kovacs (2016)	S	HE	X	X	X	X	S	LNS
Chassaing et al. (2016)	S	HO	X	X	X	X	S	Evolutionary local search
Ritzinger et al. (2016)	S	HO	X	X	X	X	S	LNS
Detti et al. (2017)	M	HE	X	X	X		M	TS and VNS
Masmoudi et al. (2017)	S	HE	X	X	X	X	S	GA
Molenbruch et al. (2017)	S	HO	X	X	X	X	M	VNS
Pimenta et al. (2017)	S	HO	X		X		S	GRASP
Posada et al. (2017)	S	HE	X	X	X		S	/
Schönberger (2017)	M	HE	X		X	X	S	MA
Ho et al. (2018)	S	HO	X	X	X	X	M	Multi-atomic annealing
Tellez et al. (2018)	M	HE	X	X	X		M	LNS
Alisoltani et al. (2019)	M	HO	X	X			S	Cluster-based approach
Belhaiza (2019)	S	HE	X	X	X	X	S	GA and LNS
Hu et al. (2019)	S	HO	X	X	X	X	M	/
Dong et al. (2020)	S	HE	X	X	X	X	S	/
Masmoudi et al. (2020)	S	HE	X	X	X	X	S	LNS
Souza et al. (2020)	S	HE	X	X	X	X	S	VNS
Malheiros et al. (2021)	M	HE	X	X	X	X	S	Heuristic
Rist and Forbes (2021)	S	HO	X	X	X	X	S	/
Chen et al. (2022)	S	HO	X	X			M	LNS
Dong (2022)	S	HE	X	X	X	X	S	/
Fahmy (2022)	S	HO	X	X	X	X	M	GA and VNS
Grinshpoun et al. (2022)	S	HE	X	X		X	S	Shortest path
Melis and Sörensen (2022)	S	HO	X				S	LNS
Nasri et al. (2022)	S	HE	X	X	X	X	S	Evolutionary descent
Pfeiffer and Schulz (2022a)	S	HO	X				S	LNS
Pfeiffer and Schulz (2022b)	S	HO	X				S	/
Su et al. (2023)	M	HE	X	X	X	X	M	DA
Ham (2023)	S	HE	X	X			S	/
Hubert (2023)	S	HE	X	X	X	X	M	SA

Table 2.2: Literature review on static and deterministic DARP, based on Ho et al. (2018) until 2018. The letter "S" represents the term "Single", while "M" stands for "Multiple". In addition, "HO" denotes "Homogeneous" and "HE" indicates "Heterogeneous".

3 Problem description

This chapter formally defines the problem in terms of parameters, variables, and constraints. The objective function is discussed, along with the indicators used to measure driver consistency and route balancing, and how they are integrated.

In this thesis, the aim is to solve the **heterogeneous MP-DARP using a meta-heuristic approach**. Parameters, variables, and constraints required for the problem were formulated by drawing on relevant theories and insights from various sources, including Mitrović-Minić and Laporte (2004), Cordeau and Laporte (2007), Kovacs et al. (2014), Molenbruch et al. (2017b) and Paquay et al. (2020).

3.1 Parameters

- **Network**:
  - A set of locations that represent patients' homes, medical centres, and the depot;
  - Distances and travel times between each pair of locations.
- **Time horizon**:
  - Total number of working days;
  - Total duration of each working day.
- **Requests of patients**:
  - A list of request types to satisfy over the time horizon. A single patient may have varying request types over the whole time horizon, and some of these requests may be repeated on multiple days. For instance, a patient may require transportation for dialysis appointments on certain days and transportation for physical therapy appointments on other days. Each of these request types contains:
    - A departure and a delivery location;
    - Days over which the request type is repeated throughout the entire time horizon;
    - A load, split into different resource types. Resource types can, for example, include normal seats, people in wheelchairs, or accompanying people;
    - Time windows for the pickup and the corresponding delivery;
    - A maximum ride time;
    - A service time, which represents the time needed for the patient to be loaded or unloaded.

- **Drivers**:
  - Total number of drivers, which is equal to the total number of available vehicles (throughout the rest of the study, these terms will be used interchangeably);
  - Starting and ending times of the working day for all drivers;
  - Total capacity of each resource type in each vehicle;
  - Maximum route duration for each driver.

3.2 Variables

There are two main types of variables: decision variables and auxiliary variables. Decision variables are those that are subject to decisions and are adjusted to obtain the best solution to the problem. On the other hand, auxiliary variables are derived from the values of decision variables and are used to model the constraints of the problem.

3.2.1 Decision variables

- **Sequence of requests** that will be satisfied by each driver on each day, enabling the evaluation of **driver consistency**;
- **Schedule of all routes**: arrival time, start time of service and leaving time at each location.

3.2.2 Auxiliary variables

- **Routes of each driver on each day**:
  - Load of the vehicle after each visited location for each resource type;
  - Earliest and latest arrival time and departure time at each location;
  - Slack time at each location, which is defined as the difference between the latest departure time and the earliest departure time from the location (as in Mitrović-Minić and Laporte (2004));
  - Working load of the route, corresponding to the difference between the arrival time of the driver at the final depot location and its corresponding departure time from the initial depot.
- **Total travel time** for each driver over the entire time horizon, which refers to the cumulative duration of all their routes combined. It represents the sum of the working loads of all routes assigned to each driver and allows the measurement of **route balancing**.

3.3 Constraints

The problem must satisfy various constraints, including patient constraints, driver constraints, routes feasibility and schedule constraints. All of these constraints will be considered to be **hard** in the problem being studied, meaning that no violations are allowed. The aim is to prioritize the satisfaction and comfort of both patients and drivers. Hence, meeting their requirements without any violation is of utmost importance.

- **Patient constraints**:
  - Each request is satisfied exactly once;
  - Each patient is served within its specified time windows;
  - The riding time of each patient does not exceed the corresponding maximal duration.
- **Route feasibility constraints**:
  - A pickup and its corresponding delivery have to be performed by the same driver;
  - A pickup is always performed before its corresponding delivery.
- **Drivers constraints**:
  - Each driver starts and ends its route at the depot location;
  - Each driver starts and ends its route within the depot opening hours;
  - The capacity of each resource type is respected for each vehicle on each day after each visited location;
  - The duration of each route does not exceed the maximum route duration imposed by each driver.
- **Schedule constraints**:
  - The arrival time at each location is always before the service starting time, for each vehicle at each location on each day;
  - The departure time from each location is always after the starting time of the service plus the service time of each related patient for each vehicle at each location on each day;
  - There is sufficient time for the vehicles to travel from one location to the next based on the distance between them and the speed of the vehicles.

3.4 Objective function discussion

The purpose of this section is to provide a detailed explanation of how the objective function of the problem is formulated and how it incorporates both the objectives of achieving **driver consistency** and **route balancing** simultaneously.

3.4.1 Driver consistency assessment

The literature, as discussed in Section 2.3.2, has already explored different approaches to define driver consistency. Assigning a single driver to each patient was found to be too restrictive and significantly increases operational costs. Furthermore, in the problem being studied, enforcing one-to-one relationships between patients and drivers would have compromised route balancing and led to sub-optimal solutions.

Although limiting the number of drivers per patient could have been considered as a potential solution to the problem, as in the work realized by Braekers and Kovacs (2016), such an approach may lead to violating certain constraints to obtain feasible solutions. As highlighted in Section 3.3, it is crucial to ensure maximum patient and driver comfort in the problem under investigation, and thus no constraints can be violated in the pursuit of feasible solutions.

In this thesis, the approach chosen to address the driver consistency issue is to **minimize a measurement in the objective function**. Various possibilities were considered, including:

- Summing up the number of different drivers that serve each patient over the entire time horizon;
- Computing the mean number of drivers seen per patient over the entire time horizon;
- Determining the maximum number of drivers seen by any patient over the entire time horizon.

The second possibility was ultimately chosen among the considered options. This is more representative of the overall situation compared to the first option which only considers all drivers without distinguishing their distribution among the patients' journeys; in the same way, this is also more representative than the third option which only considers the maximum number of drivers for a single patient.

The **driver consistency indicator** can be mathematically represented as follows:

$$\frac{\sum_{p=1}^{P} d_p}{P}$$

where the parameter P denotes the total number of patients and d_p refers to the count of distinct drivers serving each patient p throughout all the working days considered.

3.4.2 Route balancing assessment

The definition of workload in this thesis is based on the **total route duration of each driver over the entire time horizon**, which takes into account not only the distance traveled but also the service times required for loading or unloading patients, as described in Section 2.4. This can be computed by adding up the route duration of each driver over the number of days considered. Furthermore, relying solely on the total number of satisfied patients as the objective function can result in some drivers having to cover longer distances if a patient is located far away from the others, leading to an unfair distribution of working times.

The objective function chosen to assess route balancing, as proposed by Lozano et al. (2016), is the **standard deviation among the total route duration of all drivers**. The advantage of using standard deviation as a measurement is that it reflects how the workload is distributed among drivers, and hence provides a more comprehensive evaluation of the balance among drivers compared to other metrics that only consider the workload of a single driver among the others. Other options like variance or mean absolute difference could have also been considered.

The **route balancing indicator** can be mathematically represented as follows:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

In this formula, n refers to the total number of drivers, while x_i represents the total route duration of driver i over the entire time horizon. More specifically, x_i can be calculated by summing up the route duration of driver i for each day in the time horizon. Finally, $\overline{x}$ represents the mean of the total route duration among all drivers over the entire time horizon.

3.4.3 Relation between both objectives

As explained in Section 2.2.2, when dealing with a MO-DARP with dual objectives like in this case, there are three main ways to integrate them. In this study, the chosen approach is the **lexicographic optimization method**, where the driver consistency objective is given priority over the route balancing objective. The decision to prioritize driver consistency over route balancing is grounded in the principle of prioritizing patient comfort and well-being. Additionally, the optimization of route balancing is consistently addressed as a lower-level objective in the existing literature on this topic.

The idea of using a weighted sum to combine the driver consistency and route balancing indicators was not appropriate, mainly because these indicators are not expressed in the same units, and determining the appropriate weights would have been challenging. The weights chosen would have significantly influenced the results and their relative importance compared to each other. The possibility of using a Pareto frontier was also considered, but ultimately not pursued in this work. Although it would have provided a set of solutions that would have allowed decision-makers to choose based on their preferences, it would have required more computational effort to generate the entire set of solutions.

4 Methodology

This chapter outlines the methodology followed in this thesis to solve the problem at stake. The choice of the Simulated Annealing (SA) algorithm is addressed, including the selection of the initial solution method, parameters choice, and neighborhood structures definition. Finally, the structure of the final solutions is provided.

4.1 Choice of the Simulated Annealing

A study conducted by Baugh Jr et al. (1998) shows that among all meta-heuristics commonly used to solve DARPs in the current literature, including TS, SA, DA, GA, VNS, and LNS, SA appears to be suitable for several reasons:

- The technique can be easily adapted for problems with a well-defined neighborhood structure;
- It has desirable theoretical convergence properties and can find near-globally optimal solutions in a reasonable amount of time with a suitable annealing schedule;
- Other meta-heuristics, such as TS or LNS, can be easily integrated in a SA to create effective hybrid meta-heuristics.

As demonstrated in Section 2.6, the literature has examined the use of the SA to solve the DARP problem, but its application has been less frequent compared to other meta-heuristics such as LNS or TS. Therefore, this thesis seeks to demonstrate the effectiveness of the SA and apply it to a new problem, thus contributing to the advancement of the field of meta-heuristics for solving optimization problems.

The SA is a meta-heuristic algorithm based on the annealing process of metals in metallurgy. The algorithm starts with an **initial solution** and iteratively searches the solution space by accepting new solutions that are either better or worse than the current one, based on a **probability function** that depends on the **temperature** of the system. The algorithm starts with a high temperature that allows it to accept a large number of worse solutions in order to explore the solution space. This is done to prevent the algorithm from being stuck in local optima, compared to traditional descent algorithms that iteratively update the current solution by seeking a slightly better one at each step until certain stopping conditions are met (Pirlot (1996)). In the SA, as the temperature is decreased over time, the algorithm becomes more selective and only accepts new solutions that improve the objective function.

At each iteration, a random perturbation is applied to the current solution to generate a new candidate solution. This perturbation is usually defined by a **neighborhood structure** that determines which solutions can be reached from the current solution by a single move. The probability of accepting a new solution is calculated based on a parameter called the **acceptance probability**, which is a function of the difference between the objective function value of the new solution and the current solution, as well as the temperature of the system. This function is designed to allow the algorithm to accept worse solutions with a decreasing probability as the temperature decreases, while still allowing for some exploration of the solution space. The temperature is usually decreased gradually over time, according to a cooling schedule, until a **stopping criterion** is met, such as a maximum number of iterations or a desired level of solution quality (Pirlot (1996)). A more detailed description of how the SA algorithm works can be found in Algorithm 1.

Algorithm 1 Simulated Annealing algorithm.

Input: Initial solution s, temperature T_0, cooling rate r, plateau length L, stopping criterion, neighborhood structure
Output: Best solution found s^*

```
s* ← s ;                          // Initial best solution is the starting solution
T ← T0 ;                          // Set the initial temperature
while stopping criterion not met do
    for i ← 1 to L do
        Choose randomly a new solution s′ in the neighborhood of s
        ΔI(f(s′) − f(s)) ← compute the objective indicator difference between s′ and s
        if ΔI < 0 then
            s ← s′
            if f(s) < f(s*) then
                s* ← s ;          // Update best solution found so far
            end

        else
            p ← acceptance probability based on ΔI and current temperature T
            Choose a random number u between 0 and 1;
            if u < p then
                s ← s′
            end
        end
    end
end
T ← r · T ;                       // Cool down the temperature
return s*
```

Algorithm 1 takes as input an initial solution s, an initial temperature T_0, a cooling rate r, a plateau length L, a stopping criterion and a neighborhood structure, and returns the best found solution s^*. The iterative process involves generating a new solution s' in the neighborhood of s, computing the objective indicator difference ΔI between s' and s which can compare either the driver consistency indicator or the route balancing indicator, and accepting or rejecting the new solution based on the acceptance probability and the current temperature T. After a certain number of iterations, which is equal to the plateau length, the temperature is gradually decreased to enable the algorithm to converge towards a solution of higher quality.

The **stochastic nature** of the solution acceptance in SA can result in different solutions being obtained in various runs of the algorithm, which is a potential drawback. Therefore, to ensure confidence in the quality of the obtained solutions, it is necessary to run the algorithm multiple times. This issue is highlighted by Baugh Jr et al. (1998) in their study on meta-heuristics for solving DARPs.

As described in Algorithm 1, implementing a SA requires making some decisions regarding the initial solution method, the neighborhood structure, and various parameters such as the initial temperature, the cooling rate, the plateau length, and the stopping criterion. The following sections will delve into how these choices were made for the specific problem addressed in this thesis.

4.2 Choice of the initial solution method

In contrast to many other studies where the initial solution is typically generated randomly, this study adopts a different approach by using an **insertion heuristic** to construct the initial solution.

The primary objective of the initial solution creation method is to **prioritize optimizing driver consistency to the greatest feasible extent**, as it represents the higher-level objective. This objective is achieved by clustering requests based on the patients they belong to and examining patients individually. To maximize the likelihood of assigning all requests from each patient to a single driver, patients are considered in descending order based on the total number of requests they have throughout the entire time horizon. This sorting approach significantly increases the probability of efficiently allocating all transportation needs for each patient to a single driver, thus enhancing feasibility and improving overall efficiency. **The ultimate objective is therefore to allocate each patient and their associated requests to a single driver's route, considering all days in the schedule**. However, as there are usually more patients than drivers, some drivers may end up taking multiple groups of patients on their routes. Therefore, once all drivers already have the requests of a particular patient, drivers will be ranked in ascending order based on their total route duration over the entire time horizon. Then, they will be considered in this order, and the first driver on the list who can accommodate all of the patient's requests will take them all. **This approach takes route balancing into account to some extent**. If no single driver is capable of accommodating all of a patient's requests, the requests will be divided among multiple drivers' routes. Each request will thus be assigned to the first driver, in ascending order of total ride time, who can accommodate it. As the SA algorithm progresses, these initial allocations of patients to drivers' routes are expected to improve. Algorithm 2 illustrates this initial solution creation method.

With this method, the algorithm will always begin from the same solution since the order in which patients and drivers are considered is predetermined and fixed.

It is worth noting that the initial solution generated by this insertion heuristic is always feasible for the dataset under consideration (described in Section 5.1). This means that all requests can be successfully accommodated without violating any constraints. As a reminder, all constraints are considered to be hard, as described in Section 3.3, and are checked at each step of the solution construction. Furthermore, when a request can be inserted at multiple positions along a driver's route, the position that minimizes the total route duration is selected. This approach aims to reduce the total distance traveled to a certain extent.

Algorithm 2 Initial solution creation method.

Input : List of all patients p in P with all their respective requests p_r over the entire time horizon, available drivers D

Output: Initial routes on each day for all drivers in D

Order all patients in P in descending order based on the number of requests they have throughout the entire time horizon;

foreach *driver d in available drivers D* **do**

  Create empty routes for the vehicle on each day;

foreach *patient p in ordered patients P* **do**

  Rank drivers in ascending order of total ride times;

  foreach *driver d in ordered drivers* **do**

    if *driver d can accommodate all r in p_r over all days* **then**

      Assign all r in p_r to the routes of d over all days;

  if *all requests r in p_r have not been assigned to any driver* **then**

    foreach *request r in p_r* **do**

      foreach *driver d in ordered drivers* **do**

        if *driver d can accommodate the request r in its schedule on all days* **then**

          Assign r to the routes of d on all days;

        Rank again drivers in ascending order of total ride times;

In order to create **route schedules**, this study drew on insights from Paquay et al. (2020) and Mitrović-Minić and Laporte (2004) to compute the earliest and latest arrival and departure times for each location. If the initial depot is denoted by 0 and the final location by l, the scheduled arrival time A_i at location i can be any time between the earliest arrival time $\underline{A}_i$ and the latest arrival time $\overline{A}_i$. Similarly, the scheduled departure time D_i from any location i must fall between the earliest departure time $\underline{D}_i$ and the latest departure time $\overline{D}_i$. The earliest arrival time $\underline{A}_i$ (resp. the earliest departure time $\underline{D}_i$) represents the earliest time at which location i can be reached (resp. left), assuming that all preceding locations have been served at the earliest moment within their respective time windows. These earliest arrival and departure times are thus determined by performing a forward pass through the requests along the routes. Mathematically, it means that the following equations hold (equations 4.1 to 4.3, where t represents the time when the vehicle leaves the initial depot, $t_{i,j}$ denotes the time required to travel from location i to location j, a_i indicates the earliest arrival time requested by the patient at location i, and s_i demonstrates the service time required by the patient at location i):

$$\underline{A}_1 = t + t_{0,1}, \tag{4.1}$$

$$\underline{D}_i = \max\{a_i, \underline{A}_i\} + s_i, \forall i \in [1, ..., l], \tag{4.2}$$

$$\underline{A}_i = \underline{D}_{i-1} + t_{i-1,i}, \forall i \in [2, ..., l]. \tag{4.3}$$

The latest departure time $\overline{D}_i$ from location i (resp. the latest arrival time $\overline{A}_i$ to location i) is the latest possible moment at which the vehicle can leave location i (resp. arrive at location i) to be able to serve on time all locations following i. These values are computed by performing a backward pass through the requests along the assigned routes, meaning that the following equations hold (equations 4.4 to 4.6, where b_i denotes the latest arrival time requested by the patient at location i):

$$\overline{A}_l = b_l - s_l, \tag{4.4}$$
$$\overline{D}_i = \overline{A}_{i+1} - t_{i,i+1}, \forall i \in [1, ..., l-1], \tag{4.5}$$
$$\overline{A}_i = \min\{\overline{D}_i - s_i, b_i\}, \forall i \in [1, ..., l-1]. \tag{4.6}$$

During the initial solution creation process, as well as for all computations made in the study more generally, a **waiting strategy** was selected. Two main waiting strategies are discussed in Mitrović-Minić and Laporte (2004):

- **The drive-first waiting strategy** requires vehicles to drive as soon as it is feasible, leaving each location at the earliest possible departure time. This strategy is considered to be the most suitable for static VRP problems (as stated by Mitrović-Minić and Laporte (2004)).
- **The wait-first waiting strategy** requires vehicles to wait at their current location for as long as it is feasible, leaving each location at the latest possible departure time.

To ensure compliance with the maximum ride times of patients and maximum route duration of drivers, the **drive-first waiting strategy** is chosen in this study. This means that when a patient is loaded, a driver continues its route without any waiting at the location. However, to ensure compliance with the maximum route duration constraints for drivers and enable a later arrival at the first location, the **wait-first waiting strategy** is utilized when departing from the initial depot. This strategy involves waiting at the depot until the latest possible departure time is reached, thus aiming to reach the first location as late as possible. By adopting this strategy, the algorithm minimizes the overall time spent throughout the entire route.

4.3 Choice of the neighborhood structures

As a reminder, the goal is to optimize two objectives using a lexicographic objective function: one to evaluate driver consistency and the other to evaluate route balancing. In order to achieve this, the algorithm requires **at least two distinct neighborhood structures**. One should be designed to primarily optimize driver consistency, even at the expense of route balancing, while the other should aim to improve route balancing without compromising the achieved level of driver consistency.

Insights from Zidi et al. (2012), Mauri and Lorena (2006), and Braekers et al. (2014) suggest that there are several elementary local search operators that can be used to create neighbors of a current solution. These operators can be categorized into two types: **intra-route operators** and **inter-route operators**. Intra-route operators modify only one route in the neighborhood process, while inter-route operators modify multiple routes simultaneously. The following is a summary list of the most used local search operators:

1. **Intra-route operators**:
   - The *re-order route* operator: selects a location in a route, which can be either a pick-up or a delivery location, and changes its position to a new one in the same route, ensuring that a pick-up is always before its corresponding delivery;
   - The *k-opt* operator: removes k arcs from a route and tries to reconnect the remaining segments in all possible ways.

2. **Inter-route operators**:

   - The *relocate* operator: removes a request from its current route and tries to insert it into another vehicle route, at the best possible position;
   - The *exchange* operator: swaps two requests of two different routes or swaps the vehicles of two routes;
   - The *2-opt** operator: removes an arc from two routes and recombines the resulting parts, which means the first part of route one with the second part of route two, and vice versa.

In the context of this study, using intra-route operators may not be as effective in improving both driver consistency and route balancing. This is because it would still assign the same requests to the same routes, which would not reduce driver consistency since the same drivers would still serve the same requests. Additionally, it could alter the route balancing indicator since the total duration of a route would change. Since the initial positions of the requests are always determined to minimize the total route duration, altering the positions of some requests may only result in an increase in the total route duration without necessarily ensuring better route balancing. Therefore, **inter-route operators** will be used to reach the objectives of this study.

The ***relocate* operator** has been chosen among the inter-route operators. This choice is motivated by the fact that it has a higher potential to improve both driver consistency and route balancing at the same time, compared to the *exchange* operator. While the latter may improve these objectives in some cases, it may also worsen them in many situations. In contrast, the *relocate* operator allows requests to be moved from one route to another, potentially balancing the workload of drivers and reducing the overload of drivers with too many requests, while increasing the workload of drivers with fewer requests. By reallocating requests to different drivers, the operator can also improve driver consistency. The *2-opt** operator was also considered, but it was not selected because it may not improve driver consistency and route balancing as much as the *relocate* operator.

The *relocate* operator will be applied strategically to improve both driver consistency and route balancing indicators over the entire time horizon. This means that the operator will not be applied randomly. Instead, it will be used to **analyze the indicators over the entire time horizon** and **choose the requests to relocate and their new routes accordingly**. The approach will differ depending on the objective being taken into consideration, as explained in the next two subsections. For each neighborhood structure, a *destroy* and a *repair* operator will be defined. The former defines how the locations will be removed from a route, and the latter determines how these same locations will be reinserted into their new routes (as seen in Kovacs et al. (2015)).

4.3.1 Neighbor 1: focus on driver consistency

As a reminder of what was detailed in Section 3.4.1, the goal related to driver consistency is to minimize the average number of drivers that each patient sees throughout the entire time horizon. To improve this metric, it is important to pay attention to the extreme values, particularly the highest one, in order to reduce the mean value as much as possible.

The *destroy* operator identifies a **selected patient**, who is the patient that sees the largest number of drivers over the entire time horizon. Then, the operator selects an initial **selected driver** at random from the set of drivers that the selected patient sees over the time horizon. It then removes all the requests related to the selected patient on the daily roads of this selected driver.

A list of requests that need to be relocated to another driver's daily routes is generated as a result. It should be noted that all of these requests must be reassigned to the routes of **a single driver**, as assigning them to multiple drivers would often worsen driver consistency.

Choosing the appropriate driver to reassign the requests is critical, as it directly affects the driver consistency indicator. To prevent any counter-effects, it is crucial to select a driver that the selected patient already sees during the time horizon while avoiding the initially selected driver.

To achieve this, drivers are split into two groups. The first group includes **those seen by the selected patient during the time horizon, excluding the initially selected driver**. The second group consists of **all the others, including the initially selected driver**. The drivers in both groups are further **sorted based on their total route duration in ascending order**, which helps to consider route balancing to a certain extent.

The *repair* operator plays a crucial role in relocating the requests to a single driver while adhering to all constraints. The operator starts by examining the drivers in the first group. If any driver in this group can accommodate all the requests without violating any constraints, then **the requests are assigned to that driver**. However, if no driver in the first group can accommodate all the requests, the second group of drivers is considered in the same order, with requests being assigned to the first driver who can take them all, but **stopping when the initially selected driver is reached**. This is because assigning the requests to drivers with longer total route duration will not improve driver consistency or route balancing. However, assigning requests to a driver with a shorter total route duration can lead to better route balancing, even if it does not improve driver consistency. Therefore, drivers who are ranked after the initially selected drivers are not considered, while all drivers ranked before are taken into account. Furthermore, in any case, it will always be possible to insert the requests onto the initially selected driver's roads as they were already inserted there previously, ensuring feasibility, even if no improvements are made to either objective.

As in the initial solution creation method, during the request insertion, if a request can be added to a driver's road at multiple positions, the position that results in the minimum total route duration is chosen. Algorithm 3 describes these *destroy* and *repair* operators in order to create the neighborhood structure.

Since this neighborhood structure contains a **random factor**, which is related to the choice of the initially selected driver, the final algorithm will need multiple runs in order to ensure the quality of the final solution obtained.

Using this particular neighborhood structure, when starting from a current solution s with P patients and D drivers, the **size of the neighborhood** is determined by the **number of drivers that are initially assigned to the selected patient** in the current solution s. It should be emphasized that the selected patient will change throughout the process due to the redistribution of drivers among patients by the *relocate* operator. As a result, the total route duration of the drivers will also be altered. Consequentially, the order in which they will be considered for the next neighbors' computations will also change. This dynamic nature of the algorithm ensures that each iteration explores new possibilities by continually adapting the assignment of drivers and evaluating alternative configurations.

Algorithm 3 Neighbor structure focused on driver consistency.

Input : Current solution s, list of patients P, list of drivers D
Output: Neighbor solution s'
$s' \leftarrow s$;
Find the selected patient $p \in P$ who sees the largest number of drivers over the entire time horizon;
Choose the initial selected driver d_{init} at random from the set of drivers that patient p sees over the time horizon;
$R' \leftarrow \emptyset$;
for *each day* t **do**
    Remove all requests related to p on the daily routes of d_{init} and add them to R';
$D_1 \leftarrow$ drivers seen by p during the time horizon, excluding d_{init};
$D_2 \leftarrow D \setminus D_1$;
Sort drivers in D_1 and D_2 based on their total route duration in ascending order;
$assigned \leftarrow False$;
for *each driver* $d \in D_1$ **do**
    if d *can accommodate all the requests included in* R' **then**
        Add all the requests included in R' to the daily routes of d, ensuring that their insertion position result in the minimum possible total route duration;
        $assigned \leftarrow True$;
        break;
if $assigned = False$ **then**
    for *each driver* $d \in D_2$ *until* d_{init} *is reached* **do**
        if d *can accommodate all the requests included in* R' **then**
            Add all the requests included in R' to the daily routes of d, ensuring that their insertion position result in the minimum possible total route duration;
            $assigned \leftarrow True$;
            break;
return s'

4.3.2 Neighbor 2: focus on route balancing

The aim of the route balancing indicator, as described in Section 3.4.2, is to minimize the standard deviation among the total route duration of all drivers throughout the entire time horizon. To improve this metric, it is essential to target the extreme values, whether they are the highest or lowest, and make them converge towards a common value to minimize the standard deviation as much as possible.

In this case, the *destroy* operator works by identifying a **selected driver** who has the longest total route duration over the entire time horizon. Then, the operator randomly chooses a **selected request** from the set of requests that are serviced by the selected driver over the entire period of time. To account for driver consistency to a certain extent, if the selected request is an outbound travel, the corresponding inbound travel will also be chosen if it is on the selected driver's routes[1],

[1]In the current problem formulation, drivers are not explicitly required to handle both outbound and inbound requests from a single patient. However, it is important to emphasize that during the initial solution creation process, where driver consistency is prioritized, it is common for drivers to be assigned both outbound and inbound requests

and vice versa. The operator then removes these requests from all the daily routes of the selected driver if they are on the driver's routes over several days, constructing a list of daily requests to be replaced. These requests are then **to be reallocated together** to another driver's routes.

Since the workload of the most heavily loaded driver is decreased, the workload of the least occupied driver has to be increased in order to minimize the standard deviation. For this reason, **drivers are sorted in ascending order of their total route duration** over the entire time horizon. The *repair* operator will then evaluate the feasibility of inserting the requests onto the drivers' routes, starting from the driver at the top of the list and moving downwards, **until a driver capable of accommodating all the requests for the entire planning period is found**. As for the first neighborhood structure explained in the last section, a driver is considered capable of accommodating the requests only if all constraints are met, which ensures that the construction of the routes remains feasible. If none of the other drivers can accommodate the requests, they will be returned to the selected driver's routes, which will be placed last on the list. Each insertion is made in such a way that the total route duration is minimized, as in the initial solution creation method and the first neighborhood structure.

Algorithm 4 describes these *destroy* and *repair* operators in order to create this second neighborhood structure.

Algorithm 4 Neighbor structure focused on route balancing.

Input: Current solution s, list of drivers D
Output: Neighbor solution s'

Identify the selected driver d_s with the longest total route duration over the entire time horizon;
Randomly select a request r_s from the set of requests that are serviced by d_s over all days;
if *r_s is an outbound travel* **then**
  | Select the corresponding inbound travel, $r_{s'}$, if it is on d_s's routes;
end
if *r_s is an inbound travel* **then**
  | Select the corresponding outbound travel, $r_{s'}$, if it is on d_s's routes;
end
Remove r_s and $r_{s'}$ (if applicable) from all daily routes of d_s to construct a list of daily requests to be replaced, L;
Sort all drivers in D by ascendant total route duration, ensuring that d_s is at the last position;
for *each d in ordered drivers* **do**
  if *d can accommodate all r in L* **then**
    Create a new solution s' by inserting all r in L into d's daily routes in a manner that minimizes the total duration of each route;
  end
end
return s'

Similarly to the neighborhood structure detailed in Section 4.3.1, this one contains a **random factor** which is related to the choice of the request to remove from the selected driver's routes. Thus, the final algorithm will need multiple runs in order to ensure the quality of the final solution obtained.

---

from the same patient.

In this neighborhood structure, the **size of the neighborhood** for a current solution s with D drivers is determined by the **number of requests served by the selected driver throughout the planning horizon**. Specifically, **it is influenced by the count of pairs of outbound and inbound requests associated with the same patient in the solution** s. Since the workload of drivers is redistributed in a different way through the *relocate* operator, the algorithm is able to explore a large number of possibilities by changing the selected driver during the process.

Initially, an alternative approach was considered for the *destroy* operator, which involved removing all request types related to a specific patient from the selected driver's routes, rather than just one single request type repetition over the total number of days. This was done to ensure driver consistency since the same patient can have different request types over the time horizon. However, after multiple tests, it became apparent that this approach was too restrictive and constrained the algorithm to a too small number of possibilities, resulting in very limited improvement. Indeed, proceeding this way resulted in a higher total number of requests that needed to be accommodated by a new driver, making it difficult to find a feasible solution. Therefore, this approach was abandoned, and the current neighborhood structure was chosen instead.

4.3.3 Integration of both neighborhood structures

In practice, the proposed approach involves **running two successive iterations of the SA algorithm**. The first SA algorithm starts from the initial solution described in Section 4.2 and will utilize the neighborhood structure that prioritizes the improvement of driver consistency described in Section 4.3.1. The resulting solution from the first SA algorithm is an intermediary solution that will then be used as the initial solution for the second SA algorithm. The second SA algorithm employs the second neighborhood structure that focuses on enhancing route balancing, described in Section 4.3.2, without allowing any deterioration in the best level of driver consistency achieved in the first SA. This ensures that the driver consistency achieved during the first iteration is maintained while seeking to improve the overall balance of routes. This process is illustrated in Figure 4.1.

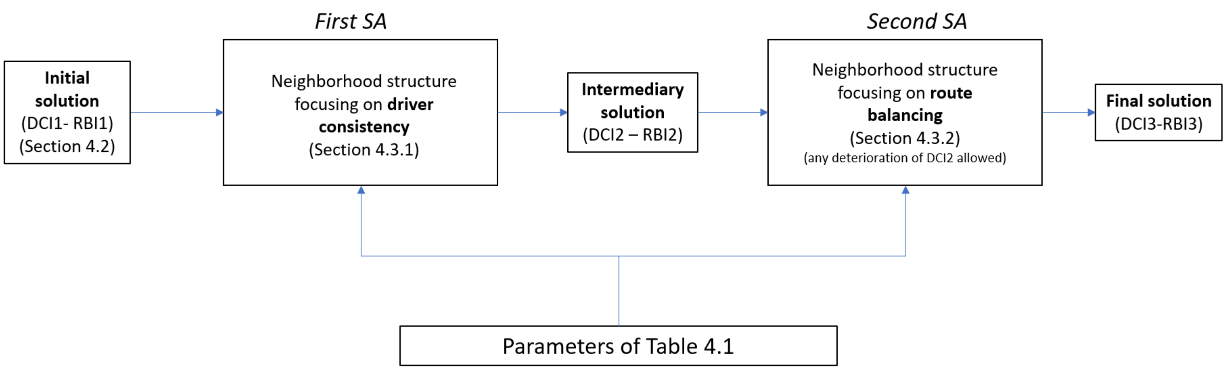

Figure 4.1: Integration of both neighborhood structures in two successive iterations of the Simulated Annealing algorithm. "DCI" stands for Driver Consistency Indicator, while "RBI" stands for Route Balancing Indicator.

4.4 Details about the insertion and deletion operators

The initial solution method and the neighborhood structures defined for this study involve two main operations on the routes: **inserting a request into a route**, and **deleting a request from a route**. These operations are performed during the SA algorithm to obtain the best solution. The subtleties and differences in the results come from the order in which the requests are considered for insertion or deletion, as well as the order in which the drivers are considered. The following two sections will provide detailed information on how these operators are implemented in the code and the key steps involved in inserting or deleting a request from a route.

4.4.1 Inserting a request into a route

To insert a request into a route, the first step is to **test the feasibility of inserting the pick-up location**. Once all the possible positions to insert the pick-up location are found and if at least one exists, each possible pick-up insertion is considered. **All the possibilities for the corresponding delivery location insertion** are then computed, considering only positions after the inserted pick-up location. This ensures that the pick-up and the delivery of a request are always satisfied by the same driver and that the pick-up location is visited before the delivery location. In practice, regardless of whether the considered location to insert is a pick-up or a delivery, the process to find the most suitable insertion position is the same.

To determine feasible insertion positions for pick-up or delivery locations within a given route, the system initially computes a ***shift*** value. This shift value represents **the additional time required to serve the new patient at a specific pick-up or delivery location**, compared to the scenario where the location is not inserted into the route. The shift value serves as a metric for evaluating the impact of inserting a pick-up or delivery location at a specific position within the route and helps identify the best insertion point. This shift is computed as follows:

$$shift_i = t_{i-1,newloc} + servicetime_{newloc} + t_{newloc,i+1}$$

Here, $i-1$ and $i+1$ represent respectively the locations immediately preceding and following the insertion point. $t_{i,j}$ represents the travel time between locations i and j, while $servicetime_i$ represents the time needed to serve the patient at location i. The resulting shift value, $shift_i$, represents the amount of extra time needed to serve the patient at its pick-up or delivery location if it is inserted at position i on the route.

Once the $shift$ is computed, an **insertion position is tested only if all subsequent locations have slack times greater than or equal to this $shift$.** As a reminder, the slack time of a location refers to the time difference between the latest possible departure time and the earliest possible departure time from that location. It indicates the flexibility available at a specific location, allowing for adjustments and accommodating additional requests without violating time window constraints. This initial filtering step reduces the number of positions that need to be considered and makes the algorithm more efficient.

However, the identified insertion possibilities are not guaranteed to be feasible at this stage since only the time window constraints are taken into consideration. Other constraints must be verified, such as the vehicle capacity constraints, the maximum route duration for the drivers, and the maximum ride times for each user, among others (cited in Section 3.3). Therefore, for each insertion possibility, all of the constraints are checked, and if an insertion is not feasible, it will be discarded. On the other hand, if an insertion is feasible, the new total route duration will be calculated and recorded. After trying all the possible insertion positions, **the one that results in the minimum total route duration will be selected as the optimal insertion position**.

After an insertion is performed, the schedule of the route is adjusted accordingly. This includes updating the earliest and latest arrival and departure times, as well as slack times, to account for the new insertion. Additionally, the effective arrival and departure times at each location are updated, still based on the **drive-first strategy** described in Section 4.2. The starting time of service at each location is also updated accordingly. Finally, the **loads** of all resource types are also updated between the pick-up and delivery insertions.

While most studies on the DARP aim to minimize total distance or route duration, this study prioritizes driver consistency and route balancing over distance or duration. Nonetheless, this insertion strategy still ensures that the resulting routes have relatively short distances. Moreover, given that the constraints of maximum route duration for drivers and maximum ride times for users are always satisfied without any violation, it can be guaranteed that the total distance traveled or total route duration will never violate the preferences of the patients and drivers.

4.4.2 Removing a request from a route

Removing a request from a route is less challenging than the insertion process because **no feasibility check needs to be performed**. If a route is feasible with a given set of requests, it will still be feasible with a removed request. Removing a request will shorten the total route duration and increase slack times, making the route more flexible for future insertions. The only necessary adjustment is to **update the schedule and loads at each location** to reflect the removal of the request.

4.5 Choice of parameters for the SA implementation

This section aims to define the **initial temperature** T_0, **the cooling rate** α, the **plateau length** L and the **stopping criterion**, controlled by two parameters, which are K and ϵ. Before discussing the chosen parameters for this study, here is a brief overview of the role of each parameter (Pirlot (1996) and Baugh Jr et al. (1998)):

- The **temperature function** regulates the likelihood of accepting bad moves during the search process by controlling the degree of randomness. This function is defined as a non-increasing function of time, and it allows for the computation of the probability p ($p = e^{-\frac{\triangle C}{T}}$, where $\triangle C$ represents the difference in the objective function value), which represents the **probability of accepting a solution that worsens the objective function**. The temperature function is characterized by an **initial temperature** T_0, defined by an **initial acceptance probability** p_0, and a **cooling rate** α. The former represents the starting temperature of the algorithm and should be defined in a way that almost all transitions are accepted at the beginning of the optimization process. The latter determines the rate of temperature decrease throughout the algorithm and typically ranges between 0,8 and 0,99.

- The **plateau length** L is equal to the number of iterations during which the temperature is maintained constant before being decreased. After L iterations, $T = \alpha \times T$. This number of iterations L should be big enough in order to explore most of the neighboring solutions at each temperature level and to reach quasi-equilibrium, and should thus be based on the neighborhood size.

- The **stopping criterion** is determined by a **number K of plateau lengths L** to consider before deciding whether to continue the algorithm or not. The work of Pirlot (1996) integrates two different stopping criteria once the algorithm reaches $K \times L$ iterations:
  - The algorithm continues if the proportion of accepted moves during the preceding $K_1 \times L$ iterations is greater than or equal to $\epsilon_1 \times L$;
  - If the best solution found so far has not been improved by ϵ_2% after K_2 consecutive sets of L iterations, the algorithm stops.

  The research conducted by Baugh Jr et al. (1998) and Braekers and Kovacs (2016) specifically use the first stopping criterion, where K is set to 3. In Baugh Jr et al. (1998), ϵ_1 is assigned a value of 0, while in Braekers and Kovacs (2016), ϵ_1 is set to 5%.

To ensure the algorithm's effectiveness in exploring a broader range of solutions and avoiding being stuck within a limited solution space, **a combination of the two stopping criteria will be integrated into this study**. By utilizing both criteria, the algorithm will be able to make informed decisions on whether to continue or terminate based on different aspects of its performance. This approach aims to strike a balance between accepting sufficient moves and seeking substantial improvements in the best solution found so far.

The choice of parameters for a SA algorithm greatly affects its success, as they define the entire shape of the optimization process. The set of parameters selected for this study is inspired by Baugh Jr et al. (1998) and Braekers and Kovacs (2016). They are presented in Table 4.1.

α	T_0	L	K_1/K_2	ϵ_1 and ϵ_2
0,8	$-\frac{\triangle C}{\log 0{,}5}$	D	1	$\epsilon_1 = 5\%$, $\epsilon_2 = 0\%$

Table 4.1: Set of parameters used.

In this table, the variable D represents the total number of available drivers. Initially, other parameters were tested such as $\alpha = 0{,}99$, $K_1 = K_2 = 3$, $L = D \times R$ (where R is the total number of different request types across all patients), and $p_0 = 0{,}9$. However, it was observed that the algorithm took a significantly long time to run without any improvement in the objective function. To address this issue, the values of α, L, and p_0 were reduced, which resulted in a substantial reduction in the running time without any degradation in the quality of the obtained solutions. Similarly, the value of ϵ_2 was set to 0 in order to stop the algorithm as soon as any improvement is made after $K_2 \times L$ iterations.

4.6 Solution representation

The final solutions obtained for the problem are a series of T daily solutions (T represents the number of working days considered in the planning period), with each daily solution containing D routes, where D is the number of available drivers. A specific driver's **route** for a given day is represented by a **vector of vectors**, and each sub-vector represents a **specific location** that is visited. These sub-vectors contain relevant information about the visited location, such as:

- The **location** that is visited (represented by a number, 0 representing the depot);
- The **patient** served at that location (represented by a number);
- The **earliest and latest arrival times** at the location;
- The **earliest and latest departure times** from the location;
- The **slack time** at each location, which corresponds to the difference between the latest departure time and the earliest departure time;
- The **effective arrival time** at the location, which has to be between the earliest and the latest arrival times;
- The **effective departure time** from the location, which has to be between the earliest and the latest departure times;
- The **start time of service**, which can be after the effective arrival time at the location and depends on the time windows specified by the patient;
- The **service time**, which represents the time needed to load or unload the patient at the given location;
- The **load of each resource type** in the vehicle after having visited the location;
- Whether the location is a **pick-up or delivery point** (respectively 1 or 2).

This information is stored in a **vector of size 17x1**. Locations, patients, and requests are assigned unique identification numbers, enabling their accurate identification and representation. Time is represented using numbers from 0 to 720, which correspond to the available minutes in a 12-hour working day. Therefore, all vectors have the following structure:

(Stop number, Location number, Patient number, Earliest arrival time, Effective arrival time, Latest arrival time, Starting time of service, Service time, Earliest departure time, Effective departure time, Latest departure time, Load of resource type A[2]*, Load of resource type B, Load of resource type C, Number of the associated request, Pick-up (1) or Delivery (2) location)*

Here is a numerical example of a route representation:

((1,0,0,0,0,0,0,0,0,185,185,185,0,0,0,0,0)
(2,1,1,15,200,200,200,5,20,205,205,185,2,1,0,1,1)
(3,2,1,34,219,219,219,5,209,224,389,180,0,0,0,1,2)
(4,2,1,209,224,389,374,5,379,379,685,306,2,1,0,2,1)
(5,1,1,394,394,700,394,5,399,398,705,306,0,0,0,2,2)
(6,0,0,414,414,720,0,0,0,0,0,360,0,0,0,0,0))

This example involves a route that includes 6 locations, which serves two requests from the patient n°1. The route includes both outbound (request n°1) and inbound (request n°2) trips, starting from location 1 (which represents the home location) and going to delivery location 2, before returning to location 1. These request types consist of one seated patient accompanied by one person, resulting in a service time of 5 minutes. The depot is represented by the first and last lines of the route, with

[2] As it will be explained in Section 5.1, the problem under consideration contains 3 resource types.

the identification number equal to 0. Thus, the starting depot (stop number 1) is provided only with information regarding departure times, while the final depot (stop number 6) is provided only with information regarding arrival times.

5 Computational experiments and results

In this chapter, the dataset used in this study is explained, as well as the implementation details. Finally, the final results obtained are presented and analyzed.

5.1 Dataset

To implement the model presented in this study, the **benchmark dataset developed by Braekers and Kovacs (2016) was utilized**[1]. This choice was made after a thorough analysis of the existing literature, which revealed that the benchmark instances proposed by Braekers and Kovacs (2016) are the most relevant for the problem at hand. This is because all the other studies in the literature have focused on static versions of the single-period DARP, meaning without considering the multi-periodicity of the problem. In contrast, the instances developed by Braekers and Kovacs (2016) include a realistic set of requests and drivers over a **five-day time horizon**, making them well-suited for the purposes of this study.

This dataset originally comprised 432 instances of the problem, with each instance solved for three levels of **driver consistency** (the number of drivers seen per patient over the time horizon is bounded by the authors to respectively 1, 2 and 3). Therefore, a total of 1296 instances were considered. To ensure maximum diversity in the dataset, Braekers and Kovacs (2016) used four characteristics for constructing the instances:

- ***h(0)*** for a *heterogeneous fleet of vehicles* or ***h(1)*** for a *homogeneous fleet*;
- ***t(0)*** for *each user having only one request type over the week* (that can be repeated) or ***t(1)*** for *each user being able to have up to 2 different request types per week* (with a probability of 20%);
- ***f(0)*** for *low frequency regarding trips of the same type*, ***f(1)*** for *medium frequency*, and ***f(2)*** for *high frequency*;
- ***c(0)*** for *randomly distributed locations*, ***c(1)*** for *randomly distributed home locations but clustered destination locations*, and ***c(2)*** for *both clustered home and destination locations* (these clusters levels are represented in Figure 5.1).

Each instance has thus a possible combination of all these characteristics. Moreover, the dataset contains **12 levels in terms of the number of users**, ranging from 5 to 50 by steps of 5, with 75 and 100 additionally.

[1] Available at `http://alpha.uhasselt.be/kris.braekers/`.

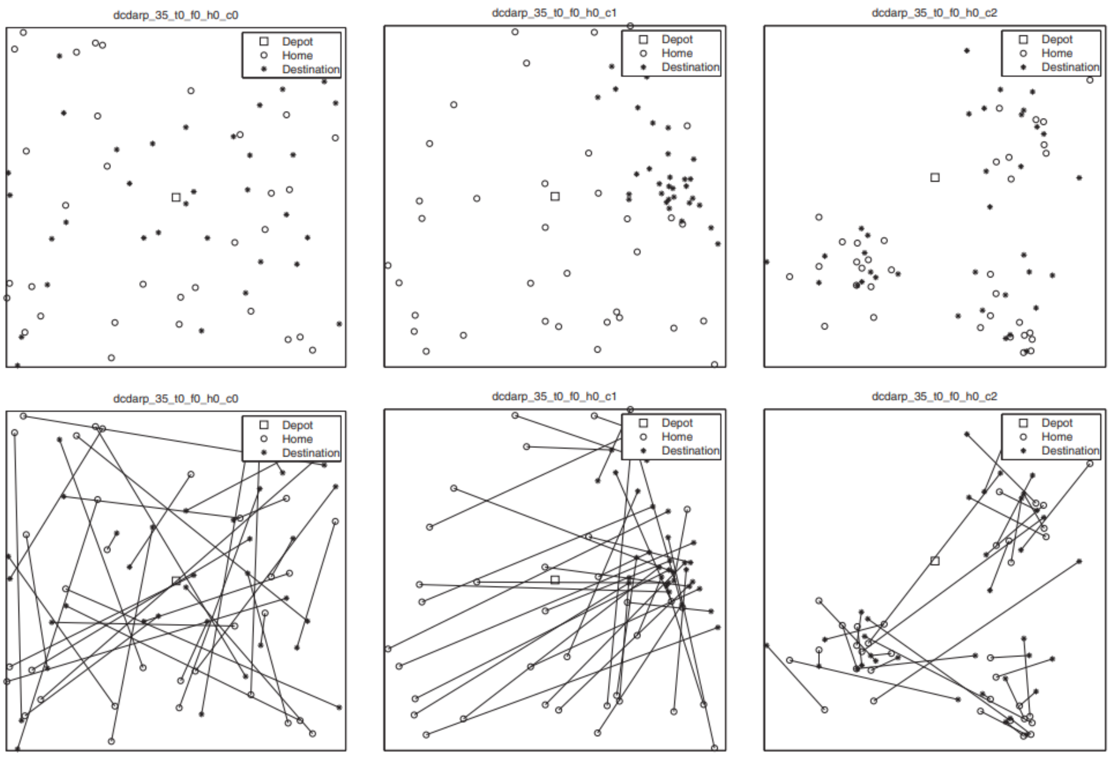

Figure 5.1: The top figures display the pickup and drop-off locations, while the bottom figures show how the locations are paired. The left, center, and right images correspond to the *c(0)*, *c(1)* and *c(2)* hypotheses, respectively (Braekers and Kovacs (2016)).

To narrow down the scope of the study, only **24 instances out of the 432 available were selected for implementation**. Initially, all instances falling under ***h(0)*** assumption were eliminated, leaving only instances with a heterogeneous fleet of vehicles. Next, to ensure a representative sample, two instances were chosen from each of the 12 user number levels. For each user number level, one instance followed the ***t(0)*** assumption, while the other followed the ***t(1)*** assumption. The selection was further refined by choosing instances that represented a mix of the 3 levels of frequency and clustering, as shown in Table 5.1. The names of the 24 final chosen instances can be found in Appendix A. The instances were initially downloaded in TXT format and subsequently converted into the XLSX format for easier manipulation.

	8 inst. under c(0)	**8 inst. under c(1)**	**8 inst. under c(2)**
8 inst. under f(0)	3 inst. under f(0)-c(0)	3 inst. under f(0)-c(1)	2 inst. under f(0)-c(2)
8 inst. under f(1)	3 inst. under f(1)-c(0)	2 inst. under f(1)-c(1)	3 inst. under f(1)-c(2)
8 inst. under f(2)	2 inst. under f(2)-c(0)	3 inst. under f(2)-c(1)	3 inst. under f(2)-c(2)

Table 5.1: Instances selection based on frequency and clustering levels.

Here are the parameters defined in the dataset by Braekers and Kovacs (2016):

- The **time horizon** considers **5 days**;
- **Working days** are considered to last **12 hours** (from 8 a.m. to 8 p.m.);
- All the locations are confined in a **region of 20 km^2**;

- There is a **single depot located at the centre** of the region;
- The **number of available vehicles** is imposed and is considered sufficient to serve all requests;
- The **maximum route duration** for each driver is equal to **8 hours**;
- All vehicles have an **average speed of 40km/h** and it is assumed that drivers go from one location to another in a straight line;
- **Time windows of 15 minutes** are imposed on all the delivery locations for outbound requests, and on the pickup location for inbound requests;
- **Maximum user ride time** for a request is **two times the direct ride time**, which corresponds to the direct travel time between the pick-up and the delivery locations, **with a minimum of 20 minutes** to ensure efficient planning flexibility;
- Each user has a **home location**;
- Each user has a **capacity requirement for each resource type**, and the available resource types are regular users and users transported in a wheelchair. Additionally, the problem also takes into account accompanying people;
- Each user specifies their **requests types and their frequency of occurrence** throughout the week;
- Each user has a **service time of 5 minutes** for regular users and **10 minutes** for users using a wheelchair.

An additional clarification is needed regarding the different **resource types** mentioned in the article. Although the use of a stretcher was mentioned as a possibility, it was excluded for simplification purposes as it would have primarily impacted the feasibility of the routes rather than the driver consistency or route balancing assessment. Thus, this study deals with three types of resources, each requiring a specific number to be fulfilled by each patient:

1. **Type A** resources which include **both staff or patient seats**;
2. **Type B** resources which include **patient seats only** (count included in resource type A);
3. **Type C** resources which are dedicated to accommodating **wheelchairs**.

For instance, if a patient's request specifies a capacity requirement of 2 for Type A, 1 for Type B, and 1 for Type C, it implies that the patient requires two seats, one for a patient (since the capacity requirement for Type B is equal to 1) and the other for an accompanying person, as well as space for a wheelchair in the vehicle.

5.2 Implementation details

The SA algorithm was implemented in **Julia Programming Language (Version 1.7.2)**, which is a high-level, dynamic programming language designed for numerical and scientific computing, data analysis, and visualization. All the computations were carried out on a personal laptop (*Acer Aspire A515-55G*) with an *Intel Core i5-1035G1* processor running at 1.00 gigahertz, with 8.0 gigabytes of RAM, and a *64-bit Windows 10* operating system. To ensure fair and consistent computational results, no other applications were running in the background while the instances were being solved and all instances were solved under the same conditions.

5.3 Initial solution

Even before the improvement phase is applied, the creation of the initial solution process can offer valuable insights and enable analysis.

It should be emphasized that the method used imposes various conditions. Firstly, all constraints are considered hard, as explained in Section 3.3. Secondly, requests are systematically processed in a predetermined order, prioritizing patients based on the number of requests they have over the entire time horizon. Similarly, drivers who handle each request are selected in a specific order based on their accumulated ride time over the entire time horizon. Consequently, this insertion heuristic does not incorporate randomization or exploration of alternative possibilities. This characteristic implies that **the algorithm will always start from the same initial solution**, which restricts the solution space and may limit the potential for finding better solutions.

Table 5.2 presents the values of both the Driver Consistency Indicator (DCI) and Route Balancing Indicator (RBI) for the initial solutions of each instance. Additionally, this table not only indicates the average number of drivers seen by each patient represented by the DCI but also highlights the minimum and the maximum number of drivers encountered by any individual patient in Column 3. Furthermore, the DCI is compared to the mean number of requests per patient over the entire time horizon.

It is noteworthy that **the employed method yields a low initial level of the DCI** across all instances. Specifically, the indicator is consistently lower than the average number of requests per patient over the entire time horizon and always ranges between 1 and 2. Upon closer analysis of the obtained values, it becomes evident that a significant majority of patients, ranging from 72% in instance 20 to 100% in instance 2, already have the opportunity to be loaded by only one driver throughout the entire time horizon. However, there is a small proportion of patients who encounter a relatively large number of drivers. In most cases, only one patient experiences the maximum number of drivers, while there are a few instances where multiple patients share this maximum. Still, when considering the overall ratio relative to the total number of patients, the occurrence of reaching the maximum number of drivers remains minimal. It is expected that the SA algorithm will reduce this maximum number since the objective is to minimize the mean number of drivers required.

Instance	DCI	Min. - Max. drivers seen	RBI	Mean nb. of requests/cust.
1	1,2	1 *(80%)* - 2 *(20%)*	48,98	3,6
2	1	1 *(100%)*	53,13	4,4
3	1,2	1 *(90%)* - 3 *(10%)*	131,03	7,8
4	1,3	1 *(80%)* - 3 *(10%)*	40,83	3,8
5	1,07	1 *(93,3%)* - 2 *(6,7%)*	94,15	4,8
6	1,47	1 *(86,7%)* - 5 *(6,7%)*	162,07	6,3
7	1,4	1 *(85%)* - 4 *(10%)*	130,16	8,4
8	1,2	1 *(85%)* - 3 *(10%)*	109,38	6,5
9	1,12	1 *(88%)* - 2 *(12%)*	232,17	4
10	1,48	1 *(80%)* - 5 *(4%)*	105,69	7,44
11	1,67	1 *(83,3%)* - 7 *(3,3%)*	115,54	8,6
12	1,13	1 *(86,7%)* - 2 *(13,3%)*	229,26	4,8
13	1,54	1 *(77,1%)* - 6 *(2,8%)*	105,49	6,3
14	1,54	1 *(74,3%)* - 6 *(2,8%)*	227,26	9,31
15	1,58	1 *(80%)* - 6 *(5%)*	170,48	8,1
16	1,9	1 *(62,5%)* - 6 *(2,5%)*	89,32	6,45
17	1,29	1 *(73,3%)* - 3 *(2,2%)*	76,29	3,5
18	1,29	1 *(91,1%)* - 5 *(6,7%)*	104,90	8,62
19	1,54	1 *(78%)* - 6 *(2%)*	100,51	6,28
20	1,48	1 *(72%)* - 4 *(8%)*	102,83	4,32
21	1,33	1 *(80%)* - 8 *(1,3%)*	122,36	3,71
22	1,56	1 *(81,3%)* - 7 *(1,3%)*	193,31	8,3
23	1,06	1 *(98%)* - 4 *(2%)*	317,64	8,2
24	1,35	1 *(84%)* - 4 *(7%)*	183,61	6,22

Table 5.2: Indicators for the initial solutions. In this table, Column 1 represents the instance, while Column 2 (resp. Column 4) displays the DCI (resp. the RBI) of each initial solution. Additionally, Column 3 includes the minimum and maximum number of drivers seen per patient in the initial solution, along with the corresponding frequency of encountering this specific number of drivers. The mean number of requests per patient in the instance is shown in Column 5.

5.4 Results obtained

In this section, the final results obtained are presented and analyzed.

As previously mentioned, the initial solution method does not involve random factors and does not lead to multiple possibilities. However, both neighborhood structures defined in Sections 4.3.1 and 4.3.2 utilize random factors. In the first neighborhood structure, which focuses on driver consistency, the random factor is linked to the number of available drivers, while in the second neighborhood structure focusing on route balancing, randomness is based on the number of unique request types. To fully explore the solution space and account for the random factors in each neighborhood structure, multiple runs of the SA algorithm are necessary. In this study, **each instance has been run 8 times**. The selection of this number is not based on any specific criteria or requirement but rather chosen in a way that aims to find a middle ground between exploring various potential solutions and minimizing the need for extensive computational resources.

Table 5.3 presents the best-achieved value of the DCI as well as the average results obtained for both DCI and RBI over the 8 runs for each instance. Additionally, it includes the average running time observed among all conducted runs, as well as the standard deviation of these times. It is worth reminding that the instances were solved using a lexicographic objective function, with driver consistency being prioritized and route balancing considered as a secondary objective. Consequently, among all the solutions obtained for each instance, the best solution selected was the one with the lowest DCI. In cases where multiple solutions had the same DCI, the one with the smallest RBI was chosen. These best outcomes are compared to their respective initial values for reference. The cases where the best value found of an indicator represents an improvement compared to the initial solution appear in bold. The detailed results from all runs for each instance can be found in Appendix B.

Inst.	**Init. DCI**	**Best - Mean DCI**	**Init. RBI**	**New - Mean RBI**	**Mean - Std. dev. of time (sec.)**
1	1,2	**1** - 1,1	48,98	58,91 - 92,20	0,06 - 0,03
2	1	1 - 1	53,13	53,13 - 53,13	0,04 - 0,01
3	1,2	**1,1** - 1,1	131,03	159,02 - 159,02	0,82 - 0,21
4	1,3	**1,2**- 1,2	40,83	40,83 - 147,25	0,43 - 0,21
5	1,07	**1** - 1	94,15	121,63 - 121,63	0,55 - 0,12
6	1,47	**1,4** - 1,4	162,07	**157,72** - 185,42	2,00 - 0,17
7	1,4	**1,35** - 1,35	130,16	137,35 - 144,78	4,90 - 0,74
8	1,2	**1,15** - 1,15	109,38	127,65 - 128,55	2,15 - 0,52
9	1,12	1,12 - 1,12	232,17	232,17 - 232,17	2,09 - 0,74
10	1,48	**1,44** - 1,44	105,69	**98,92** - 120,37	4,42 - 0,07
11	1,67	**1,63** - 1,63	115,54	**112,84** - 150,53	9,62 - 1,86
12	1,13	1,13 - 1,13	229,26	**151,62** - 151,62	5,13 - 0,96
13	1,54	**1,51** - 1,51	105,49	**100,69** - 128,60	6,50 - 1,44
14	1,54	**1,51** - 1,51	227,26	**145,40** - 145,40	14,55 - 4,66
15	1,58	**1,55** - 1,55	170,48	**133,40** - 133,40	15,19 - 1,46
16	1,90	**1,88** - 1,88	89,32	**85,02** - 88,78	17,53 - 3,72
17	1,29	**1,27** - 1,27	76,29	**62,76** - 62,85	5,09 - 0,71
18	1,29	**1,27** - 1,27	104,90	**99,37** - 103,28	18,01 - 2,21
19	1,54	**1,52** - 1,52	100,51	**99,22** - 102,41	28,24 - 2,92
20	1,48	**1,46** - 1,46	102,83	107,63 - 122,64	13,92 - 0,78
21	1,33	**1,31** - 1,31	122,36	128,09 - 128,09	7,53 - 0,72
22	1,56	**1,55** - 1,55	193,31	**155,28** - 155,28	107,40 - 9,53
23	1,06	**1,05** - 1,05	317,64	**316,58** - 317,16	49,35 - 4,35
24	1,35	**1,33** - 1,33	183,61	186,55 - 186,55	22,81 - 2,78

Table 5.3: Results obtained. Column 1 represents the considered instance. Columns 2 and 4 contain the initial indicators obtained from the initial solution. Column 3 displays the minimum DCI achieved, along with the mean DCI across all runs. Similarly, Column 5 contains the new RBI included in the solution that achieves the best DCI value mentioned in Column 3. The last column illustrates the average running times across all runs, as well as the standard deviation of these times.

5.4.1 Exploration of the solution space

An initial observation from the 8 runs conducted for each instance reveals that **all runs yielded consistent levels of DCI** (with the exception of the first instance, where 50% of the runs yielded a DCI of 1, while the remaining 50% resulted in a DCI of 1,2). This is evident from the fact that the best DCI obtained in each instance is always equal to the mean DCI across all runs. In contrast, **when considering the RBI, it appears that the same value is observed in 45,8% of the instances**. In the remaining 54,2% of instances, the comparison between the new RBI to the mean of all RBI values from all runs indicates that there is relatively low variability among the obtained RBI values, except for instance 4. These findings reveal that the selected parameters and neighborhood structures consistently resulted in solutions with similar values for both indicators, despite the inherent randomness of the algorithm. This observation suggests that the algorithm extensively explores the entire available solution space multiple times.

Furthermore, upon examining the moves performed during the algorithm runs for instances where the RBI varies, it is observed that **the SA algorithm explores a restricted set of feasible solutions** due to the chosen neighborhood structures. As a result, in each iteration, the algorithm frequently revisits previously encountered solutions. This repetitive exploration allows certain parameters to be significantly reduced, as discussed in Section 4.5.

5.4.2 Comparison with the initial solutions

Based on the comparison of the best values obtained with the initial values of DCI and RBI for all instances, it can be observed that **the DCI does not show any significant decrease during the improvement phase, and does not even change for 12,5% of the instances**. The maximum improvement achieved is only 0,2 in instance 1, and then 0,1 in instances 3 and 4. In the remaining instances, the improvement is lower than 0,1. This observation suggests that the potential improvement in DCI is limited by the method employed to generate the initial solution, as it significantly impacts the starting point of the optimization process. However, it is important to note that this starting point already optimizes the DCI to a certain extent by attempting to assign each patient to a single driver. The subsequent SA algorithm may not have the capacity to substantially enhance this indicator.

Table 5.4 provides a comparison between the initial solution and the best solution found regarding the minimum and the maximum number of drivers assigned to a single patient. It also includes the frequency of occurrence for these values among all patients in each instance. Any decrease in the minimum or maximum number of drivers compared to the initial solution or an increase in the frequency of the minimum number of drivers is highlighted in bold.

Compared to the initial solution, the analysis reveals that **more patients are served by only one driver throughout the entire time horizon**. However, it is worth noting that **even though the maximum number of drivers assigned to patients decreases in most cases, it remains relatively high (meaning greater than 2) in over two-thirds of the instances**. Among these instances, the maximum number of drivers is observed for three patients in 18,7% of cases, for two patients in 25% of cases, and for only one patient in the remaining 56,3% of cases. This highlights that **although the majority of situations involve a single patient, this high maximum number of drivers cannot be ignored**. A comprehensive distribution of all patients with their respective number of encountered drivers, including the full distribution for both the initial solutions and the best solutions found, can be found in Appendix C.

Instance	Min. - Max. - Initial solution	Min. - Max. - Best solution
1	1 (80%) - 2 (20%)	**1 (100%)**
2	1 (100%)	1 (100%)
3	1 (90%) - 3 (10%)	1 (90%) - **2 (10%)**
4	1 (80%) - 3 (10%)	1 (80%) - **2 (20%)**
5	1 (93,3%) - 2 (6,7%)	**1 (100%)**
6	1 (86,7%) - 5 (6,7%)	1 (86,7%) - **4 (13,3%)**
7	1 (85%) - 4 (10%)	1 (85%) - **4 (5%)**
8	1 (85%) - 3 (10%)	1 (85%) - **2 (15%)**
9	1 (88%) - 2 (12%)	1 (88%) - 2 (12%)
10	1 (80%) - 5 (4%)	1 (80%) - **4 (4%)**
11	1 (83,3%) - 7 (3,3%)	1 (83,3%) - **6 (10%)**
12	1 (86,7%) - 2 (13,3%)	1 (86,7%) - 2 (13,3%)
13	1 (77,1%) - 6 (2,8%)	1 (77,1%) - **5 (2,8%)**
14	1 (74,3%) - 6 (2,8%)	**1 (77,1%)** - 6 (2,8%)
15	1 (80%) - 6 (5%)	**1 (85%)** - 6 (5%)
16	1 (62,5%) - 6 (2,5%)	1 (62,5%) - **5 (5%)**
17	1 (73,3%) - 3 (2,2%)	1 (73,3%) - 3 (2,2%)
18	1 (91,1%) - 5 (6,7%)	1 (91,1%) - **5 (4,4%)**
19	1 (78%) - 6 (2%)	1 (78%) - **5 (2%)**
20	1 (72%) - 4 (8%)	1 (72%) - **4 (6%)**
21	1 (80%) - 8 (1,3%)	**1 (81,3%) - 7 (1,3%)**
22	1 (81,3%) - 7 (1,3%)	1 (81,3%) - 7 (1,3%)
23	1 (98%) - 4 (2%)	1 (98%) - **4 (1%)**
24	1 (84%) - 4 (7%)	**1 (86%) - 4 (6%)**

Table 5.4: Column 2 (resp. Column 3) shows the minimum and maximum number of drivers assigned to patients in the initial solution (resp. best solution found), with the frequency of occurrence for each of these values among all patients.

On the other hand, the analysis reveals that the RBI demonstrates varying patterns of improvement. Specifically, **the RBI is enhanced in 54,2% of the instances, worsened in 33,3% of the instances, and remains unchanged in the remaining 12,5%**.

In cases where the DCI improves but the RBI worsens, the decision-maker faces a choice between the routes established in the initial solution, which have a lower RBI but a higher DCI and the routes obtained through the SA algorithm, which have a lower DCI but a higher RBI. This situation occurs for 33,3% of the instances, as indicated in Table 5.5. When considering the number of available drivers in these instances, it can be observed that the increase in RBI is relatively minor compared to the available driver capacity, resulting in a relatively modest impact on workload imbalance. Similarly, the decrease in DCI is not significantly substantial. Ultimately, the decision rests with the decision-maker, who must weigh the trade-off and make a choice based on their specific priorities and considerations.

Instance	△ DCI	△ RBI	Number of drivers
1	- 0,2	+ 9,93	2
3	- 0,1	+ 27,99	6
5	- 0,07	+ 27,48	6
7	- 0,05	+ 7,19	12
8	- 0,05	+ 18,27	10
20	- 0,02	+ 4,8	16
21	- 0,02	+ 5,73	21
24	- 0,02	+ 2,94	65

Table 5.5: Trade-off between the best DCI achieved and the worsening of the RBI.

5.4.3 Running time

There can be variations in running times to obtain the results of Table 5.3, mainly due to the inherent randomness of the neighborhood structures. Figure 5.2 provides a visual representation of the minimum and the maximum running time achieved per instance, put in comparison with the mean running time and the standard deviation across all runs for each instance. Additionally, the figure demonstrates how the running time increases as the dataset size grows, indicating the augmentation of computational requirements with larger datasets.

The data presented in Figure 5.2 indicates that the standard deviation remains consistently low across different instances. This implies that the duration of the 8 runs for each instance was relatively similar. More generally, the variability does not exhibit a specific pattern and is not particularly influenced by the instance size. This observation can likely be attributed to the inherent randomness present within the neighborhood structures.

It is important to note that **the majority of instances can be solved within a relatively short running time on average**. However, it is noteworthy to mention that there is one particular instance, namely instance 22, which stands out with a running time of approximately 1,5 minutes. This instance seems to be an exception, as the rest exhibit much shorter average running times. The longer running times observed for this specific instance can likely be attributed to several factors. One possible element is a higher number of requests compared to the available number of drivers, which can lead to increased complexity in integrating them efficiently. Additionally, there may be specific characteristics or constraints within this instance that contribute to its long running time. To gain a deeper understanding, further examination and analysis of this specific instance should be conducted.

In a broader context, these data demonstrate the efficiency and feasibility of solving the problem at hand. Moreover, the results suggest that the problem could be effectively addressed even for instances with more than 100 patients, utilizing the parameters outlined in Section 4.5. The algorithm exhibits the capability to handle large instances and generate satisfactory solutions within reasonable computation times. Therefore, based on the observed performance, it can be concluded that **the algorithm is scalable and reliable for solving instances of considerable size**.

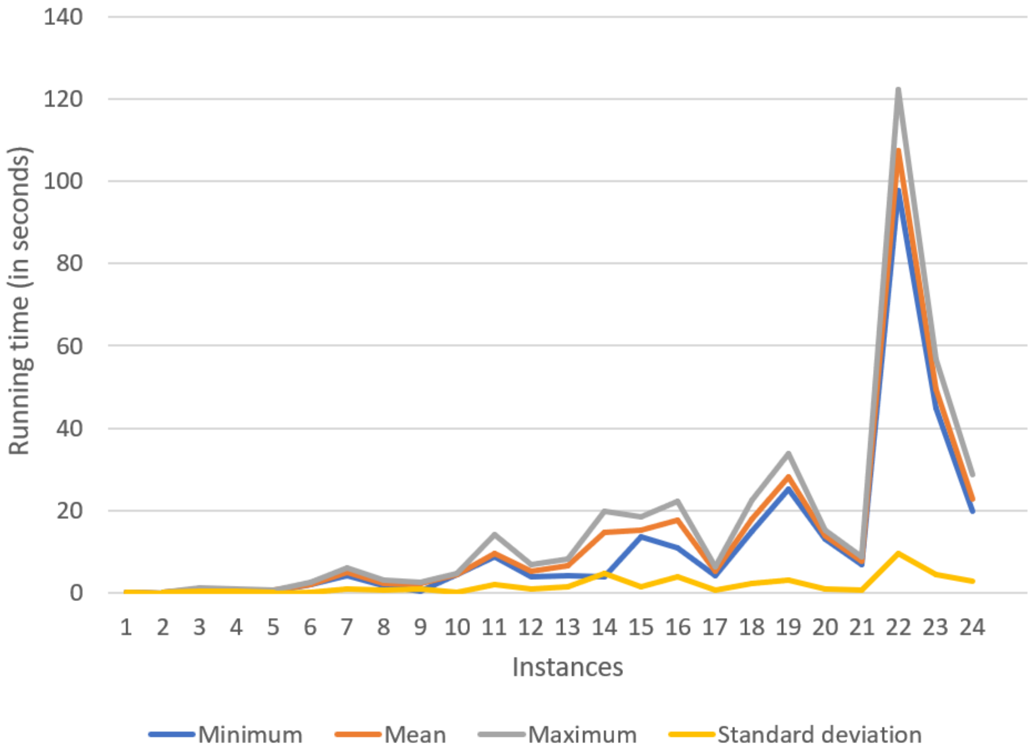

Figure 5.2: Comparison between the minimum, the maximum, and the mean running times for each instance. This figure illustrates how the running time increases as the dataset size grows.

5.4.4 Neighborhood structures

This section will perform a comprehensive analysis of the efficiency of the two neighborhood structures used in the algorithm. Specifically, a comparison will be made between the mean number of iterations needed to optimize the higher-level objective (driver consistency) during the first SA algorithm run and the mean number of iterations needed to optimize the lower-level objective (route balancing) during the second SA algorithm run. This comparison will be conducted for each instance, allowing for an assessment of which neighborhood structure was more effective in the optimization process.

Figure 5.3 shows the mean number of iterations required to optimize driver consistency and the number of iterations required to optimize route balancing among all executed runs for each instance. The abbreviation "DC" corresponds to the Driver Consistency indicator, while "RB" corresponds to the Route Balancing indicator. The specific values of these means for each instance can be found in Appendix D.

A first deduction from the data is that, in 37,5% of the instances, a higher number of iterations is required to optimize driver consistency compared to route balancing. Conversely, in 8,3% of the instances, a higher number of iterations is needed to optimize the route balancing objective. For the remaining 54,2% of the instances, the number of iterations needed to optimize both objectives is equal.

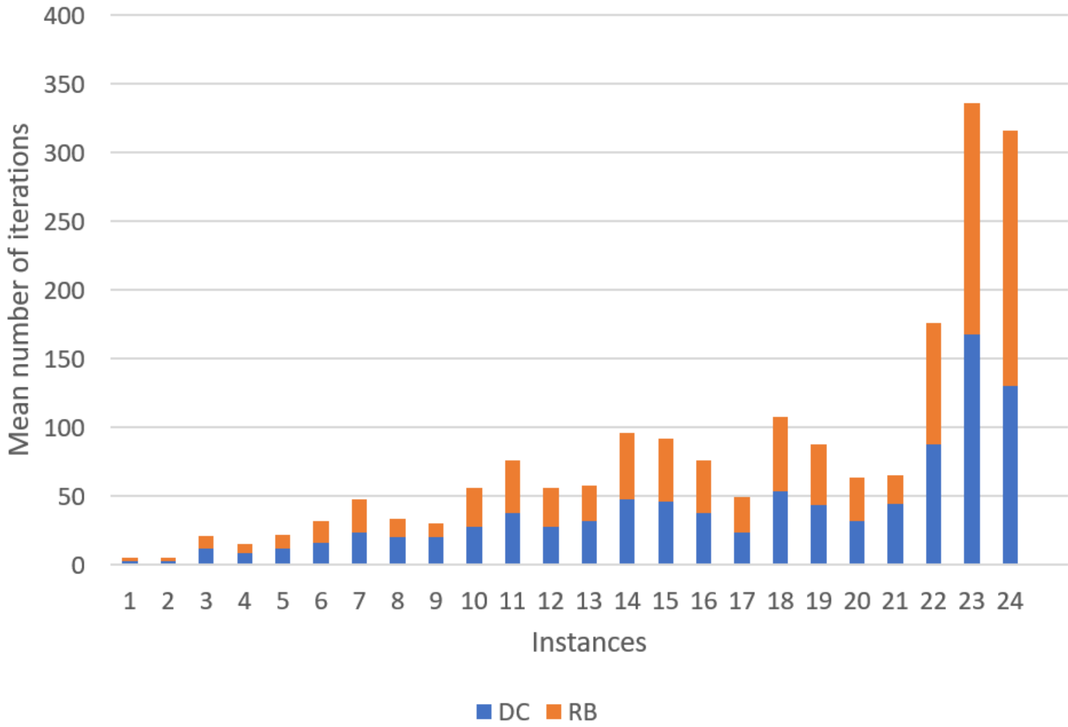

Figure 5.3: Mean number of iterations used to optimize driver consistency in the first SA and route balancing in the second SA for each instance.

Further examination of the obtained values reveals a logical and significant finding. **In 71% of the instances, the best solution is achieved immediately after optimizing the driver consistency objective**. This finding aligns with the *repair* operator of the neighborhood structure described in Section 4.3.1, which takes into account drivers and their total route duration across all days. Consequently, route balancing is inherently optimized to some extent during the optimization of the driver consistency objective. As a result, it can be deduced that **during the optimization of the driver consistency objective, an optimal or nearly optimal level of route balancing is already achieved**. Therefore, in the second phase of the optimization process, there is limited potential for further improvement in the route balancing objective, leading to a smaller number of iterations. Thus, it can be concluded that the neighborhood structure described in Section 4.3.1, which primarily focuses on driver consistency, is highly effective in optimizing both driver consistency and route balancing in the majority of cases.

The neighborhood structure described in Section 4.3.2, which specifically targets route balancing, is only necessary for 29% of the instances to further enhance the optimization of route balancing. Therefore, this additional neighborhood structure plays a valuable role in fine-tuning the optimization process and attaining better results in approximately one-third of the cases.

In Appendix E, the achieved results for the DCI and the RBI are presented for both the first and second runs of the SA algorithm. These results provide an assessment of the effectiveness of both SA algorithms and correspond to the best solutions obtained across all runs conducted for each instance.

5.4.5 Comparison of the results obtained with the characteristics of the instances

This section aims to compare the results obtained based on the characteristics of the instances to determine if some characteristics result in better outcomes compared to others. The goal is to investigate if any predefined conditions lead to improved results in terms of driver consistency and route balancing.

5.4.5.1 Number of request types per patient

As explained in Section 5.1, the number of request types per patient can follow two scenarios. In the first scenario (t(0) assumption), each patient is allowed to have only one request type throughout the week, which can be repeated. In the second scenario (t(1) assumption), patients are allowed to have up to two different request types per week, with a probability of 20%.

Figure 5.4 illustrates the comparison of both DCI and RBI levels between the instances under the t(0) assumption and the instances under the t(1) assumption.

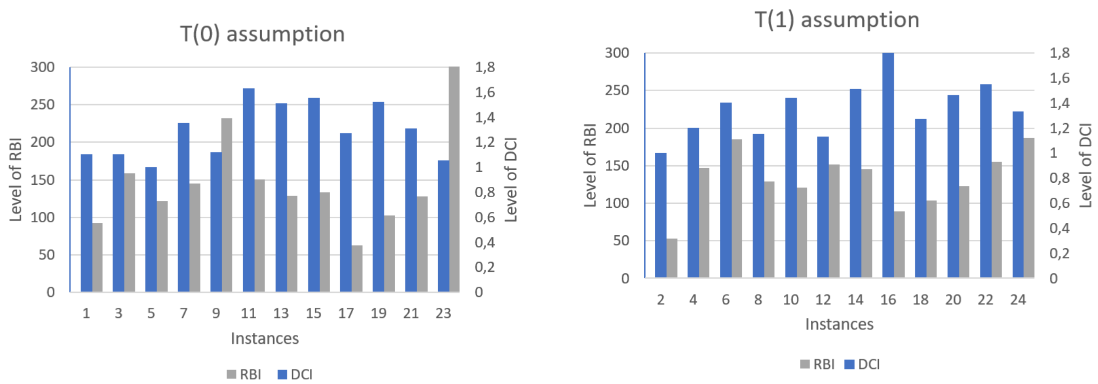

Figure 5.4: Comparison of DCI and RBI for instances under t(0) and t(1) assumptions.

The analysis reveals that **the mean DCI is better under t(0) assumption (1,29)** compared to t(1) assumption (1,36). However, **the average RBI is better under t(1) assumption (132,36)** compared to t(0) assumption (147,74).

These findings align with logical expectations. Under t(0) assumption, where patients are limited to one request type over the week, there are fewer total requests per patient. As a result, fewer drivers are needed to handle the requests of single patients, leading to better driver consistency. However, this can lead to an imbalanced workload among drivers if some drivers have to serve more patients compared to others, also depending on their respective level of request repetition. The allocation of drivers to patients during the driver consistency optimization phase plays a crucial role in determining the workload distribution among drivers in this case. On the other hand, under t(1) assumption, where patients can have up to two different request types per week, the number of total requests per patient increases. This higher demand necessitates potentially more drivers to fulfill the requests of single patients, resulting in worsened driver consistency but improved route balancing since it potentially addresses workload imbalances.

5.4.5.2 Frequency of request types repetition per patient

As a reminder, the frequency of request types per patient can be categorized into three scenarios. In the first scenario (f(0) assumption), each patient has a low frequency of trips of the same type. In the second scenario (f(1) assumption), this frequency is moderate. Finally, in the third scenario (f(2) assumption), this frequency is relatively high.

Figure 5.5 illustrates the comparison of both DCI and RBI levels between the instances under the f(0) assumption, the instances under f(1) assumption and the instances under f(2) assumption.

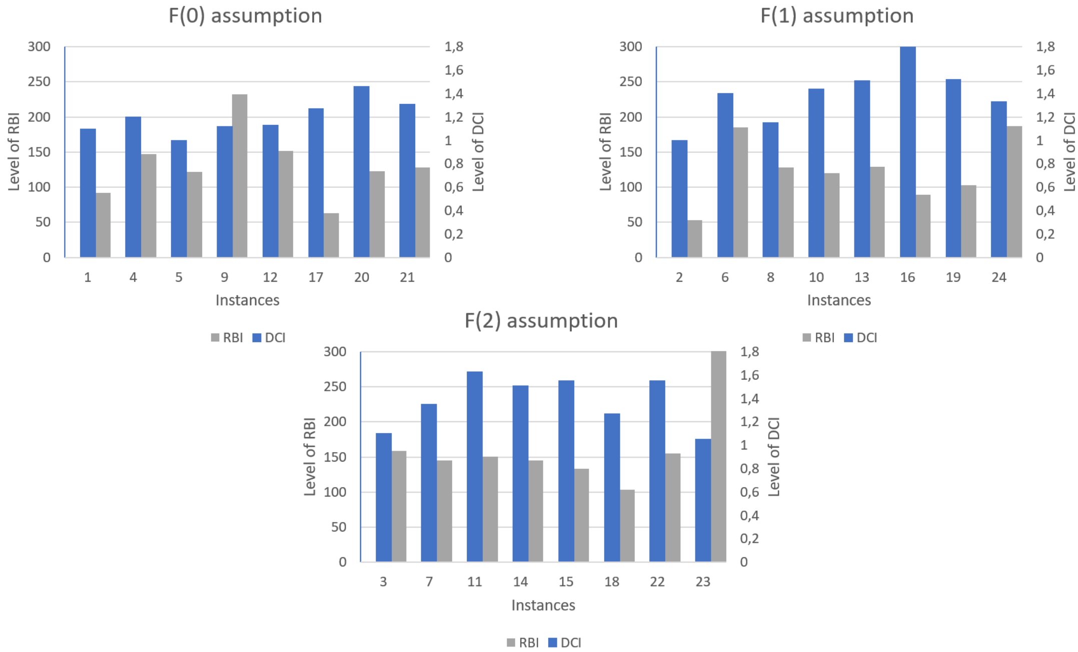

Figure 5.5: Comparison of DCI and RBI for instances under f(0), f(1), and f(2) assumptions.

Based on the data and a closer analysis of the values, it can be observed that **the DCI is, on average, better under f(0) assumption (1,20)** compared to f(1) assumption (1,40) and f(2) assumption (1,38). This suggests that when patients have a lower frequency of trips of the same type, the driver consistency tends to be higher. In contrast, **the RBI is, on average, better under f(1) assumption (124,23)** compared to f(0) assumption (132,31) and f(2) assumption (163,61). This indicates that when patients have a moderate frequency of trips of the same type, the RBI tends to be minimized.

The frequency of each request type directly influences the total number of requests per patient. When the frequency of each request type is lower as under f(0) assumption, there will be fewer total requests per patient. This lower demand provides more opportunities to optimize the DCI since there is a smaller pool of requests per patient to be assigned to single drivers. However, it can result in a more imbalanced workload distribution among available drivers. In such cases, some drivers may encounter a higher number of requests to serve compared to others, due to the specific patient assignments made during the driver consistency optimization and the nature of their requests.

Conversely, if the frequency of all request types is higher as in f(2) assumption, it will result in a higher total number of requests. This increased demand, when combined with driver consistency optimization, can potentially lead to imbalances in workload distribution among drivers. If the demand exceeds the available capacity of drivers, some drivers may be assigned a disproportionately high number of requests compared to others, depending on the patients they are associated with through the driver consistency optimization phase. It can also negatively impact driver consistency as there will be a greater number of requests to assign, potentially causing more variations in driver assignments. To achieve the best RBI, a moderate frequency of request types is necessary. This allows for a more balanced workload distribution among drivers, which optimizes the utilization of resources.

5.4.5.3 Clustering level of home and destination locations

As a reminder, the datasets considered in this study exhibit three different levels of clustering among locations. In the first scenario (c(0) assumption), all locations, including both home and destination locations, are randomly distributed. In the second scenario (c(1) assumption), home locations are still randomly distributed, but destination locations exhibit clustering patterns. Finally, in the third scenario (c(2) assumption), both home and destination locations are clustered. These varying levels of clustering provide insights into the spatial distribution of locations and can impact the optimization of driver consistency and route balancing objectives.

Figure 5.6 illustrates the comparison of both DCI and RBI levels between the instances under c(0) assumption, the instances under c(1) assumption and the instances under c(2) assumption.

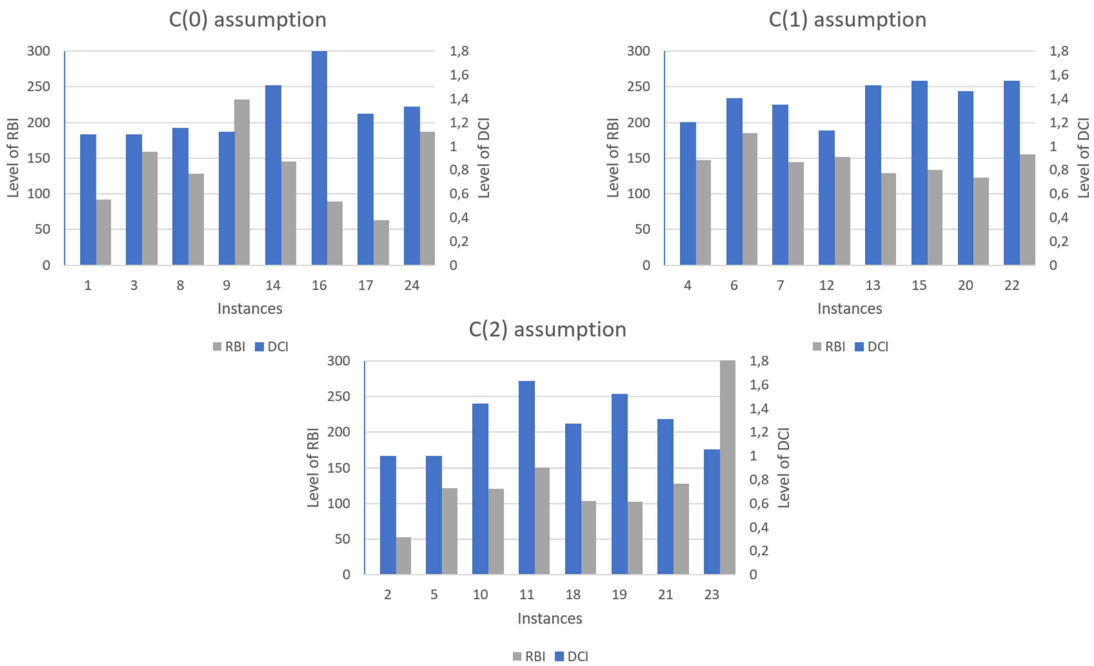

Figure 5.6: Comparison of DCI and RBI for instances under c(0), c(1) and c(2) assumptions.

Upon comparing the data across the three clustering scenarios, it appears that **the average DCI is better under c(2) assumption (1,28)** compared to c(1) assumption (1,39) and c(0) assumption (1,31). This suggests that when both home and destination locations are clustered, there is a higher level of driver consistency achieved. On the other hand, **the average RBI is**

better under c(0) and c(2) assumptions (136,94 for c(0) and 137,08 for c(2)) compared to c(1) assumption (146,12). This indicates that when either all locations are randomly distributed or both home and destination locations are clustered, there is a more balanced distribution of workload among drivers, leading to a better RBI.

When both home and destination locations are clustered (c(2) assumption), it means that there is a concentration of requests in specific areas. This clustering can facilitate the optimization of both driver consistency and route balancing. By assigning drivers to specific geographical areas with concentrated demand, travel times between destinations are reduced, improving both driver consistency and route balancing. On the other hand, when all locations (c(0) assumption) are randomly distributed, the requests are more evenly spread across the geographic area. This can result in some cases in a more balanced workload distribution among drivers, contributing to better route balancing. However, this random distribution of locations also leads to challenges in driver consistency, as it may take longer for drivers to travel between various locations, making it harder to fulfill all the requests of individual patients.

In summary, clustering of both home and destination locations tends to improve both driver consistency and route balancing. Random distribution of locations can contribute to good route balancing in some cases, but it may come at the cost of reduced driver consistency.

5.4.5.4 Summary

After analyzing the findings for each characteristic of the instances, two main conclusions can be drawn.

1. In the analyzed instances, it is observed that **driver consistency tends to be better, on average, when the frequency of requests per patient is the lowest (under f(0) assumption)**. This can be attributed to two main factors. Firstly, when patients have a low number of request repetitions, it becomes relatively easier to assign all of their requests to a single driver, resulting in improved driver consistency. Secondly, as the total number of requests per patient increases, the limited number of available drivers becomes a constraint, leading to a decrease in driver consistency. It is important to note that **the availability of additional drivers could positively impact both driver consistency and route balancing**. The key advantage would be the ability to distribute requests more evenly among the drivers, which in turn would lead to a more balanced workload distribution. In addition, the allocation of patients among a larger pool of drivers would have a positive impact on driver consistency. Indeed, it would enable the assignment of all the requests from patients who have the highest number of requests or are currently assigned to the largest number of drivers to the newly available driver.

2. Furthermore, the findings indicate that **clustering both home and destination locations can lead to improvements in both driver consistency and route balancing**. When locations are clustered, drivers can be assigned to specific geographical areas, allowing them to focus on serving a dedicated set of patients within their assigned cluster in order to improve driver consistency. This clustering also reduces the travel distances between locations within each cluster. However, it is important to ensure a similar number of requests in each geographical area to distribute the workload evenly among drivers to optimize route balancing to the best extent.

5.5 Integration of driver consistency and route balancing

The computational experiments conducted in this study demonstrate **the potential to simultaneously optimize driver consistency and route balancing within the same problem formulation**.

In the ideal scenario, each patient is assigned to a single driver who will fulfill all its requests throughout the entire time horizon, thereby optimizing **driver consistency**. To achieve an optimal level of **route balancing**, it is necessary for drivers to have a similar distance to travel over the entire time horizon. By prioritizing the improvement of driver consistency initially in the employed method, it becomes feasible to assign the requests of each patient to a minimum number of drivers, even if the ideal scenario cannot be reached. This was proved since the majority of patients only require the services of a single driver throughout the entire time horizon in the obtained results. Even more, the obtained DCI is equal to 1 for some instances. In a subsequent phase, the workload among the drivers was balanced to ensure that each driver has a comparable ride time over the time horizon, thus enhancing route balancing to the best possible extent. This integrated approach allows for the attainment of solutions that not only ensure a minimal distribution of drivers among patients but also minimize workload imbalances among the drivers themselves. Although some instances pose challenges, the utilized approach offers substantial opportunities for optimization.

However, it is important to acknowledge **the challenge posed by the limited number of available drivers in the analyzed dataset**. Even in the best solution found, some patients still encountered a relatively high number of different drivers over the time horizon. Having one or more additional drivers (according to the instances) could have been beneficial in reducing the DCI and the RBI even further. To determine the optimal number of drivers needed, a comprehensive analysis considering the costs associated with hiring or laying off drivers, as well as the costs of patients' inconvenience and workload imbalances, could have been conducted. This analysis would have helped in evaluating the trade-offs between the benefits of improved driver consistency, route balancing and the associated costs.

5.6 Managerial implications and limitations

Once the optimal number of drivers is determined to optimize both driver consistency and route balancing in a specific instance, the practical challenge lies in having this exact number of drivers available, since **achieving the optimal number of drivers may not always be feasible in practice**. It highlights the need to consider practical constraints and potential trade-offs between driver consistency, route balancing, and the availability of resources in real-world implementations. Two scenarios can occur:

- In some scenarios, a **shortage of available drivers** can arise, negatively impacting both driver consistency and route balancing. The limited number of drivers creates less flexibility in managing their schedules to maintain driver consistency, and as a result, the workload assigned to each driver increases, potentially leading to an unfair distribution. This shortage can also lead to the violation of constraints related to both patient and driver satisfaction. These circumstances undermine the ability to achieve the desired objectives and can result in a sub-optimal allocation of resources.
- On the other hand, there may be a situation where **more drivers are available than initially planned**, which can result in the need to lay off or dismiss certain drivers. This

can lead to dissatisfaction among these drivers, as they may have anticipated a consistent workload and stable employment. Such workforce reductions can have adverse effects on morale, motivation, and overall driver satisfaction.

Furthermore, **the optimal number of drivers varies inevitably depending on the week considered and the requests that need to be fulfilled**. Given that it is not feasible to engage or lay off drivers based on short-term needs, it becomes essential to **define an optimal number of drivers that minimizes the negative externalities associated with having an excessive or insufficient number of drivers**. This requires implementing proactive management strategies that align the number of drivers with long-term potential demand, taking into account factors such as workload distribution and driver consistency. By effectively managing this balance, it is possible to optimize the allocation of resources, improve overall service quality, and foster a positive working environment for drivers. This approach aims to enhance operational efficiency, patient satisfaction, and the well-being of staff members.

In addition to the limited availability of drivers, **budgetary constraints** pose a crucial limitation when aiming to achieve an optimal number of drivers. These constraints encompass expenses related to driver recruitment, training, vehicle maintenance, fuel consumption, insurance coverage, and unforeseen costs. As such, they can significantly impact the ability to attain the desired number of drivers.

Effective planning and managing staff capacity in the context of patient transportation necessitates having **sufficient advanced knowledge of patients' requests**. However, in practical terms, some patients may not be able to determine the exact day and time of their medical appointments well in advance due to factors beyond their control, such as medical facility schedules and availability. This introduces a challenge in aligning patient requests with driver availability and necessitates flexibility in the planning process to accommodate dynamic and changing circumstances.

In conclusion, **reaching the ideal scenario of establishing one-to-one relationships between drivers and patients while maintaining an optimal level of route balancing is often infeasible in real-life situations**. The dynamic nature of patient requests, unpredictable changes in scheduling, and varying driver availability pose challenges to achieving such an idealized scenario. It is important to acknowledge these practical constraints and **focus on finding a balance that maximizes efficiency, as well as patients' and drivers' satisfaction within the limitations of the system**. This requires ongoing evaluation, adaptation, and optimization to achieve the best possible outcomes in patient transportation.

6 Conclusion and future research

In this chapter, the study findings are summarized, including a discussion of the potential limitations encountered. Additionally, potential directions for future research are highlighted.

6.1 Purpose and findings of the study

The rising concerns about the environment, traffic congestion, fuel costs, and limited mobility for individuals have led to an increased need for shared mobility systems. As the population continues to grow and age, there is a significant growing need for on-demand transportation services, accompanied by higher patient expectations regarding service quality. In addition, in a world with abundant job opportunities but recurring labor shortages, it becomes crucial to prioritize and address the working conditions of employees. To organize it, advancements in computational technologies offer opportunities to enhance digital mobility planning systems.

The DARP focuses on the transportation of individuals, particularly elderly and disabled people, which sets it apart from traditional VRPs. More precisely, the MP-DARP involves generating routes and schedules for a fleet of vehicles to fulfill patients' transportation requests over multiple working days while adhering to various constraints related to route feasibility, schedule feasibility, and the requirements of both patients and drivers. The inherent human aspect of the DARP makes it a distinctive and increasingly relevant problem in today's context.

An increasing number of studies focus on patients' satisfaction in the context of DARP and strive to incorporate it either as a constraint or directly in the objective function. It has been demonstrated that patients attach great importance to having a consistent set of drivers fulfilling their transportation requests over the time horizon. This enables them to establish stronger relationships with the drivers who, in turn, can better understand and cater to their individual needs. This aspect is called **driver consistency**. Some previous studies, such as the recent work by Braekers and Kovacs (2016), have considered this aspect in the MP-DARP.

The well-being and productivity of drivers are influenced by factors such as fair treatment and equal distribution of workload. Achieving a balanced allocation of routes among drivers over the entire time horizon is a critical aspect known as **route balancing**. This ensures equitable working hours and helps promote driver satisfaction and engagement. To date, no study has incorporated route balancing in any formulation of the DARP.

The main objective of this thesis was to investigate the integration of driver consistency and route balancing aspects within a unified problem formulation for the MP-DARP. Proposed solutions were obtained using a SA algorithm.

To achieve this goal, the study utilized the dataset created by Braekers and Kovacs (2016), and the proposed algorithm was implemented using the Julia Programming Language (Version 1.7.2). The initial solution for the problem was generated using an insertion heuristic method, and two distinct neighborhood structures were defined to facilitate the exploration and optimization process.

The study demonstrated that significant improvements can be achieved by optimizing driver consistency and route balancing together. Although reaching the absolute minimum values for both DCI and RBI together may not be achievable, the results showcased the potential to substantially enhance both indicators simultaneously. The findings emphasize the importance of considering these objectives jointly and striving for optimal solutions that strike a balance between them.

6.2 Limitations of the methodology applied

Regarding the methodology employed in the study, several limitations need to be acknowledged and considered.

Regarding the meta-heuristic utilized in the study, it is important to note that **alternative meta-heuristics**, such as Tabu Search (TS) or Large Neighborhood Search (LNS), **could have been explored and analyzed** to determine if they would yield improved results. Additionally, **the study could have explored hybridization approaches** by integrating the SA with LNS or Variable Neighborhood Search (VNS), allowing for the consideration of several move possibilities during the local search process and potentially reaching better solutions. These alternative methodologies and hybridization techniques could have provided valuable insights and potentially improved the overall optimization process.

In addition, while the study considered all constraints as hard constraints to maximize patient and driver satisfaction, it could have been beneficial to **allow for the possibility of violating some constraints during the optimization process**. By incorporating a flexible constraint violation mechanism, the study could have explored a larger set of solutions while ensuring that the final solution obtained complies with all constraints. This approach would have allowed for a more comprehensive search of the solution space, potentially leading to the identification of better and more diverse solutions that still adhere to the necessary constraints.

Testing the impact of increasing the number of available drivers would have been valuable for exploring potential improvements and determining the optimal number of drivers. However, by keeping the number of drivers as a parameter in the tested dataset, the analysis provided a more realistic representation of real-life situations. This approach acknowledges the constraints and challenges faced in managing driver availability, allowing for a more accurate evaluation of the proposed solutions within practical limitations.

Furthermore, it is important to note that **the specific dataset used in this study confined all patients within a square region of 20km^2**. As a result, the total route duration for each driver became heavily dependent on the number of requests they had on their routes within this limited area. To explore the boundaries and challenges of the problem, it would be beneficial for future studies to examine scenarios with dispersed demand across a larger geographical area. This way, researchers could better understand the impact on driver allocation, route planning, and overall system performance. This analysis would provide insights into the scalability and adaptability of the proposed solutions, considering different demand patterns and their effects on driver consistency and route balancing.

Lastly, it is important to note that while the **algorithm code** may have demonstrated its efficiency in providing results, it **was only suited for the specific tested instances**. The scalability and performance of the algorithm on larger, more complex instances with different constraints may differ and could introduce additional challenges. Therefore, the applicability and generalizability of the algorithm to real-world scenarios with larger datasets should be further investigated to ensure its effectiveness in practical implementation.

6.3 Future research

Further research on the topic could be conducted to delve deeper into the possibility of enhancing both patient and staff well-being within a single DARP formulation.

To enhance the problem at hand, an option that could be considered is the inclusion of the **territory consistency** concept, as described by Tellez et al. (2022). Territory consistency involves dividing drivers into specific geographic areas where they exclusively operate, resulting in higher driver consistency. Assigning drivers to patients within their designated areas improves familiarity with the region and leads to more efficient service delivery. When determining the boundaries of these geographic areas, careful thought needs to be given to the locations of medical facilities. The division should aim to create sections with comparable sizes and travel times to the facilities. This ensures that the workload distribution is balanced among drivers, ultimately enhancing operational efficiency and patient service.

With sufficient time, it would have been valuable to **establish contact with individuals working in the field to understand their real-life operational constraints**. This would have helped to determine if considerations such as driver consistency and route balancing are already incorporated into their planning strategies. By engaging with industry professionals and gathering insights from their experiences, it would have been possible to gain a deeper understanding of the practical challenges faced in implementing these strategies and identify potential areas for improvement in the proposed solutions.

To further align shared mobility systems with environmental goals, additional experiments could be conducted by **incorporating ecological measures into the objective functions**. These ecological measures can include minimizing fuel consumption, reducing carbon emissions, or promoting the use of electric or hybrid vehicles. By conducting experiments that incorporate ecological considerations, it becomes possible to identify and promote solutions that strike a balance between operational efficiency, patient satisfaction, and environmental responsibility. This approach supports the broader objective of creating shared mobility systems that are both socially and environmentally sustainable.

To better reflect real-life operational constraints, another valuable approach would be to **incorporate the impact of expected congestion during peak hours on the speed of vehicles**. This variation in vehicle speed can be addressed by exploring time-dependent DARP models, as demonstrated in previous research such as Ichoua et al. (2003). By incorporating time-dependent constraints, the models can better capture the actual travel conditions experienced by drivers on the route. By comparing the results obtained from this approach to those obtained under the assumption of constant vehicle speed, it becomes possible to evaluate the impact of considering congestion on driver consistency and route balancing. This analysis would provide insights into the potential differences and improvements that arise when realistic speed variations are incorporated into the optimization process, leading to more accurate and effective solutions for shared mobility systems.

In line with the previous suggestion, the study primarily focused on static and deterministic DARPs. However, as mentioned in Section 2.2.3, **the dynamic aspect is more representative of real-life scenarios**. Therefore, it would be beneficial to incorporate dynamic considerations into the problem at hand, taking into account various unpredictable events such as patient no-shows, delays in medical appointments, accidents on the route, or deviations. To address these dynamics, a clear framework should be established to outline the necessary adjustments in planning to maintain driver consistency and route balancing. Moreover, in the case in which one or more patients cannot be served due to unforeseen circumstances, it is crucial to incorporate mechanisms for adjusting the planning and adapting between days. This requires integrating feedback and information about missed requests into the system, allowing for informed decisions and efficient resource allocation in subsequent scheduling.

In conclusion, **this thesis represents the pioneering integration of driver consistency and route balancing considerations within a unified MP-DARP formulation**. As the first work in this area, it is essential to acknowledge its potential limitations. Nonetheless, this study serves as a foundation for a broad range of topics that warrant further analysis and exploration. Future research can build upon this initial work to delve deeper into the subject.

Appendices

A. List of the 24 chosen instances from Braekers and Kovacs (2016)

- dcdarp-5-t(0)-f(0)-h(1)-c(0)
- dcdarp-5-t(1)-f(1)-h(1)-c(2)
- dcdarp-10-t(0)-f(2)-h(1)-c(0)
- dcdarp-10-t(1)-f(0)-h(1)-c(1)
- dcdarp-15-t(0)-f(0)-h(1)-c(2)
- dcdarp-15-t(1)-f(1)-h(1)-c(1)
- dcdarp-20-t(0)-f(2)-h(1)-c(1)
- dcdarp-20-t(1)-f(1)-h(1)-c(0)
- dcdarp-25-t(0)-f(0)-h(1)-c(0)
- dcdarp-25-t(1)-f(1)-h(1)-c(2)
- dcdarp-30-t(0)-f(2)-h(1)-c(2)
- dcdarp-30-t(1)-f(0)-h(1)-c(1)
- dcdarp-35-t(0)-f(1)-h(1)-c(1)
- dcdarp-35-t(1)-f(2)-h(1)-c(0)
- dcdarp-40-t(0)-f(2)-h(1)-c(1)
- dcdarp-40-t(1)-f(1)-h(1)-c(0)
- dcdarp-45-t(0)-f(0)-h(1)-c(0)
- dcdarp-45-t(1)-f(2)-h(1)-c(2)
- dcdarp-50-t(0)-f(1)-h(1)-c(2)
- dcdarp-50-t(1)-f(0)-h(1)-c(1)
- dcdarp-75-t(0)-f(0)-h(1)-c(2)
- dcdarp-75-t(1)-f(2)-h(1)-c(1)
- dcdarp-100-t(0)-f(2)-h(1)-c(2)
- dcdarp-100-t(1)-f(1)-h(1)-c(0)

B. Obtained values from all runs for each instance

Instance	Run	Obtained DCI	Obtained RBI	Time elapsed (seconds)
1	1	1	58,91	0,13
	2	1,2	48,98	0,05
	3	1,2	48,98	0,07
	4	1	160,92	0,04
	5	1	160,92	0,05
	6	1,2	48,98	0,04
	7	1,2	48,98	0,04
	8	1	160,92	0,09
2	1	1	53,13	0,02
	2	1	53,13	0,01
	3	1	53,13	0,04
	4	1	53,13	0,05
	5	1	53,13	0,04
	6	1	53,13	0,05
	7	1	53,13	0,05
	8	1	53,13	0,05
3	1	1,1	159,02	0,59
	2	1,1	159,02	0,88
	3	1,1	159,02	0,74
	4	1,1	159,02	0,81
	5	1,1	159,02	0,56
	6	1,1	159,02	0,83
	7	1,1	159,02	0,88
	8	1,1	159,02	1,23
4	1	1,2	134,70	0,45
	2	1,2	231,83	0,45
	3	1,2	134,70	0,25
	4	1,2	40,83	0,45
	5	1,2	134,70	0,78
	6	1,2	134,70	0,15
	7	1,2	231,83	0,34
	8	1,2	134,70	0,36
5	1	1	121,62	0,50
	2	1	121,62	0,55
	3	1	121,62	0,59
	4	1	121,62	0,73
	5	1	121,62	0,48
	6	1	121,62	0,41
	7	1	121,62	0,42
	8	1	121,62	0,70
6	1	1,4	184,30	2,36
	2	1,4	215,36	1,88
	3	1,4	184,30	1,88

	4	1,4	157,72	2,00
	5	1,4	157,72	1,82
	6	1,4	215,36	2,03
	7	1,4	184,30	2,02
	8	1,4	184,30	2,03
7	1	1,35	148,21	4,42
	2	1,35	139,53	4,09
	3	1,35	148,21	4,18
	4	1,35	137,35	4,54
	5	1,35	148,21	4,71
	6	1,35	151,14	5,52
	7	1,35	148,21	5,71
	8	1,35	137,35	6,00
8	1	1,15	127,65	1,72
	2	1,15	127,65	1,59
	3	1,15	127,65	3,13
	4	1,15	130,05	1,77
	5	1,15	130,05	2,44
	6	1,15	130,05	2,51
	7	1,15	127,65	2,17
	8	1,15	127,65	1,85
9	1	1,12	232,17	2,23
	2	1,12	232,17	2,11
	3	1,12	232,17	2,54
	4	1,12	232,17	2,39
	5	1,12	232,17	0,35
	6	1,12	232,17	2,62
	7	1,12	232,17	2,47
	8	1,12	232,17	1,99
10	1	1,44	129,16	4,42
	2	1,44	119,05	4,28
	3	1,44	103,79	4,52
	4	1,44	129,16	4,42
	5	1,44	124,55	4,45
	6	1,44	129,16	4,35
	7	1,44	98,92	4,47
	8	1,44	129,16	4,42
11	1	1,63	147,06	8,91
	2	1,63	182,11	8,86
	3	1,63	152,68	8,84
	4	1,63	152,68	9,23
	5	1,63	122,12	9,03
	6	1,63	152,68	9,05
	7	1,63	152,68	9,05
	8	1,63	182,11	14,20
	1	1,13	151,62	4,39
	2	1,13	151,62	5,24

12

	3	1,13	151,62	4,57
	4	1,13	151,62	3,91
	5	1,13	151,62	4,59
	6	1,13	151,62	5,94
	7	1,13	151,62	6,82
	8	1,13	151,62	5,61
13	1	1,51	132,85	6,45
	2	1,51	100,67	4,02
	3	1,51	130,12	4,93
	4	1,51	132,85	6,83
	5	1,51	132,85	7,90
	6	1,51	165,91	8,22
	7	1,51	132,85	6,26
	8	1,51	100,69	7,42
14	1	1,51	145,50	13,84
	2	1,51	145,50	15,86
	3	1,51	145,50	16,81
	4	1,51	145,50	14,69
	5	1,51	145,50	17,00
	6	1,51	145,50	14,59
	7	1,51	145,50	3,95
	8	1,51	145,50	19,67
15	1	1,55	133,40	18,37
	2	1,55	133,40	13,59
	3	1,55	133,40	15,38
	4	1,55	133,40	14,89
	5	1,55	133,40	15,74
	6	1,55	133,40	14,97
	7	1,55	133,40	14,06
	8	1,55	133,40	14,52
16	1	1,88	89,32	10,90
	2	1,88	85,02	13,76
	3	1,88	89,32	22,07
	4	1,88	89,32	18,84
	5	1,88	89,32	20,03
	6	1,88	89,32	20,33
	7	1,88	89,32	18,21
	8	1,88	89,32	16,09
17	1	1,27	62,76	4,63
	2	1,27	62,76	5,06
	3	1,27	63,52	5,23
	4	1,27	62,76	4,06
	5	1,27	62,76	5,10
	6	1,27	62,76	6,28
	7	1,27	62,76	4,55
	8	1,27	62,76	5,79
18	1	1,27	99,37	14,87

	2	1,27	106,25	17,44
	3	1,27	101,99	22,48
	4	1,27	106,25	19,34
	5	1,27	101,99	16,90
	6	1,27	106,25	18,39
	7	1,27	101,99	17,22
	8	1,27	101,99	17,43
19	1	1,52	101,45	26,98
	2	1,52	101,45	30,53
	3	1,52	105,42	33,80
	4	1,52	102,21	26,63
	5	1,52	102,21	26,26
	6	1,52	101,95	29,99
	7	1,52	99,22	26,61
	8	1,52	105,42	25,14
20	1	1,46	110,26	15,29
	2	1,46	110,26	14,21
	3	1,46	131,89	13,33
	4	1,46	131,89	13,18
	5	1,46	125,44	13,60
	6	1,46	107,63	14,12
	7	1,46	131,89	14,64
	8	1,46	131,89	13,02
21	1	1,31	128,09	7,00
	2	1,31	128,09	8,09
	3	1,31	128,09	8,66
	4	1,31	128,09	7,58
	5	1,31	128,09	8,25
	6	1,31	128,09	6,78
	7	1,31	128,09	6,87
	8	1,31	128,09	7,01
22	1	1,55	155,28	97,90
	2	1,55	155,28	122,36
	3	1,55	155,28	102,46
	4	1,55	155,28	106,08
	5	1,55	155,28	122,05
	6	1,55	155,28	100,02
	7	1,55	155,28	105,80
	8	1,55	155,28	102,52
23	1	1,05	316,58	44,99
	2	1,05	317,45	48,02
	3	1,05	316,58	47,58
	4	1,05	317,45	47,32
	5	1,05	316,58	56,78
	6	1,05	317,45	47,35
	7	1,05	317,45	55,68
	8	1,05	317,45	47,08

24	1	1,33	186,55	20,70
	2	1,33	186,55	21,95
	3	1,33	186,55	19,90
	4	1,33	186,55	23,18
	5	1,33	186,55	21,05
	6	1,33	186,55	28,76
	7	1,33	186,55	22,69
	8	1,33	186,55	24,22

C. Comparison of the number of drivers assigned to each patient between the initial solutions and the best solutions found

Instance	Initial solution	Best solution
1	1 (80%) - 2 (20%)	1 (100%)
2	1 (100%)	1 (100%)
3	1 (90%) - 3 (10%)	1 (90%) - 2 (10%)
4	1 (80%) - 2 (10%) - 3 (10%)	1 (80%) - 2 (20%)
5	1 (93,3%) - 2 (6,7%)	1 (100%)
6	1 (86,7%) - 4 (6,6%) - 5 (6,7%)	1 (86,7%) - 4 (13,3%)
7	1 (85%) - 3 (5%) - 4 (10%)	1 (85%) - 2 (10%) - 4 (5%)
8	1 (85%) - 2 (5%) - 3 (10%)	1 (85%) - 2 (15%)
9	1 (88%) - 2 (12%)	1 (88%) - 2 (12%)
10	1 (80%) - 3 (16%) - 5 (4%)	1 (80%) - 3 (16%) - 4 (4%)
11	1 (83,3%) - 3 (13,4%) - 7 (3,3%)	1 (83,3%) - 3 (6,7%) - 6 (10%)
12	1 (86,7%) - 2 (13,3%)	1 (86,7%) - 2 (13,3%)
13	1 (77,1%) - 2 (8,6%) - 3 (2,9%) - 4 (8,6%) - 6 (2,8%)	1 (77,1%) - 2 (8,6%) - 3 (2,9%) - 4 (8,6%) - 5 (2,8%)
14	1 (74,3%) - 2 (8,6%) - 3 (5,7%) - 4 (8,6%) - 6 (2,8%)	1 (77,1%) - 2 (8,6%) - 3 (2,9%) - 4 (8,6%) - 6 (2,8%)
15	1 (80%) - 2 (5%) - 3 (5%) - 4 (2,5%) - 5 (2,5%) - 6 (5%)	1 (85%) - 2 (5%) - 3 (2,5%) - 5 (2,5%) - 6 (5%)
16	1 (62,5%) - 2 (10%) - 3 (10%) - 4 (12,5%) - 5 (2,5%) - 6 (2,5%)	1 (62,5%) - 2 (10%) - 3 (10%) - 4 (12,5%) - 5 (5%)
17	1 (73,3%) - 2 (24,5%) - 3 (2,2%)	1 (73,3%) - 2 (24,5%) - 3 (2,2%)
18	1 (91,1%) 2 (2,2%) - 5 (6,7%)	1 (91,1%) - 2 (2,2%) - 4 (2,2%) - 5 (4,4%)
19	1 (78%) - 2 (6%) - 3 (4%) - 4 (10%) - 6 (2%)	1 (78%) - 2 (6%) - 3 (4%) - 4 (10%) - 5 (2%)
20	1 (72%) - 2 (16%) - 3 (4%) - 4 (8%)	1 (72%) - 2 (16%) - 3 (6%) - 4 (6%)
21	1 (80%) - 2 (17,3%) - 6 (1,4%) - 8 (1,3%)	1 (81,3%) - 2 (16%) - 6 (1,3%) - 7 (1,3%)
22	1 (81,3%) - 2 (2,7%) - 3 (5,3%) - 4 (4%) - 5 (4%) - 6 (1,4%) - 7 (1,3%)	1 (81,3%) - 2 (2,7%) - 3 (5,3%) - 4 (4%) - 5 (4%) - 6 (1,3%) - 7 (1,3%)
23	1 (98%) - 4 (2%)	1 (98%) - 3 (1%) - 4 (1%)
24	1 (84%) - 2 (4%) - 3 (5%) - 4 (7%)	1 (86%) - 2 (5%) - 3 (3%) - 4 (6%)

Table 6.2: Column 2 (resp. Column 3) shows the number of drivers assigned to each patient in the initial solution (resp. best solution found), with the frequency of occurrence for each of these values among all patients.

D. Average iterations count among all runs in both simulated annealing algorithms for each instance

Instance	Mean number of iter. first SA	Mean number of iter. second SA
1	3	2,75
2	3,25	2
3	12	9
4	9	6
5	12	9,75
6	16	16
7	24	24
8	20	13,75
9	20	10
10	28	28
11	38	38
12	28	28
13	32	26
14	48	48
15	46	46
16	38	38
17	24	25,5
18	54	54
19	44	44
20	32	32
21	44,63	21
22	88	88
23	168	168
24	130	186,55

Table 6.3: Column 2 (resp. Column 3) shows the average number of iterations needed by the first SA algorithm (resp. second SA algorithm) to optimize the driver consistency objective (resp. route balancing objective).

E. Comparison of the DCI and RBI values obtained after each simulated annealing algorithm in the best solutions found

Instance	DCI first SA	RBI first SA	DCI second SA	RBI second SA
1	1	160,92	1	58,91
2	1	53,13	1	53,13
3	1,1	159,02	1,1	159,02
4	1,2	40,83	1,2	40,83
5	1	121,62	1	121,62
6	1,4	157,77	1,4	157,77
7	1,35	137,35	1,35	137,35
8	1,15	127,65	1,15	127,65
9	1,12	232,17	1,12	232,17
10	1,44	98,92	1,44	98,92
11	1,63	112,84	1,63	112,84
12	1,13	229,26	1,13	151,62
13	1,51	100,69	1,51	100,69
14	1,51	240,65	1,51	145,40
15	1,55	169,62	1,55	133,40
16	1,88	85,02	1,88	85,02
17	1,27	79,81	1,27	62,76
18	1,27	99,37	1,27	99,37
19	1,52	99,22	1,52	99,22
20	1,46	107,63	1,46	107,63
21	1,31	128,09	1,31	128,09
22	1,55	192,10	1,55	155,28
23	1,05	316,58	1,05	316,58
24	1,33	186,55	1,33	186,55

Table 6.4: Column 2 (respectively, Column 3) represents the achieved level of DCI (respectively, RBI) for the best solutions found during the first SA algorithm, which prioritizes optimizing driver consistency. Column 4 (respectively, Column 5) displays the obtained level of DCI (respectively, RBI) also for the best solutions found during the second SA algorithm, which focuses on enhancing route balancing.

References

Alisoltani, N., M. Zargayouna, and L. Leclercq (2019). Data-oriented approach for the dial-a-ride problem. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–6. IEEE.

Atahran, A., C. Lenté, and V. T'kindt (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.

Azad, S., L. Badia, A. Rahman, and J. M. Zain (2016). Raising fairness issue of vehicle routing problem. In *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pp. 13–17. IEEE.

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

Belhaiza, S. (2019). A hybrid adaptive large neighborhood heuristic for a real-life dial-a-ride problem. *Algorithms 12*(2), 39.

Braekers, K., A. Caris, and G. K. Janssens (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. and A. A. Kovacs (2016). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological 94*, 355–377.

Chassaing, M., C. Duhamel, and P. Lacomme (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.

Chen, L.-W., T.-Y. Hu, and Y.-W. Wu (2022). A bi-objective model for eco-efficient dial-a-ride problems. *Asia Pacific management review 27*(3), 163–172.

Chevrier, R., A. Liefooghe, L. Jourdan, and C. Dhaenens (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.

Coelho, L. C., J.-F. Cordeau, and G. Laporte (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies 24*, 270–287.

Coelho, L. C. and G. Laporte (2013). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research 40*(2), 558–565.

Cordeau, J.-F. and G. Laporte (2003). The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1*, 89–101.

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

Cordeau, J.-F., G. Laporte, J.-Y. Potvin, and M. W. Savelsbergh (2007). Transportation on demand. *Handbooks in operations research and management science 14*, 429–466.

Cortés, C. E., M. Matamala, and C. Contardo (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.

Dantzig, G. B. and J. H. Ramser (1959). The truck dispatching problem. *Management science 6*(1), 80–91.

Detti, P., F. Papalini, and G. Z. M. de Lara (2017). A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega 70*, 1–14.

Dong, S. (2022). *New Formulations and Solution Methods for the Dial-a-ride Problem.* Ph. D. thesis, UNSW Sydney.

Dong, X., D. Rey, and S. Travis Waller (2020). Dial-a-ride problem with users' accept/reject decisions based on service utilities. *Transportation research record 2674*(10), 55–67.

D'Souza, C., S. Omkar, and J. Senthilnath (2012). Pickup and delivery problem using metaheuristics techniques. *Expert Systems with Applications 39*(1), 328–334.

Eveborn, P., P. Flisberg, and M. Rönnqvist (2006). Laps care—an operational system for staff planning of home care. *European journal of operational research 171*(3), 962–976.

Fahmy, S. A. (2022). Hybrid and approximate algorithms for the dial-a-ride problem with adaptive ride times considering different service strategies. *Engineering Optimization*, 1–15.

Feillet, D., T. Garaix, F. Lehuédé, O. Péton, and D. Quadri (2014). A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks 63*(3), 211–224.

Feng, B. and L. Wei (2022). An improved multi-directional local search algorithm for vehicle routing problem with time windows and route balance. *Applied Intelligence*, 1–13.

Francis, P., K. Smilowitz, and M. Tzur (2006). The period vehicle routing problem with service choice. *Transportation science 40*(4), 439–454.

Francis, P., K. Smilowitz, and M. Tzur (2007). Flexibility and complexity in periodic distribution problems. *Naval Research Logistics (NRL) 54*(2), 136–150.

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

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

Goeke, D., R. Roberti, and M. Schneider (2019). Exact and heuristic solution of the consistent vehicle-routing problem. *Transportation Science 53*(4), 1023–1042.

Grinshpoun, T., E. Shufan, H. Ilani, V. Levit, and H. Brama (2022). Graph representation of the fixed route dial-a-ride problem. *Journal of scheduling*.

Groër, C., B. Golden, and E. Wasil (2009). The consistent vehicle routing problem. *Manufacturing & service operations management 11*(4), 630–643.

Guerreiro, P. M., P. J. Cardoso, and H. C. Fernandes (2020). A comparison of multiple objective algorithms in the context of a dial a ride problem. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part VII*, pp. 382–396. Springer.

Guerriero, F., M. E. Bruni, and F. Greco (2013). A hybrid greedy randomized adaptive search heuristic to solve the dial-a-ride problem. *Asia-Pacific Journal of Operational Research 30*(01), 1250046.

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

Ham, A. (2023). Dial-a-ride problem: mixed integer programming revisited and constraint programming proposed. *Engineering Optimization 55*(2), 257–270.

Ho, S. C., W. Y. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering, and T. W. Tou (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological 111*, 395–421.

Ho, S. G., R. R. Pandi, S. C. Nagavarapu, and J. Dauwels (2018). Multi-atomic annealing heuristic for the dial-a-ride problem. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 268–273. IEEE.

Hu, T.-Y., G.-C. Zheng, and T.-Y. Liao (2019). Multi-objective model for dial-a-ride problems with vehicle speed considerations. *Transportation research record 2673*(11), 161–171.

Ichoua, S., M. Gendreau, and J.-Y. Potvin (2003). Vehicle dispatching with time-dependent travel times. *European journal of operational research 144*(2), 379–396.

Ilani, H., E. Shufan, T. Grinshpoun, A. Belulu, and A. Fainberg (2014). A reduction approach to the two-campus transport problem. *Journal of Scheduling 17*, 587–599.

Jingjing, L., Y. Fang, and N. Tang (2022). A cluster-based optimization framework for vehicle routing problem with workload balance. *Computers & Industrial Engineering 169*, 108221.

Jorgensen, R. M., J. Larsen, and K. B. Bergvinsdottir (2007). Solving the dial-a-ride problem using genetic algorithms. *Journal of the operational research society 58*(10), 1321–1331.

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. and R. W. Calvo (2013). A granular tabu search algorithm for the dial-a-ride problem. *Transportation Research Part B: Methodological 56*, 120–135.

Kovacs, A. A., B. L. Golden, R. F. Hartl, and S. N. Parragh (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks 64*(3), 192–213.

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

Kovacs, A. A., S. N. Parragh, and R. F. Hartl (2014). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks 63*(1), 60–81.

Lehuédé, F., R. Masson, S. N. Parragh, O. Péton, and F. Tricoire (2014). A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society 65*, 983–1000.

Lespay, H. and K. Suchan (2021). A case study of consistent vehicle routing problem with time windows. *International Transactions in Operational Research 28*(3), 1135–1163.

Lim, A., Z. Zhang, and H. Qin (2017). Pickup and delivery service with manpower planning in hong kong public hospitals. *Transportation Science 51*(2), 688–705.

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

Lozano, J., L.-C. González-Gurrola, E. Rodriguez-Tello, and P. Lacomme (2016). A statistical comparison of objective functions for the vehicle routing problem with route balancing. In *2016 Fifteenth Mexican international conference on artificial intelligence (MICAI)*, pp. 130–135. IEEE.

Luo, Z., H. Qin, C. Che, and A. Lim (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research 243*(3), 731–744.

Malheiros, I., R. Ramalho, B. Passeti, T. Bulhões, and A. Subramanian (2021). A hybrid algorithm for the multi-depot heterogeneous dial-a-ride problem. *Computers & Operations Research 129*, 105196.

Mancini, S., M. Gansterer, and R. F. Hartl (2021). The collaborative consistent vehicle routing problem with workload balance. *European Journal of Operational Research 293*(3), 955–965.

Masmoudi, M. A., K. Braekers, M. Masmoudi, and A. Dammak (2017). A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Computers operations research 81*, 1–13.

Masmoudi, M. A., M. Hosny, E. Demir, and E. Pesch (2020). Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem. *Journal of heuristics 26*(1), 83–118.

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

Matl, P., R. F. Hartl, and T. Vidal (2018). Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science 52*(2), 239–260.

Mauri, G. and L. N. Lorena (2006). A multiobjective model and simulated annealing approach for a dial-a-ride problem. In *Workshop dos cursos de computaçao*, pp. 1–12.

Mauri, G. R., L. Antonio, and N. Lorena (2009). Customers' satisfaction in a dial-a-ride problem. *IEEE Intelligent Transportation Systems Magazine 1*(3), 6–14.

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

Melis, L. and K. Sörensen (2022). The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment. *International Transactions in Operational Research 29*(3), 1417–1453.

Mitrović-Minić, S. and G. Laporte (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological 38*(7), 635–655.

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

Molenbruch, Y., K. Braekers, and A. Caris (2017b). Typology and literature review for dial-a-ride problems. *Annals of Operations Research 259*, 295–325.

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

Molenbruch, Y., K. Braekers, P. Hirsch, and M. Oberscheider (2021). Analyzing the benefits of an integrated mobility system using a matheuristic routing algorithm. *European Journal of Operational Research 290*(1), 81–98.

Mourad, A., J. Puchinger, and C. Chu (2019). A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological 123*, 323–346.

Nasri, S. and H. Bouziri (2017). Improving total transit time in dial-a-ride problem with customers-dependent criteria. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1141–1148. IEEE.

Nasri, S., H. Bouziri, and W. Aggoune-Mtalaa (2021). Customer-oriented dial-a-ride problems: A survey on relevant variants, solution approaches and applications. In *Emerging Trends in ICT for Sustainable Development: The Proceedings of NICE2020 International Conference*, pp. 111–119. Springer.

Nasri, S., H. Bouziri, and W. Aggoune-Mtalaa (2022). An evolutionary descent algorithm for customer-oriented mobility-on-demand problems. *Sustainability 14*(5), 3020.

Paquay, C., Y. Crama, and T. Pironet (2020). Recovery management for a dial-a-ride system with real-time disruptions. *European Journal of Operational Research 280*(3), 953–969.

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

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

Paquette, J., J.-F. Cordeau, G. Laporte, and M. M. Pascoal (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., J.-F. Cordeau, K. F. Doerner, and R. F. Hartl (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR spectrum 34*, 593–633.

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

Parragh, S. N., J. Pinho de Sousa, and B. Almada-Lobo (2015). The dial-a-ride problem with split requests and profits. *Transportation Science 49*(2), 311–334.

Pfeiffer, C. and A. Schulz (2022a). An alns algorithm for the static dial-a-ride problem with ride and waiting time minimization. *Or Spectrum 44*(1), 87–119.

Pfeiffer, C. and A. Schulz (2022b). A new lower bound for the static dial-a-ride problem with ride and waiting time minimization. In *Dynamics in Logistics: Proceedings of the 8th International Conference LDIC 2022, Bremen, Germany*, pp. 231–243. Springer.

Pimenta, V., A. Quilliot, H. Toussaint, and D. Vigo (2017). Models and algorithms for reliability-oriented dial-a-ride with autonomous electric vehicles. *European Journal of Operational Research 257*(2), 601–613.

Pirlot, M. (1996). General local search methods. *European journal of operational research 92*(3), 493–511.

Posada, M., H. Andersson, and C. H. Häll (2017). The integrated dial-a-ride problem with timetabled fixed route service. *Public Transport 9*, 217–241.

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

Qu, Y. and J. F. Bard (2015). A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science 49*(2), 254–270.

Reinhardt, L. B., T. Clausen, and D. Pisinger (2013). Synchronized dial-a-ride transportation of disabled passengers at airports. *European Journal of Operational Research 225*(1), 106–117.

Rist, Y. and M. A. Forbes (2021). A new formulation for the dial-a-ride problem. *Transportation Science 55*(5), 1113–1135.

Ritzinger, U., J. Puchinger, and R. F. Hartl (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research 54*(1), 215–231.

Rodríguez-Martín, I. and H. Yaman (2022). Periodic vehicle routing problem with driver consistency and service time optimization. *Transportation Research Part B: Methodological 166*, 468–484.

Ronald, N., R. Thompson, and S. Winter (2015). Simulating demand-responsive transportation: a review of agent-based approaches. *Transport Reviews 35*(4), 404–421.

Sánchez, A. L., M. Lujak, F. Semet, and H. Billhardt (2022). On balancing fairness and efficiency in routing of cooperative vehicle fleets. In *Twelfth International Workshop on Agents in Traffic and Transportation co-located with the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022)*, Volume 3173.

Schönberger, J. (2017). Scheduling constraints in dial-a-ride problems with transfers: a metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. *Public Transport 9*, 243–272.

Smilowitz, K., M. Nowak, and T. Jiang (2013). Workforce management in periodic delivery operations. *Transportation Science 47*(2), 214–230.

Sörensen, K. and F. Glover (2013). Metaheuristics. *Encyclopedia of operations research and management science 62*, 960–970.

Souza, A. L. S., J. B. Chagas, P. H. Penna, and M. J. Souza (2020). A hybrid heuristic algorithm for the dial-a-ride problem. In *Variable Neighborhood Search: 7th International Conference, ICVNS 2019, Rabat, Morocco, October 3–5, 2019, Revised Selected Papers 7*, pp. 53–66. Springer.

Su, Y., N. Dupin, and J. Puchinger (2023). A deterministic annealing local search for the electric autonomous dial-a-ride problem. *European Journal of Operational Research*.

Su, Y., J. Puchinger, and N. Dupin (2022). A deterministic annealing local search for the electric autonomous dial-a-ride problem.

Tarantilis, C. D., F. Stavropoulou, and P. P. Repoussis (2012). A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications 39*(4), 4233–4239.

Tellez, O., S. Vercraene, F. Lehuédé, O. Péton, and T. Monteiro (2018). The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C: Emerging Technologies 91*, 99–123.

Tellez, O., S. Vercraene, F. Lehuédé, O. Péton, and T. Monteiro (2022). The time-consistent dial-a-ride problem. *Networks 79*(4), 452–478.

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

Wolfler Calvo, R. and N. Touati-Moungla (2011). A matheuristic for the dial-a-ride problem. In *Network Optimization: 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*, pp. 450–463. Springer.

Woodward, C. A., J. Abelson, S. Tedford, and B. Hutchison (2004). What is important to continuity in home care?: Perspectives of key stakeholders. *Social science & medicine 58*(1), 177–192.

Yang, M., Y. Ni, and Q. Song (2022). Optimizing driver consistency in the vehicle routing problem under uncertain environment. *Transportation research part E: logistics and transportation review 164*, 102785.

Yazawa, M., K. Omae, Y. Shibagaki, M. Inaba, K. Tsuruya, and N. Kurita (2020). The effect of transportation modality to dialysis facilities on health-related quality of life among hemodialysis patients: results from the japanese dialysis outcomes and practice pattern study. *Clinical Kidney Journal 13*(4), 640–646.

Zhang, H., H. Ge, J. Yang, and Y. Tong (2021). Review of vehicle routing problems: Models, classification and solving algorithms. *Archives of Computational Methods in Engineering*, 1–27.

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

Zhong, H., R. W. Hall, and M. Dessouky (2007). Territory planning and vehicle dispatching with driver learning. *Transportation Science 41*(1), 74–89.

Zidi, I., K. Mesghouni, K. Zidi, and K. Ghedira (2012). A multi-objective simulated annealing for the multi-criteria dial a ride problem. *Engineering Applications of Artificial Intelligence 25*(6), 1121–1131.

Executive summary

The classic Dial-A-Ride Problem (DARP) is commonly encountered in door-to-door transportation services catering to elderly or disabled individuals (Cordeau and Laporte (2003)). In the DARP, the goal is to plan a set of routes and associated schedules for a fleet of vehicles in order to fulfill outbound and inbound requests from patients while adhering to various constraints. The objective function of the DARP can vary depending on the specific application, encompassing economic and service-level considerations.

The DARP has been extensively studied in the research community for several decades; some notable recent comprehensive surveys on the topic can be found in Ho et al. (2018) and Molenbruch et al. (2017b).

The primary objective of the Dial-A-Ride Problem (DARP) is to fulfill the transportation requirements of patients while ensuring their comfort and satisfaction. However, it is equally important to consider the well-being of the drivers who provide these services. To date, no research has focused on investigating the satisfaction of both patients and drivers within the context of the DARP. Specifically, only a few studies have explored the preference of patients to be served by a consistent set of drivers over multiple time periods, known as driver consistency (Braekers and Kovacs (2016)). Additionally, the concept of route balancing, which ensures an equitable distribution of workload among drivers, has not been extensively studied in the existing literature, except in the context of Vehicle Routing Problems (VRPs) (Matl et al. (2018)).

This thesis is dedicated to the development and implementation of a Simulated Annealing (SA) algorithm specifically tailored for the Multi-Period Dial-A-Ride Problem (MP-DARP). The primary focus is on minimizing the number of distinct drivers encountered by each patient. As a secondary objective, the algorithm aims to achieve a balanced distribution of workload among drivers. The main goal of this research is to demonstrate the feasibility of simultaneously optimizing patient satisfaction and ensuring fairness among drivers within a unified problem formulation for the MP-DARP. The dataset used in this study was initially created by Braekers and Kovacs (2016) and serves as the foundation for the testing and evaluation of the algorithm.

The results obtained from this study shed light on the potential advantages of integrating both patient satisfaction and driver fairness in transportation systems. These findings contribute to improving the overall quality of service in the context of DARPs. While further research is needed to explore the potential limitations of the proposed system, this pioneering work reveals new avenues for investigation in the field of DARPs.

Word count: 22 917