NTNU
Norwegian University of
Science and Technology

Optimal Handling and Repositioning of Modern Carsharing Systems

A Hybrid Genetic Search with Adaptive Diversity Control Approach

Carl Axel Aadne Folkestad
Nora Åsheim Hansen

Industrial Economics and Technology Management
Submission date: June 2017
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Henrik Andersson, IØT
Giovanni Pantuso, Polytechnic of Milan

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Problem Description

The purpose of this thesis is to model and develop solution methods capable of solving realistic problem sizes of the Static Free-Floating Electric Vehicle Carsharing Handling Problem (SFFEVCHP). The SFFEVCHP is concerned with determining the optimal charging stations to charge rental cars and the optimal routes of service vehicles transporting operators moving the cars in a free-floating electric carsharing system. Central to the problem is the trade off between repositioning cost and the cost of not meeting demand due to an unfavorable distribution of rental cars in the system.

Preface

This master's thesis concludes our Master of Science in Industrial Economics and Technology Management at the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management. The thesis is a continuation of our specialization project completed in the fall of 2016.

We would like to express our sincere gratitude to our supervisors Professor Kjetil Fagerholt, Professor Henrik Andersson, and Postdoctoral Fellow Giovanni Pantuso for valuable discussions and constructive feedback. Your willingness to review and engage in our work has been an important contribution to the end result. We greatly appreciate your enthusiasm and involvement in our research.

Trondheim, 2017-06-06

Carl Axel A. Folkestad and Nora Å. Hansen

Abstract

Carsharing systems have emerged as an attractive form of transport in urban areas since the beginning of the 21st century. Effective carsharing systems carry significant environmental advantages as well as economic advantages for its users. Challenges faced by carsharing organizations can be modeled and solved using Operations Research (OR) methods in order to enhance system performance. As a result, carsharing has received increased attention in the OR community in the recent years. In this thesis, three gaps are identified in the existing carsharing literature: Operation of free-floating systems, repositioning under realistic conditions, and integrated routing of rental cars and operators to handle rental cars in repositioning operations.

Aiming to close the aforementioned gaps, the Static Free-Floating Electric Vehicle Carsharing Handling Problem (SFFEVCHP) is defined and modeled. In the SFFEVCHP, electric rental cars in a free-floating carsharing system are repositioned to charging stations when their battery level fall below a predefined threshold. The charging location decision takes balancing the distribution of cars in the business area into account. The rental cars are moved by operators that are transported between charging stations and rental cars by service vehicles. The problem includes several connected decisions. First, the optimal location to charge rental cars must be determined. Second, the routes of service vehicles transporting operators between rental cars in need of charging must be decided. The time and location to drop off an operator affects the pick up of the operator leading to temporal and spatial interdependencies between the routing of operators and service vehicles. Central to the problem is the trade off between the cost of not meeting demand due to a disadvantageous distribution of cars in the system and the cost of transporting operators to charge and reposition rental cars.

We first model the SFFEVCHP as a mixed integer program. The complexity of the problem requires an approximate solution method in order to solve real world instances. A Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) based on the work of Vidal et al. (2012) is therefore developed. To be able to use the HGSADC on this complex problem type, new chromosomes to represent individuals have been developed and the Genetic Algorithm (GA) operators are adapted to the problem. Also, a novel construction heuristic inspired by construction heuristics for the Dial-A-Ride problem is proposed.

To demonstrate the capabilities of the presented algorithm, 15 instances with sizes ranging from 100 to 200 rental cars in need of handling are solved. Ten algorithm runs are executed for each instance. The algorithm is able to solve all the instances with an average gap to the best known solution of 1.3 percent and a gap coefficient of variance of 0.9 percent in less than an hour. Furthermore, solutions with an average gap of 1.9 percent are found after ten minutes with a coefficient of variance of 0.9 percent. To demonstrate the value of repositioning in conjunction with handling, the algorithm is run while prohibiting repositioning. Comparing these results to the results with repositioning for an instance with 100 rental cars yield a net total cost improvement of 3.7 percent. As the total cost

captures the profit effect of the repositioning activities, this improvement can likely be directly transferred to the gross profit margin of the carsharing companies.

The results are a clear indication of the strength of the proposed algorithm. Realistic systems studied typically have 100 to 150 rental cars in need of handling at a given time. Hence, we are consistently able to solve realistic problem sizes yielding high quality solutions. Furthermore, the problem formulation combining repositioning with necessary charging and maintenance provide the opportunity for operators to realize the added profits of repositioning while only marginally increasing operational costs. We believe this presents a significant enhancement of existing models and algorithms. In addition to establish the effectiveness of the HGSADC for the SFFEVCHP, this thesis exhibits the capabilities of the algorithm for routing problems with complex synchronization constraints including spatial and temporal interdependencies.

Sammendrag

Siden begynnelsen av 2000-tallet har bildelingstjenester vokst frem som en attraktiv transportform i urbane strøk. I tillegg til økonomiske fordeler for sine brukere har effektive bildelingstjenester betydelige miljømessige fordeler. Utfordringer som bildelingsselskaper står overfor kan modelleres og løses ved hjelp av operasjonsanalyse. På bakgrunn av dette har bildeling fått økt oppmerksomhet fra forskningsmiljøet innenfor operasjonsanalyse de siste årene. I denne oppgaven har vi identifisert tre mangler i den eksisterende litteraturen på området: fritt-flytende systemer, reposisjonering av leiebiler under realistiske antakelser og integrert ruting av leiebiler og operatører for å håndtere leiebiler i reposisjoneringsoperasjoner.

Med mål om å adressere de ovennevte manglene definerer og modellerer vi det Statiske Fritt-Flytende El-bildelingstjeneste Håndteringsproblemet (Static Free Floating Electric Vehicle Carsharing Handling Problem - SFFEVCHP). I SFFEVCHP reposisjoneres elbiler i et fritt-flytende bildelingssystem til ladestasjoner når batterinivået faller under en predefinert terskel. Leiebilene flyttes av operatører som blir transportert til leiebiler og fra ladestasjoner av servicebiler. Problemet inneholder flere sammenkoblede beslutninger. For det første må den optimale lokasjonen for å lade hver elbil velges. Deretter må rutene til servicebilene som transporterer operatørene bestemmes. På grunn av at tidspunktet og lokasjonen en operatør slippes av påvirker hvor og når han/hun blir plukket opp, oppstår det gjensidige, geografiske og temporære avhengigheter når problemet skal løses. Sentralt for problemet er avveiningen mellom kostnaden som oppstår ved at selskapet ikke er i stand til å tilfredstille etterspørselen på grunn av en ufordelaktig fordeling av biler i systemet og kostnaden ved å gjennomføre reposisjonering.

I denne oppgaven modellerer vi først SFFEVCHP som et blandet heltallsproblem. Problemet er komplisert å løse, noe som gjør det nødvendig med en tilnærmet løsningsmetode for å løse virkelige problemstørrelser. Et hybrid genetisk søk med adaptiv mangfoldskontroll (Hybrid Genetic Search with Adaptive Diversity Control - HGSADC) basert på arbeidet av Vidal et al. (2012) er derfor utviklet. For å kunne bruke et HGSADC på denne kompliserte problemtypen har vi utviklet nye kromosomer for å representere individene. I tillegg er nye operatorer for de ulike modulene i den genetiske algoritmen presentert. Til slutt er en ny kontruksjonsheuristikk inspirert av kontruksjonsheuristikker for Dial-A-Ride problemet utviklet.

For å demonstrere kvalitetene til den presenterte algoritmen har tester på instanser med 100 til 200 leiebiler med behov for lading blitt gjennomført. Algoritmen har blitt kjørt ti ganger for hver instans. Resultatene viser at HGSADC-algoritmen er i stand til å løse alle instansene med et gjennomsnittlig gap til den beste kjente løsningen på 1,3 prosent og en variasjonskoeffisient av gapet på 0,9 prosent. Videre finner algorithmen løsninger med gap til beste kjente løsning på 1,9 prosent og variasjonskoeffisient på 0,9 prosent etter en kjøretid på ti minutter. For å tydeliggjøre verdien av å gjøre reposisjonering samtidig som flytting av biler til ladestasjoner har algoritmen blitt kjørt uten reposisjonering. Når disse resultatene sammenlignes med resultatene med reposisjonering ser vi en netto forbedring

av totalkostnaden på 3,7 prosent for en instans med 100 leiebiler. Siden totalkostnaden inneholder profitteffekten av reposisjoneringaktivitetene er det rimelig å anta at denne forbedringen kan overføres direkte til selskapenes driftsmargin.

Resultatene av denne rapporten viser tydelig styrken av den presenterte algoritmen. Realistiske bildelingssystemer har typisk 100 til 150 leiebiler med behov for lading på et gitt tidspunkt. Dermed er algoritmen i stand til å løse realistiske problemstørrelser med konsistente løsninger av høy kvalitet. I tillegg gjør muligheten til å reposisjonere biler samtidig som de flyttes til ladestasjoner at bildelingsselskaper kan realisere fordelene ved reposisjonering med kun en marginal økning i operasjonskostnadene. Vi anser dette som en betydelig forbedring over eksisterende modeller og algoritmer. Ved siden av å tydeliggjøre verdien av HGSADC-algoritmen for SFFEVCHP viser denne oppgaven potensialet til algoritmen på rutingproblemer med komplekse synkroniseringsrestriksjoner, inkludert problemer med gjensidige geografiske og temporære avhengigheter.

Table of Contents

List of Tables

List of Figures

Abbreviations

EV	=	Electric Vehicle
CSO	=	Carsharing Organization
DARP	=	Dial-A-Ride Problem
GA	=	Genetic Algorithm
HGSADC	=	Hybrid Genetic Search with Adaptive Diversity Control
MIP	=	Mixed Integer Program
OR	=	Operation Research
PIX	=	Periodic Crossover with Insertions
SFFEVCHP	=	Static Free Floating Electric Vehicle Carsharing Handling Problem
TSP	=	Traveling Salesman Problem
UHGS	=	Unified Hybrid Genetic Serach

Chapter 1

Introduction

Since the beginning of the 21st century, carsharing systems have emerged as an attractive means of transport in urban areas across the globe. A car used to be the ultimate symbol of freedom and independence. However, consumers are rethinking the value of ownership, starting to buy mobility instead of vehicles (Belk, 2014). Changed consumer behavior has opened the doors for carsharing systems that provide their users with the convenience and flexibility of ownership without the financial commitment, burden of maintenance, and parking space requirement of car ownership.

Changed consumer behavior is not the only driving force behind carsharing systems. According to Frost and Sullivan (2010), each shared vehicle replaces on average 15 personally owned vehicles and users of carsharing services drive 31 percent less than when owning a car. These two factors result in an annual reduction of 482,170 tons carbon dioxide emissions globally and less travel congestion in urban areas. Furthermore, a private vehicle is on average parked 23 hours a day and accounts for around 688,000 acres of land, representing 25 percent of urban surface occupancy, in the US (Li et al., 2016). Hence, large-scale carsharing adoption represents an opportunity for significant reductions in emissions and traffic congestion, in addition to more effective use of space in urban areas.

In this thesis, we define carsharing as short-term vehicle access among a group of members enabled by a third-party organization taking care of ownership, day-to-day maintenance and vehicle insurance. Carsharing organizations face many strategic and operational challenges. We focus on operational challenges, as these are crucial to create profitable systems. Some of these challenges are maintenance, like refueling/charging, cleaning and minor repairs, and repositioning of the rental cars. Repositioning is done to redistribute the fleet of rental cars to achieve a more favorable distribution to improve the operator's ability to meet customer demand. These operations need dedicated staff resulting in substantial costs for carsharing organizations. Efficient maintenance and repositioning can increase both the resource utilization of the operators and lead to greater customer satisfaction due to higher availability of rental cars. Ultimately, it is essential for the financial viability and sustainability of carsharing systems to find good solutions to these problems.

The evolution of carsharing systems has resulted in increased attention in the Operations Research (OR) community during the last years. However, the existing research fails to address the full complexity of the challenges under realistic conditions. Generally, the available models and algorithms either solve only parts of the problems or solve for system setups that do not resemble the operations of current state of the art systems.

The purpose of this thesis is to develop an optimization model and solution method that address charging and repositioning of rental cars in a carsharing system with a fleet of electrical vehicles. At a given point in time, the carsharing organization know the current distribution of rental cars, the state of charge of each car, and the availability of charging stations. The carsharing organization must then decide at which charging station to charge the rental cars, which operator to take care of each car, and the route of service vehicles transporting the operators between charging stations and rental cars. Deciding the assignment of rental cars to charging stations aiming to reach a more favorable distribution of cars in the system can significantly increase profitability of the carsharing organizations. Modeling the described problem is involved because the assignment of rental cars to charging stations and the routing of service vehicles transporting operators are closely linked. This results in spatial and temporal interdependencies because the pick up and drop off destinations of the operators, i.e. the routing of service vehicles, are dependent on the destination, handling order, and operator assigned to handle each rental car. As all these decisions are made by the model, the problem represents a complex class of vehicle routing problems.

This work is constituted of two main parts. First, a novel mathematical model is proposed, making the following decisions:

- Which charging station to charge each rental car while evaluating the distribution of rental cars in the system.
- Which operator to handle each rental car and in which order.
- The routes of service vehicles transporting the operators from charging stations to rental cars.

The problem is formulated as a *mixed integer program* (MIP). Due to the complexity of the problem, the MIP is computationally demanding to solve limiting the problem sizes solvable using a commercial solver. Consequently, a *hybrid genetic algorithm with adaptive diversity control* (HGSADC) is developed to be able to solve realistic problem sizes. The algorithm should be able to solve problems with up to 100 rental cars in need of handling. To our knowledge, this algorithm type has not been applied to vehicle routing problems with spatial and temporal interdependencies before. The main contributions of this thesis are thus:

- A novel model considering the full complexity of electric carsharing organizations' daily operations by integrating charging of rental cars and repositioning.
- An efficient algorithm capable of solving real-life problem sizes.
- A demonstration of the performance of Genetic Algorithms (GA) on vehicle routing problems with spatial and temporal interdependencies.

These contributions represent a substantial improvement of the available models and algorithms for carsharing systems. In addition, they advance the knowledge of heuristic approaches to routing problems with complex synchronization constraints.

The thesis is organized as follows. First, we give an introduction to the carsharing business worldwide in Chapter 2. Second, a detailed presentation of the problem studied is given in Chapter 3 followed by an extensive literature review on research related to carsharing operations in Chapter 4. Then, a mathematical model of the problem is proposed in Chapter 5 before the hybrid genetic algorithm with adaptive diversity control is described in Chapter 6. A computational study is presented in Chapter 7, while Chapter 8 discusses the practical use of the model. Finally, we conclude on the work of the thesis and suggest future research avenues in Chapter 9.

Chapter 2

Background

In this chapter, an overview of modern carsharing systems is given. First, common characteristics of carsharing systems are presented in Section 2.1. Then, the history of carsharing systems is briefly reviewed in Section 2.2.

2.1 Characteristics of Carsharing Systems

Carsharing is generally defined as short-term vehicle access among a group of members who share the use of a vehicle fleet that is owned, maintained, managed, and insured by a *Carsharing Organization* (CSO). The users are able to access the shared cars without interacting with the operator, i.e. reservations, pick ups, and returns are self-serviced. Carsharing services can be divided into two categories: *free-floating systems* and *station-based systems*. Free-floating systems enable users to pick up available cars and deliver them anywhere within a specified business area. In a station-based system, the cars are allocated at dedicated stations. A station-based system is either a *two-way* or a *one-way* system. In a two-way system, the user must pick up and return the car at the same station, while the user can pick up and return the cars at different stations in a one-way system. The fleet of a system can consist of both gasoline powered cars and/or electrical vehicles (EVs).

The main sources of revenue for a typical CSO are subscription fees from users and per minute or distance rates when renting a car. Examples from existing carsharing companies are given in Table 2.1. Most companies require monthly or yearly subscriptions. On top of the subscription fee, all companies researched charge a time-based rate. Some operators also practice distance-based rates. The costs borne by a CSO mainly consist of capital costs related to rental cars, service vehicles (i.e. vehicles used when travelling between rental cars to perform daily operations, e.g. maintenance) and rental stations, and operational costs. The operational costs are driven by staff costs to maintain, handle, and refuel/recharge cars. In addition, the CSO incur costs for fuel or charging when a vehicle is used, but this scales directly with usage and is thus a less important aspect of the operational costs of a carsharing company.

The revenues of a CSO are heavily dependent on the utilization of the rental car fleet.

Demand in different geographical areas may change throughout a day or week. In addition, the characteristics of an average trip may differ. E.g., trips originating in the city center may on average have shorter duration than trips originating in the outskirts of a city. Hence, forecasting and understanding these trends are important to operate the system efficiently. This may also motivate operators to employ staff to move cars between areas to achieve better utilization and thus higher revenues.

Table 2.1: Subscription fees and usage rates for ZipCar, Autolib and Car2Go, Autolib converted to approximate USD amounts. The information is retrieved from the respective companies' websites May 11th, 2017.

	ZipCar	**Autolib**	**Car2Go**
Subscription Fee	USD 7 per month	USD 11 per month	N/A
Duration rates	USD 7.75 per hour	USD 0.3/min (20 min. minimum)	USD 0.41 per min
Per kilometer rates	USD 0.45 per mile if trip exceeds 180 miles	N/A	N/A

Some operational challenges faced by a CSO are maintenance, refuelling/charging and cleaning. These challenges may be handled in numerous ways, for example by having dedicated staff that refuels/recharges and maintains cars, reimburse or incentivize users to handle it, or have payment solutions like fuel cards placed in the rental cars. Another challenge is *repositioning* of cars in one-way or free-floating systems, i.e. moving cars from one location to a more favorable one. After operating a while, the system may end up with an unfavorable distribution of cars. Some strategies to avoid this are *proactive approaches*, *user based repositioning* and *operator based repositioning*. These strategies may be used alone or combined. Proactive approaches try to avoid repositioning by denying unfavorable trips, i.e. a user have to specify its destination and if the trip is unfavorable, the CSO denies the reservation. In user-based repositioning the users reposition the cars either by being motivated by monetary compensation or by the CSO making users split or share trips. Operator based repositioning is done by designated CSO staff. It is common to distinguish between *static repositioning* and *dynamic repositioning*. In static repositioning, the relocation of cars is done when the carsharing system is closed or selected cars are made unavailable for users for repositioning purposes. In dynamic repositioning, the repositioning process is done while the system is operating, taking into account changes in e.g. traffic, demand, and distribution of rental cars during the handling and repositioning period.

2.2 Carsharing History

The following section is based on the article by Shaheen et al. (2015) elaborating on the history of carsharing.

Table 2.2: Lessons learned from early carsharing systems (Shaheen et al., 2015).

Service	Location	Operation dates	Lessons learned
Proctip	Montpellier, France	1971-1973	Failed due to lack of proper control systems and technological issues
Witkar	Amsterdam, Netherlands	1974-1986	Failed because of high costs, lack of governmental support, and technological limitations
Liselec Yélomobile	La Rochelle, France	Since 1993	Successful due to continued governmental support
Praxitéle	Saint-Quentin-en-Yvelines, France	1997-1999	Failed because of high costs and low demand
CarLink II	San Francisco Bay USA	2001-2002	Terminated after transfer from pilot to third-party operator due to financial concerns; limited scale
UCR IntelliShare	University of California, Riverside, USA	1999-2010	Successful due to advanced technologies and support from agencies and industry
Honda DIRACC	Singapore	2003-2008	Terminated due to declining service quality

Carsharing is not a new phenomenon. The first known carsharing organization "Sefage" started in Zurich, Switzerland, in 1948. The motivation for early carsharing was economic benefits, aiming to increase people's welfare. Starting in the 1980s carsharing was increasingly fuelled by environmental concerns. Since the 1990s pilot programs have been launched all over the world and the subject has been getting increased attention in academia. Lately, the main research focus for carsharing organizations have been one-way station-based systems, free-floating systems and systems with EV fleets.

Though there have been many attempts, the first nation to reach a successful carsharing system was Switzerland with "Mobility Car Sharing Switzerland", founded in 1997. This system was a merger between two earlier systems, AutoTeilet-Genossenschaft (ATG) and ShareCom. Almost all of the earliest programs closed operation after a few years. The reasons for failure could be many; overly ambitious projects given available technology at the time, inadequate planning and marketing, or lack of support from local governments. Table 2.2 gives an overview of lessons learned from early carsharing systems.

Carsharing first gained its popularity in Europe, which still accounts for the majority of members and number of cars. Lately carsharing has become increasingly popular in North America. The first US adopters were motivated rather by convenience than economic or environmental reasons, possibly due to lower driving costs in the US. In Asia, carsharing systems have gained popularity more recently, especially in Japan and Singapore. The main focus of CSOs in Japan is business use, while the focus is more shifted towards households in Singapore (Correia and Antunes, 2012). Figure 2.1 shows the development of carsharing members and shared vehicles between 2006 and 2014.

Since the beginning of carsharing the information technologies have evolved dramat-

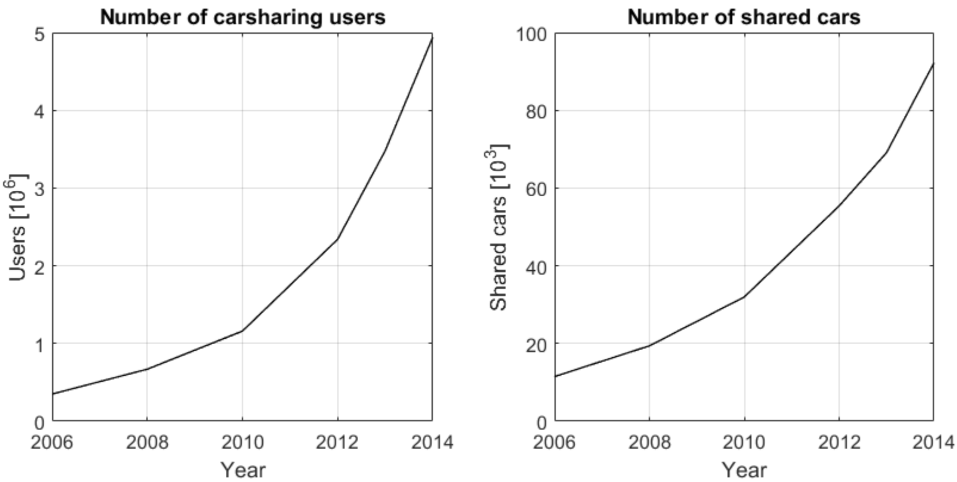

Figure 2.1: The evolution of carsharing members and number of shared cars from 2006-2014 (Frost and Sullivan, 2014)

ically, facilitating modern carsharing systems. Technological limitations that made early carsharing systems fail do not longer apply and CSOs experience an increased governmental support due to the environmental benefits. These factors have resulted in a bloom of carsharing systems worldwide the last few years. Challenges faced by a carsharing system can be modeled and solved using mathematical programming in order to enhance performance to create profitable systems. As a result, carsharing has become an interesting Operation Research (OR) topic.

Chapter 3

Problem Description

In this chapter, the Static Free-Floating Electric Vehicle Carsharing Handling Problem (SFFEVCHP) is defined. The system considered is free-floating, but cars are charged at fixed charging stations. Central to the problem is the trade off between the cost of not meeting demand due to a disadvantageous distribution of cars in the system and the cost of transporting operators to charge and reposition rental cars. Figure 3.1 shows the conceptual principles of the model solving the SFFEVCHP. Necessary definitions of the problem are described in Section 3.1. Furthermore, detailed descriptions of the problem specifics are introduced in Section 3.2. The chapter concludes with a summary of the problem in Section 3.3 and an illustrative example in Section 3.4.

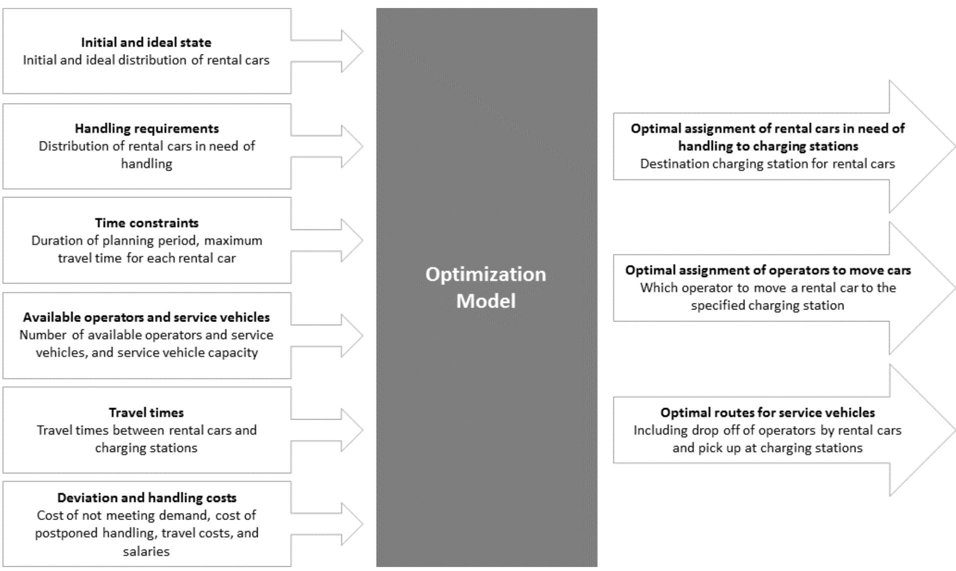

Figure 3.1: Conceptual overview of the model solving the SFFEVCHP, inputs on the left side and outputs on the right side.

3.1 Definitions

When formulating the SFFEVCHP two types of cars are considered: *rental cars* and *service vehicles*. Rental cars are the EVs available for customers of the CSO, while cars used to transport operators within the business area are denoted service vehicles. In practice these vehicles could be equal, but because they are assigned to different tasks they are separated. Rental cars with battery level lower than a pre-specified threshold must be charged. When a rental car needs charging, an operator *handles* the rental car from its original position to a charging station. The *planning period* is the total time available for handling the fleet of rental cars, typically between one and four hours.

The rental cars are charged at charging stations. Each charging station has a given number of available charging slots and is assigned a *surrounding area*. From here on, the surrounding area is included when discussing charging stations. Only one rental car can be handled to each available charging slot within a planning period. If a rental car occupies a charging slot at the beginning of a planning period, the charging slot will be marked as available if the battery level of the rental car is above a given threshold. An operator can then unplug the rental car when handling another car to the given charging station.

To be able to quantify the distribution of rental cars in the system in order to make rebalancing decisions, the concept of states are introduced for each charging station, i.e. the charging station and its surrounding area. The *initial state* describes the number of rental cars available for customers at the charging station when the planning period starts, i.e. all rental cars at the charging station or in the surrounding area with a battery level above a given threshold. The *ideal state* is the ideal number of rental cars at the charging station with a battery level above the threshold. After solving the problem, the *final state* is reached. The final state equals the initial state in a charging station plus the number of rental cars handled to the station. As rental cars moved to a charging station becomes available for customers after a given time, they are counted in the final state. The ideal and final state are not necessarily equal. The problem at hand is static, meaning that all input data is kept constant throughout the whole planning period.

3.2 Problem Specification

In this section the specifics of the considered problem are presented. Section 3.2.1 describes the objectives of the CSO. Furthermore, Sections 3.2.2 to 3.2.5 describe details of the routing of service vehicles, handling of rental cars, service vehicle and operator capacity, and time usage, respectively.

3.2.1 Objective

The objective of the problem is to minimize the cost of handling rental cars, the cost of postponing handling, and the cost of deviations from the ideal state at each charging station. The cost of handling cars includes the cost of transporting operators with service vehicles to the rental cars in need of charging, and the cost of transporting operators from charging stations back to the depot.

The CSO incurs deviation costs when there is a deviation between the ideal and final state at a charging station and the surrounding area at the end of the planning period. These costs represent lost potential revenue. Rental cars can be charged at a charging station even though there are more rental cars than ideal at the station. Hence, the deviation can both be positive and negative. If the deviation is positive it is expected that the CSO will not be able to meet demand in the period following the planning period and if the deviation is negative it is expected that the rental car will be underutilized. Therefore, both too few and too many cars in the final state are penalized.

A penalty cost is incurred if handling of a rental car is postponed, i.e. not performed within the given planning horizon. This is to capture the trade off between added travel cost and not fulfilling handling needs. At last, a fixed cost is added for each service vehicle and operator used to handle the fleet of rental cars.

3.2.2 Routing

The SFFEVCHP consists of two integrated routing problems: the routing of rental cars to charging stations, and the routing of service vehicles. The CSO has to decide at which charging station to charge a rental car. In addition, the CSO has to assign operators to handle cars. Operators do not necessarily have to be picked up by the same service vehicle that dropped them off, but service vehicles are the only available means of transport. Furthermore, service vehicles are allowed to wait at charging stations if necessary to pick up operators transporting rental cars to that station. Operators can handle multiple rental cars during the planning period and both service vehicles and operators are allowed to visit the same charging station several times.

3.2.3 Handling of Rental Cars

The CSO has to decide whether a rental car should be handled in the current planning period or postponed. There must be an available charging slot if a car is handled to a charging station. Cars with a battery level above a threshold at the charging station are considered to not take up a slot, as the operators can move the car upon arrival. Finally, the time each rental car can travel is limited based on the current battery level.

3.2.4 Service Vehicle and Operator Capacity

A given number of service vehicles are available to transport operators. Each vehicle has a fixed capacity to carry operators. This capacity cannot be exceeded on any route driven by the service vehicle. Also, there is a given number of operators available that can be assigned to handle rental cars. A fixed cost is associated with each service vehicle and operator used.

3.2.5 Time Usage

Available time to handle rental cars is limited. Service vehicles and rental cars use the same amount of time to travel between rental car positions and charging stations. The time required to plug in the rental car and potentially move fully charged rental cars from the charging slot is omitted, as it is moved to the nearest available parking spot.

3.3 Summary of the SFFEVCHP

The SFFEVCHP is a complex problem having both spatial and temporal interdependencies. The decisions that must be made are the following:

- Assignment of rental cars in need of handling to charging stations, taking the cost of postponing handling and deviations from the ideal state in each charging station into account.
- Assignment of operators to move rental cars to the specified charging station.
- Routes for service vehicles, including drop off and pick up of operators such that all rental cars not postponed are handled.

3.4 Example Problem

Figures 3.2 to 3.4 illustrate an example of the SFFEVCHP. The example case includes four charging stations and five rental cars in need of handling. The planning period is two hours, and service vehicles have capacity to carry four operators. Furthermore, two service vehicles and eight operators are available at the depot. Travel times, deviation cost, and transportation cost are taken as given. The values of these parameters are not important in the given example, as the problem is not solved to optimality. Rather, a feasible solution is given for illustrative purposes.

The initial state of the system is shown in Figure 3.2. For each charging station (including the adjacent area), the parentheses show the current available number of charging slots, the initial state, and the ideal state. The status of each rental car is illustrated using color codes. Also, dotted lines indicate the area appointed to each charging station. In the initial state, there is a total of eight deviations from the ideal state, counting both positive and negative deviations.

Figure 3.3 shows a feasible solution to the problem. The planning period stars at 13:00, lasts two hours and ends at 15:00. Arrival times at different cars and charging stations are indicated at the arrowheads. In the given solution, rental cars *1* and *2* are charged at charging station *1*, car *3* at charging station *2*, and cars *4* and *5* at charging station *3*. The first service vehicle leaves the depot carrying three operators. First, the vehicle drives to rental car *4* and drops off one operator. Then, the vehicle drives to car *5* and drops off another operator before the last operator is dropped off at car *3*. The service vehicle proceeds to charging station *3* to pick up two operators and then to station *2* to pick up one more operator. The operator handling car *3* is arriving at the charging station two minutes after the service vehicle. Hence, the service vehicle has to wait. Finally, the service vehicle drives to the depot to end the route. The second service vehicle leaves the depot carrying two operators. The first operator is dropped off at rental car *2* and then the second operator is dropped off at rental car *1*. The service vehicle then drives to charging station *1* to pick up the two operators and then drives to the depot to end the route.

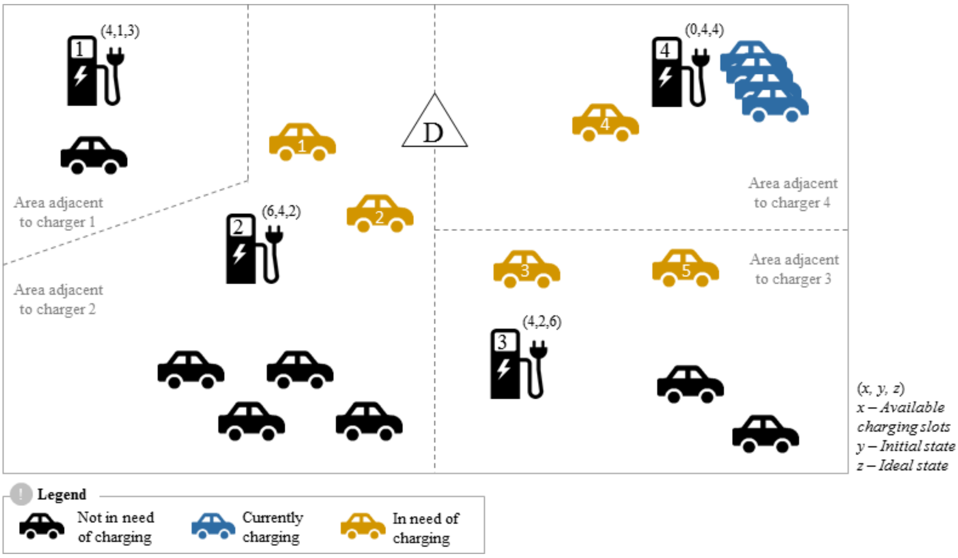

Figure 3.2: Initial state: Five cars need handling. Total deviations from the ideal state equal eight.

After the planning period is over, the system is in its final state, shown in Figure 3.4. All cars in need of handling are handled, and the solution reduces the imbalance in the system by three, from eight to five total deviations from the ideal. In this solution, all cars are handled and a more favorable distribution of the fleet of cars is achieved.

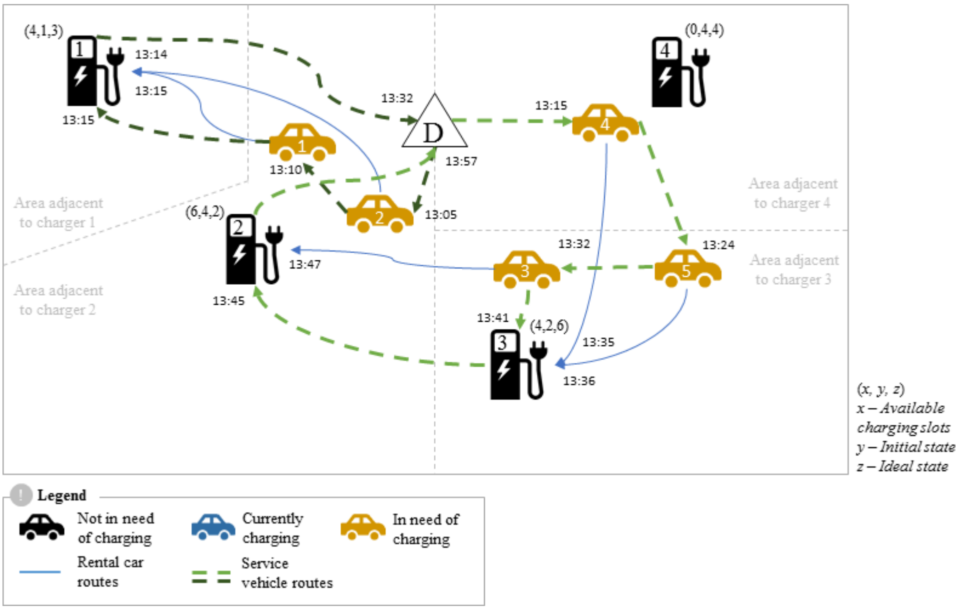

Figure 3.3: Feasible Solution: Two service vehicles and eight operators are available, planning period of two hours.

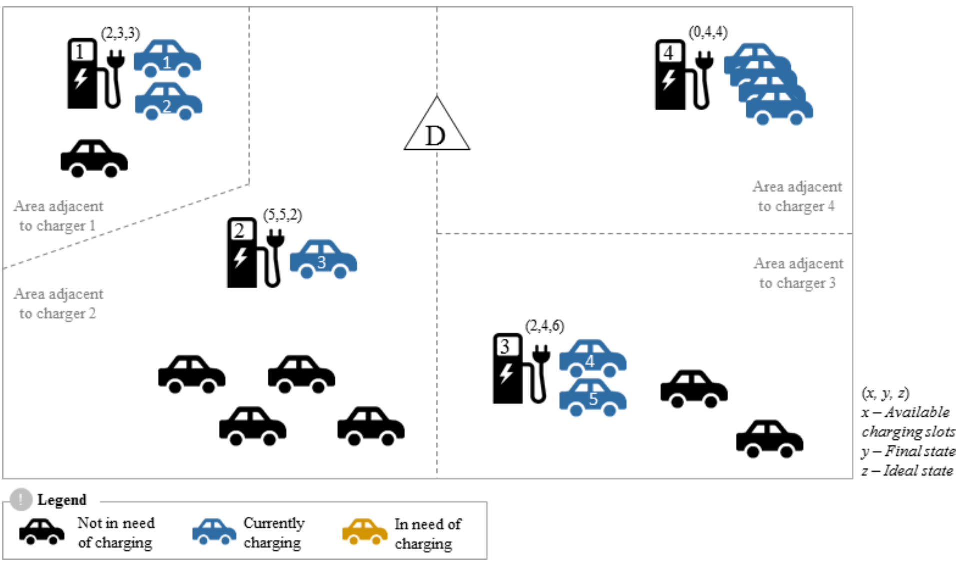

Figure 3.4: Final state: All cars handled, total deviations from the ideal state equal five.

Chapter 4

Literature Survey on Carsharing

This chapter elaborates on the literature on operational problems faced by carsharing organizations, specifically repositioning, recharging, and refueling. The SFFEVCHP has similarities with repositioning in station-based carsharing systems, as the objective is to reposition rental cars to achieve a favorable distribution while minimizing the repositioning cost. Hence, the main focus of this literature survey is vehicle repositioning, discussed in Section 4.1. Operational perspectives of recharging and refueling of carsharing systems are often not the subject of optimization models, but some literature found concerning the subject is presented in Section 4.2. Finally, Section 4.3 motivates the scientific contribution of this thesis.

4.1 Vehicle Repositioning

In this section, literature on the three different repositioning approaches introduced in Chapter 2 are discussed. Operator-based repositioning presented in Section 4.1.1 is the most relevant for the SFFEVCHP and hence the main focus of this chapter. User-based repositioning and proactive approaches are discussed in Section 4.1.2 and 4.1.3, respectively. These methods are potential extensions of the SFFEVCHP that can further enhance the performance of carsharing systems.

4.1.1 Operator-Based Repositioning

The problem of finding the optimal repositioning of rental cars in a carsharing system has become an interesting topic of research the last years. The literature contains several MIP optimization models trying to solve the problem as well as simulation models analyzing the performance of different repositioning techniques. The main focus is often to determine how many cars to reposition between stations in station-based systems, but some literature also takes routing of service vehicles and operators into account.

Kek et al. (2006) conduct a qualitative analysis on operator-based repositioning to provide insights on key issues. To support the analysis, a simulation model is developed. The simulation model proposed is used to evaluate the impact of different repositioning techniques. Using it for optimization would however be impractical due to the large number of required simulation runs. Kek et al. (2009) propose a three-phase optimization-trend-simulation decision system for carsharing operators, using the simulator from Kek et al. (2006) as an evaluation tool. A MIP optimizer outputs the lowest cost resource allocation, staff activities, and repositionings. Then, a trend filter receives the optimized outputs and "filters" them through a series of heuristics to obtain a set of recommended operating parameters like shift hours and repositioning technique. The main contributions of the research by Kek et al. (2009) are the formulation of the vehicle repositioning problem as a MIP problem in phase one and the introduction of heuristics that convert the optimal solution to more practical parameters in phase two.

Jorge et al. (2014) solve a classic transportation problem to determine repositioning of rental cars between stations, assuming apriori knowledge of demand. A simulation model that determines optimal repositioning using a minimum cost network flow algorithm is also introduced to study different real-time repositioning policies. By integrating the repositioning policies with the optimization model, the results of the simulation can be improved.

Boyaci et al. (2015) develop and solve a multi-objective MIP model taking electrical vehicle charging requirements into account. To cope with the large number of repositioning variables, an aggregated model using the concept of a virtual hub is introduced allowing the problem to be solved with a branch-and-bound approach. Multi-objective MIP optimization is also used by Boyaci et al. (2017), who develop an optimization framework involving three mathematical models. The first is station clustering to cope with the dimensionality of the operational problem, the second is an optimization of operations including repositioning of vehicles, and the third is optimizing the flow of personnel. The objective is to maximize trips served, minimize repositioning cost, and maximize charging time for the cars. A simulator tests the feasibility of the optimization outcome in terms of vehicle recharging requirements, and the model is solved iteratively adding the new constraints restricting cars that need further charging until the results are feasible.

The focus of some studies is on system optimization under uncertainty. Fan et al. (2008) formulate a multistage stochastic linear integer model with recourse that account for system uncertainties, with the objective of maximizing CSO profits. A Monte Carlo sampling-based stochastic optimization method is used to solve the problem. The output of the model is the number of cars the operator should reposition at the end of a given day and to what stations to move these cars. Nair and Miller-Hooks (2011) develop a stochastic MIP involving joint chance constraints that minimizes repositioning cost under demand uncertainty. This stochastic MIP has a non-convex feasible region. To overcome this, a divide-and-conquer algorithm for generating p-efficient points is used to transform the problem into a set of disjunctive, convex mixed integer programs, and handle dual bound chance constraints. The model decides how many rental cars to be moved from one zone to another.

The work presented by Kek et al. (2009) and Boyaci et al. (2017) take operator activities in a repositioning process into account. In addition, staff must be routed between stations due to the staff imbalance that results from repositioning cars. Nourinejad et al. (2015) solve the joint optimization of car and staff repositioning by introducing two integrated multi-traveling salesman formulations. One of the TSPs represents the car repositioning and the other the staff relocation. The objective is to find the optimal set of vehicle and staff repositioning tasks to minimize operator costs. Bruglieri et al. (2017) introduce four heuristics that can be applied to a MIP formulation of the car repositioning problem in a one-way carsharing system based on operators moving between rental cars and charging stations using a bike. Two of the proposed heuristics are greedy heuristics while the other two are structured heuristics that exploit some general properties of the feasible solutions.

Research on algorithmic solution methods for the repositioning problem is limited. Herbawi et al. (2016) is the only contribution found addressing algorithmic solutions for repositioning in free-floating systems. Herbawi et al. (2016) model the repositioning problem as a generalization of the pick up and delivery problem. The objective is to maximize the number of cars repositioned within a given time limit. The repositioning is done by multiple operators transported by a single service vehicle. To solve the problem, an evolutionary algorithm outputting the route of the service vehicle is proposed.

To enable dynamic repositioning throughout a day, future demands has to be known for very small time steps and carsharing demand models thus have to be reliable. Weikl and Bogenberger (2013) develop an integrated two-step model for optimal rental car repositioning for a free-floating system. Their main focus is an offline demand module carried out periodically to identify periodically repeating spatial-temporal demand patterns within a specific business area. A second online optimization module is carried out several times a day to determine optimal repositioning operations to minimize repositioning costs and penalty costs for not meeting demand. Repoux et al. (2015) introduce an event-based simulator to understand demand patterns and explore repositioning possibilities. Based on the simulation they develop a new repositioning strategy to minimize demand loss due to rental car unavailability. The strategy is to update repositioning plans with a MIP optimization framework that utilizes the current state of the system and partial knowledge of near future demand based on reservations. The optimization framework decides which repositionings to perform and assigns necessary personnel. Choosing what cars to move correspond to finding the shortest repositioning path between two distributions, i.e. reaching the ideal distribution of cars with minimum effort and cost.

4.1.2 User-Based Repositioning

Cepolina and Farina (2012) present a carsharing system relying solely on user-based repositioning. When users are leaving the city center with a rental car, they are assumed indifferent to which station they use to leave the car, as all stations are public transportation hubs considered equal. Therefore, a system administrator can balance the system by deciding which station users should return their cars to. Another way of influencing the users to achieve better balance in the system has been explored by Kaspi et al. (2014). The paper explores the effects of parking reservations in one-way systems. The proposed scheme requires the user to provide his destination. This information is then used to alert users if there are no available parking spaces in their desired destination and to suggest alternative stations. Hence, the parking reservation policy ensures that there will be available parking space at the destination, reducing the time users have to spend travelling to other stations. Although the approach is not strictly a repositioning strategy, the scheme can have positive effects on the system balance, especially if users are directed to close-by stations based on the current distribution of cars.

More targeted schemes are proposed by Weikl and Bogenberger (2013). First, pricing incentives are presented. By reducing the price for trips with destination in areas that have too few cars, or even make them free, users can be inclined to move cars, reducing the amount of repositioning required by the operator. Second, trip sharing and trip splitting are introduced. The idea is that users can drive together in the same car if they have similar destinations and they are travelling from a high demand area. If two users are traveling together to an area with too few cars from a low demand area, the system manager can ask the users to split the trip instead. These schemes can reduce the risk of the CSO not meeting demand in some areas. Trip splitting and trip sharing are likely to be combined with pricing incentives in order to make it attractive for users.

4.1.3 Proactive Approaches

In the literature, schemes aiming to reduce the need for repositioning are proposed. Fan et al. (2008) introduce an optimization model maximizing operator profits in a one-way station-based system where the operator is allowed to deny trips if they are not profitable or if there are no cars available. Building on this concept, Correia and Antunes (2012) investigate how the distribution of cars in the system develops when three different trip selection criteria are used. Based on this they present a station location optimization model maximizing operator profits. The first scheme proposed assumes that the CSO has total control of trip selection and is free to accept or reject requested trips. Hence, a trip is only accepted if it contributes to increasing the profits of the operator. The second scheme is the standard approach found in most literature; accepting all trip requests. The third scheme proposed is a hybrid, that only allows denying trips if the pick up station has no available cars. When applied to a case study from Lisbon, Portugal, allowing the CSO to accept and deny trips yielded significant improvements in operator profits. Furthermore, both partial and full control of trip selection yielded positive profits for most cases, in contrast to the scheme where all trips must be accepted.

Building on Correia and Antunes (2012), Correia et al. (2014) assess the added value of accounting for users' flexibility in one-way fixed station systems. In addition to the trip selection schemes developed by Correia and Antunes (2012), the model proposed allows users to pick up cars at the second closest, and even third closest station from their origin and destination. Combined with real time information about the number of cars at each station, accounting for user flexibility allows the system to be more efficient. Boyaci et al. (2017) build on this concept and include user flexibility in a repositioning optimization model. In the model, all stations within a 500 meter radius are bundled together, i.e. making all the stations in one cluster available to serve demand at any of the stations. The results of the paper indicate that accounting for user flexibility can have a significant effect on the need for repositioning of cars.

4.2 Refueling and Recharging

Li et al. (2016) propose an optimization model that aims to determine optimal station locations and fleet size of a one-way fixed station carsharing system with electric vehicles.

The paper takes charging of the EVs into account. Charging is assumed to be done at the stations, and the model keeps track of the battery level of each car, demanding that each car is sufficiently charged before it can be booked by users. Similarly, Cepolina and Farina (2012) assume charging at stations and make cars that do not have sufficiently charged batteries unavailable for users. Kuhne et al. (2016) extend the emphasis on charging introduced by Cepolina and Farina (2012) and Li et al. (2016) by incorporating different charging speeds, i.e. fast and regular charging with limited capacity of fast chargers. The objective of the model is to determine optimal locations of stations and fleet size based on the availability of fast and regular chargers.

An approach for refueling is presented in the work of Santos and Correia (2015). The paper describe a new MIP optimization model designed as an upgrade of the work by Kek et al. (2009), to manage repositioning and maintenance operations of a one-way carsharing system in real time. The three main upgrades of the model is that it is prepared to be used in a rolling horizon approach, that it allows trip joining, meaning that staff can travel together in the same car, and that it is prepared to consider two maintenance procedures (including refueling) with different time durations. The model is able to decide the best schedule for each staff member.

4.3 Conclusions and Motivation of the Thesis

The SFFEVCHP addresses three gaps in the existing literature: Operation of free-floating systems, repositioning under realistic conditions, and integrated routing of rental cars and operators in repositioning operations. Table 4.1 provides an overview of the 19 articles surveyed in this literature review. It is evident that one-way carsharing systems have received increased attention in the literature the last ten years. We believe this is due to the attractiveness of these systems to the users compared to two-way systems, while at the same time causing significant operational complexities for the CSOs. A free-floating system provides even greater flexibility. However, there is little research on how such systems can be operated efficiently and hence be economically viable for CSOs. Only two of the 19 articles surveyed are concerned with free-floating systems.

In the literature, repositioning is proven to increase the efficiency of carsharing systems and increase CSO profits, even though it is a costly procedure. The gains of repositioning may be lost to increased operating costs if repositioning procedures are suboptimal, making CSOs reluctant to perform repositioning. Unlike repositioning, recharging and refueling of the fleet of rental cars are strictly necessary. We believe that combining necessary daily operations like recharging with repositioning give more realistic repositioning conditions for CSOs. That way, the full benefits of repositioning may be realized while only marginally increasing the operational costs.

Finally, few of the surveyed articles include integrated routing of rental cars and operators. Bruglieri et al. (2017) propose a model for a fixed station system with electric vehicles, with charging being performed at the stations. Operators travel between stations with folding bicycles which may be infeasible for realistic business areas. Similarly, Boyaci et al. (2015) and Boyaci et al. (2017) propose a model for a station-based electric one-way system. However, routing of rental cars and operators are performed separately, failing to account for the trade off between the costs and additional revenue of repositioning. Nourinejad et al. (2015) address the trade off between repositioning costs and gains by including both routing of rental cars and operators in the same problem. However, only repositioning under the assumption that the CSO must fulfill all demand is performed, with no attention on daily operations like recharging. Herbawi et al. (2016) propose an evolutionary algorithm routing both operators and a service vehicle transporting the operators. Nevertheless, the algorithm only considers one service vehicle, which restricts the routing possibilities greatly. In the systems studied in this thesis, multiple service vehicles are used for repositioning procedures. Finally, Santos and Correia (2015) present a model for daily maintenance, refueling, and repositioning that considers routing of both rental cars and operators. The cars being refueled must be moved back to the original spot, which is unrealistic for an all electric system and excludes the opportunity to perform repositioning in conjunction with daily operations.

Consequently, the available literature fails to provide a model that determines the routing of both rental cars and operators that can be applied to free-floating systems where repositioning is performed in conjunction with daily operations. The SFFEVCHP addresses this gap.

Table 4.1: Comparison of articles included in literature review. System costs include both operating costs for the CSO and inconvenience cost for the customers.

Authors (year)	Objective Function	Model Formulation	System Configuration	Electric Vehicles
Correia and Antunes (2012)	Maximize CSO profit	MIP	One-way	No
Boyaci et al. (2015)	Maximizing system profit	Multi-objective MIP	One-way	Yes
Li et al. (2016)	Minimize system cost	Continuum Approximation	One-way	Yes
Kek et al. (2006)	Maximize resources and enhance service levels	Simulation	One-way	No
Kek et al. (2009)	Minimize system cost	MIP, simulation	One-way	No
Jorge et al. (2014)	Maximize CSO profits	MIP	One-way	No
Bruglieri et al. (2017)	Maximize CSO profits	MIP, heuristics	One-way	Yes
Boyaci et al. (2017)	Maximize trips served, minimize repositioning cost, maximize charging time	Multi-objective MIP	One-way	Yes
Fan et al. (2008)	Maximize CSO profits	Multistage MIP	One-way	No
Nair and Miller-Hooks (2011)	Minimize repositioning cost	MIP	One-way	No
Nourinejad et al. (2015)	Minimize CSO cost	Multi-TSP	One-way	No
Weikl and Bogenberger (2013)	Minimize system cost	Unknown	Free-floating	No
Repoux et al. (2015)	Minimize trips lost due to unavailable cars	Simulation	One-way	Yes
Santos and Correia (2015)	Minimize cost of repositioning, loss of demand and penalty for not fulfilling maintenance	MIP	One-way	No
Cepolina and Farina (2012)	Minimize system cost	Nonlinear MIP, heuristics	One-way	Yes
Correia et al. (2014)	Maximize CSO profits	MIP	One-way	No
Kuhne et al. (2016)	Minimize system cost	MIP	One-way	Yes
Kaspi et al. (2014)	N/A	Simulation	One-way	No
Herbawi et al. (2016)	Maximize number of repositionings	Evolutionary algorithm	Free-floating	No

Chapter 5

Mathematical Model

This chapter presents a mathematical formulation of the Static Free-Floating Electric Vehicle Carsharing Handling Problem (SFFEVCHP). Section 5.1 presents modeling assumptions, while Section 5.2 defines the applied notation. Furthermore, the mathematical formulation of the problem is described in Section 5.3. Finally, symmetry breaking constraints and valid inequalities that can be employed to improve computation time are proposed in Section 5.4.

5.1 Modeling Assumptions

The key assumptions made when modeling the SFFEVCHP are discussed in this section. The assumptions presented aim to make the model general and applicable for any CSO managing a fleet of EVs. Sections 5.1.1 - 5.1.3 addresses assumptions about nodes and states, routing and handling, and time usage, respectively.

5.1.1 Nodes and States

To formulate the problem at hand, nodes are used to represent charging stations and their corresponding surrounding area, rental cars, and the depot. Recall from Chapter 3 that each charging station is associated with a surrounding area, and that this area including the charging station has an ideal and an initial state. Charging stations can be visited multiple times by both service vehicles and operators. If a rental car is visited, it has to be handled. Hence, a rental car can only be visited once. Both service vehicles and operators start and end up in the depot. Service vehicles can not make intermediate visits to the depot. Therefore, depots have two possible visits.

5.1.2 Routing and Handling

Only rental cars in need of handling are considered as a part of the SFFEVCHP. Service vehicles drive directly between nodes. This assumption can be made without loss of generality as detours would incur extra costs without added benefits. Operators can only be dropped off in nodes representing rental cars or the depot, and picked up in nodes representing charging stations or the depot. A rental car can only be handled by one operator.

5.1.3 Time Usage

Time is continuous and monotonically increasing for every transport route or handling route of the system. The time it takes to drop off and pick up operators is assumed included in the travel times.

5.2 Defining the Notation

This section describes the sets, indices, parameters, and variables used to model the SF-FEVCHP in detail. An overview of the applied notation is shown in Table 5.1. First, sets and indices are described in Section 5.2.1. Furthermore, the parameters and the variables are described in Section 5.2.2 and Section 5.2.3, respectively.

5.2.1 Sets and Indices

The problem at hand is defined over a set of nodes, $\mathcal{N}$. $\mathcal{N}$ can be divided into three disjoint sets $\mathcal{N}^{CS}$, $\mathcal{N}^{EV}$, and $\{0\}$, representing charging stations, rental cars in need of handling, and the depot, respectively. Figure 5.1 shows an overview of the set of nodes. The set $\mathcal{M}_i$ represents all possible visits a service vehicle or operator can make to node i. If a service vehicle visits node i, the visit is denoted by (i, m). $m = 1$ for the first visit, $m = 2$ for the second visit, etc. To index visits m, n, and k are used for nodes i, j, and k, respectively. The same notation applies for visits by operators, except the indices a, b, and c are used for visits to node i, j, and k, respectively. $\mathcal{V}$ and $\mathcal{D}$ are the set of available service vehicles and operators, respectively, and the indices v and d are used to describe the elements of the sets.

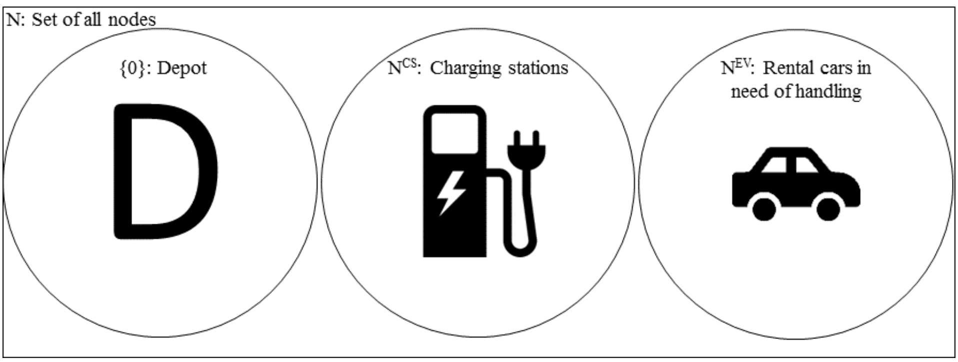

Figure 5.1: Overview of sets of nodes.

5.2.2 Parameters

The cost of deviation from the ideal state is C_i^E in node i for each unit deviation. Recall that nodes representing charging stations also includes the surrounding area of the charging station. C_i^{PH} is the costs of postponed handling of rental car i. C^V and C^D are the fixed cost of using a service vehicle and an operator, respectively. C_{ij}^T denotes the travel cost between two nodes i and j. The travel time between two nodes i and j is represented by T_{ij}, and T_i^{EV} denotes the maximum travel time for the rental car in node i, i.e. the range of the EV. The time limit for the planning period is $\overline{T}$. N_j^{CSP} gives the number of available charging slots in node j. The upper limit on the number of operators a service vehicles can transport at any given time is Q. Finally, S_j^0 and S_j^I are parameters representing the initial state and the ideal state of node $j \in \mathcal{N}^{CS}$, respectively. There are no states associated with nodes representing rental cars or the depot.

5.2.3 Variables

To track the flow of service vehicles and operators, a set of binary variables are used. Figure 5.2 shows the flow of service vehicle 1 and operators 1, 2, and 3 from the example in Chapter 3. x_{imjnv} is the arc flow variable for service vehicles, taking the value 1 if service vehicle v drives directly from visit (i, m) to visit (j, n), 0 otherwise. If a service vehicle v transports an operator d between its visit (i, m) to visit (j, n) on the operator's visit (i, a) to visit (j, b) $f_{imajnbvd}$ equals 1, 0 otherwise. The variable q_{ivd} is 1 if service vehicle v drops off operator d in node i, 0 otherwise. Similarly, g_{jnbvd} is 1 if service vehicle v on visit (j, n) picks up operator d on its visit (j, b), 0 otherwise. Furthermore, h_{ijbd} equals 1 if operator d handles a rental car from node i to visit (j, b), 0 otherwise. The continuous variables t_{imv}^V and t_{iad}^D gives the time of arrival for service vehicle v to visit (i, m) and the arrival time for operator d to visit (i, a), respectively.

If the handling of rental car i is postponed, the binary variable z_i^H equals 1, otherwise 0. The integer variable y_j gives the absolute value of the deviation between the ideal state and final state in charging station j. Finally, the binary variables s_v and w_d equal 1 if the service vehicle v and operator d are used, respectively, and 0 otherwise.

5.3 Model Formulation

This section presents the formulation of the SFFEVCHP as a MIP. First, the objective of the SFFEVCHP is presented in Section 5.3.1. Then, the constraints defining the problem are described in Sections 5.3.2 to 5.3.7. The constraints are divided into sections where each section covers similar constraints.

5.3.1 Objective

The objective function (5.1) consists of five terms. The first term accounts for the cost of deviations from the ideal state. The second term calculates the total travel cost for all service vehicles. Furthermore, the third term accounts for the penalty cost that incur if handling is postponed. Finally, the two last terms account for the fixed cost that incur if

Table 5.1: Sets, indices, parameters, and variables used to formulate the problem mathematically

Sets	
$\mathcal{N}$	Set of all nodes
$\mathcal{N}^{CS}$	Set of all charging stations, $\mathcal{N}^{CS} \subset \mathcal{N}$
$\mathcal{N}^{EV}$	Set of rental cars in need of handling, $\mathcal{N}^{EV} \subset \mathcal{N}$
$\mathcal{M}_i$	Set of all possible visits to node i
$\mathcal{V}$	Set of all service vehicles
$\mathcal{D}$	Set of all operators
Indices	
i, j, k	Node *i, j, k* $\in \mathcal{N}$
a, b, c	Operator visit *a,b,c* to node *i*, $a, b, c \in \mathcal{M}_i$
m, n, o	Service vehicle visit *m,n,o* in node *i*, $m, n, o \in \mathcal{M}_i$
v	Service vehicle $v \in \mathcal{V}$
d	Operator $d \in \mathcal{D}$
Parameters	
N_j^{CSP}	Number of available charging slots at charging station *j*
C_j^E	Deviation cost in charging station j
C_{ij}^T	Travel cost between node i and j
C_i^{PH}	Cost of postponed handling of rental car in node i
C^V	Fixed service vehicle cost
C^D	Fixed operator cost
T_{ij}	Travel time between node i and j
T_i^{EV}	Max travel time for rental car in node i
$\overline{T}$	Time limit for the planning period
Q	Service vehicle capacity
S_j^0	Initial state at charging station j
S_j^I	Ideal state at charging station j
Variables	
x_{imjnv}	1 if service vehicle *v* drives directly from visit (*i,m*) to visit (*j,n*), 0 otherwise
$f_{imajnbvd}$	1 if operator *d* is transported from visit (*i,a*) to (*j,b*) by service vehicle *v* in visit (*i,m*) to (*j,n*), 0 otherwise
q_{ivd}	1 if operator *d* is dropped off in *i* by service vehicle *v*, 0 otherwise
g_{jnbvd}	1 if operator *d* is picked up in visit (*j,b*) by service vehicle *v* in visit (*j,n*), 0 otherwise
h_{ijbd}	1 if operator *d* handles rental car *i* to charging station visit (*j, b*), 0 otherwise
t_{imv}^V	Time of arrival to visit (*i,m*) for service vehicle *v*
t_{iad}^D	Time of arrival to visit (*i,a*) for operator *d*
z_i^H	1 if the handling of rental car *i* is postponed, 0 otherwise
y_j	Deviation from ideal state in node *j*
s_v	1 if service vehicle *v* is used, 0 otherwise
w_d	1 if operator *d* is used, 0 otherwise

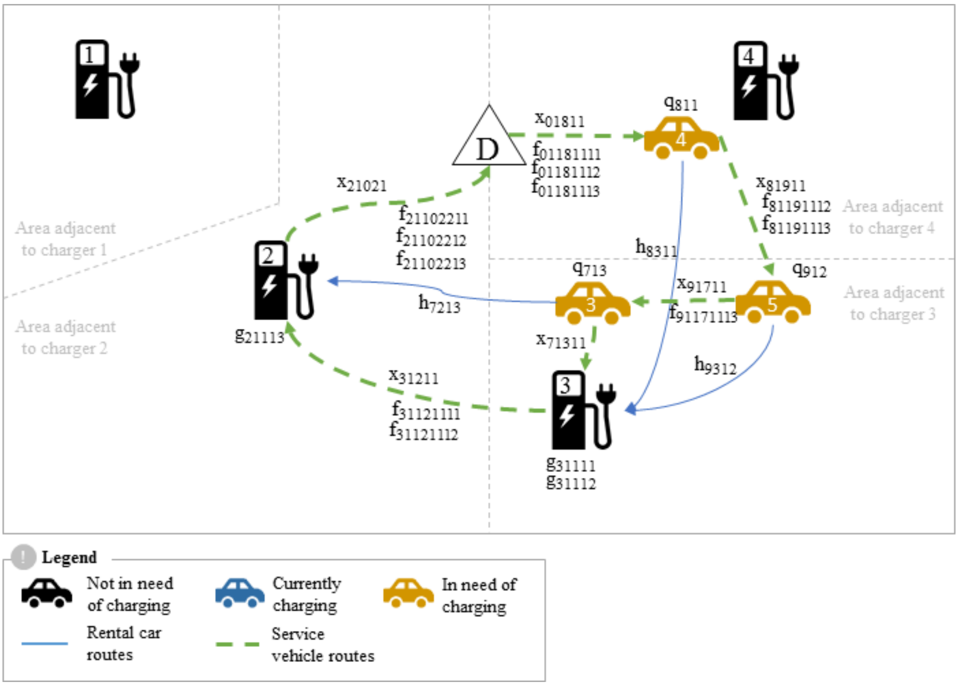

Figure 5.2: Illustration of the flow variables for one service vehicle and three operators.

a service vehicle or operator is used, respectively. The entire objective function is minimized.

$$min \sum_{j \in \mathcal{N}^{CS}} C_j^E \, y_j + \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} C_{ij}^T \, x_{imjnv} + \sum_{i \in \mathcal{N}^{EV}} C_i^{PH} \, z_i^H + \sum_{v \in \mathcal{V}} C^V \, s_v + \sum_{d \in \mathcal{D}} C^D \, w_d \tag{5.1}$$

5.3.2 Routing of Service Vehicles

Constraints (5.2) and (5.3) enforce that if a service vehicle is used, it must leave and return to the depot, respectively. Constraints (5.4) enforce that only service vehicles in use visit nodes and that only one arc is leaving a given visit (i, m). Finally, constraints (5.5) ensure that a vehicle arriving a visit (j, n) leaves the node from the same visit. This must hold for all nodes except the depot.

$$\sum_{j \in \mathcal{N} \setminus \{0\}} x_{01j1v} = s_v \qquad v \in \mathcal{V} \tag{5.2}$$

$$\sum_{j \in \mathcal{N} \setminus \{0\}} \sum_{m \in \mathcal{M}_j} x_{jm02v} = s_v \qquad v \in \mathcal{V} \tag{5.3}$$

$$\sum_{j\in\mathcal{N}\backslash\{0\}}\sum_{n\in\mathcal{M}_j} x_{imjnv} \leq s_v \qquad i \in N\backslash\{0\}, m \in \mathcal{M}_i, v \in \mathcal{V} \quad (5.4)$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i} x_{imjnv} = \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i} x_{jnimv} \qquad j \in \mathcal{N}\backslash\{0\}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.5)$$

5.3.3 Handling of Rental Cars

Constraints (5.6) ensure that the number of rental cars handled to a station does not exceed the number of available charging slots at the station. Constraints (5.7) force either the handling variable or the postponed handling variable to 1 for all rental cars. Furthermore, constraints (5.8) ensure that rental cars are handled to a charging station within reach given the car's battery capacity.

$$\sum_{i\in\mathcal{N}^{EV}}\sum_{b\in\mathcal{M}_j}\sum_{d\in\mathcal{D}} h_{ijbd} \leq N_j^{CSP} \qquad j \in \mathcal{N}^{CS} \quad (5.6)$$

$$\sum_{j\in\mathcal{N}^{CS}}\sum_{b\in\mathcal{M}_j}\sum_{d\in\mathcal{D}} h_{ijbd} + z_i^H = 1 \qquad i \in \mathcal{N}^{EV} \quad (5.7)$$

$$\sum_{b\in\mathcal{M}_j}\sum_{d\in\mathcal{D}} T_{ij}\, h_{ijbd} \leq T_i^{EV} \qquad i \in \mathcal{N}^{EV}, j \in \mathcal{N}^{CS} \quad (5.8)$$

5.3.4 Routing of Operators

Constraints (5.9) and (5.10) ensure that an operator handling a rental car is dropped off by the rental car and picked up at the charging station the rental car is handled to, respectively. Furthermore, constraints (5.11) make sure that an operator only makes a given visit b to a charging station once, either by handling to the charging station or by being transported through the charging station.

$$\sum_{j\in\mathcal{N}^{CS}}\sum_{b\in\mathcal{M}_j} h_{ijbd} = \sum_{v\in V} q_{ivd} \qquad i \in \mathcal{N}^{EV}, d \in \mathcal{D} \quad (5.9)$$

$$\sum_{i\in\mathcal{N}^{EV}} h_{ijbd} = \sum_{v\in\mathcal{V}}\sum_{n\in\mathcal{M}_j} g_{jnbvd} \qquad j \in \mathcal{N}^{CS}, \; b \in \mathcal{M}_j, d \in \mathcal{D} \quad (5.10)$$

$$\sum_{i\in\mathcal{N}^{EV}} h_{ijbd} + \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{a\in\mathcal{M}_i}\sum_{n\in\mathcal{M}_j}\sum_{v\in\mathcal{V}} f_{imajnbvd} \leq w_d \qquad j \in \mathcal{N}^{CS}, \; b \in \mathcal{M}_i, d \in \mathcal{D} \quad (5.11)$$

A service vehicle can only transport operators in use. Also, an operator cannot be picked up by more than one service vehicle in the depot. This is enforced by constraints (5.12). Constraints (5.13) ensure that operators are returned to the depot. If an operator is transported out of a node, it must be transported to that node or picked up in that node.

Similarly, if an operator is transported to a node, but not out of a node, it must be dropped off in the node. Constraints (5.14) ensure this and hence maintain the flow of operators in all nodes. Finally, constraints (5.15) make sure that a service vehicle does not exceed its seat capacity transporting operators and force the flow on arcs not driven by a service vehicle to 0.

$$\sum_{j \in \mathcal{N} \setminus \{0\}} \sum_{v \in V} f_{011j11vd} = w_d \qquad d \in \mathcal{D} \tag{5.12}$$

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{a \in \mathcal{M}_i} \sum_{v \in \mathcal{V}} f_{ima022vd} = w_d \qquad d \in \mathcal{D} \tag{5.13}$$

$$\sum_{k \in N} \sum_{o \in \mathcal{M}_k} \sum_{c \in \mathcal{M}_k} f_{jnbkocvd} = \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{a \in \mathcal{M}_i} f_{imajnbvd} + g_{jnbvd} - q_{jvd} \qquad j \in \mathcal{N} \setminus \{0\}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.14}$$

$$\sum_{a \in \mathcal{M}_i} \sum_{b \in \mathcal{M}_j} \sum_{d \in \mathcal{D}} f_{imajnbvd} \leq Q \, x_{imjnv} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.15}$$

5.3.5 Time Usage

Constraints (5.16) - (5.18) determine the service vehicle arrival time in all nodes. First, constraints (5.16) ensure that service vehicles return to the depot before the planning period is over. Constraints (5.17) track the arrival time of the service vehicles for all visits. Furthermore, constraints (5.18) make sure that a service vehicle that picks up an operator does so after the operator has arrived in the node. M_1 and M_2 are big Ms, making the constraints non-restrictive when they do not apply.

$$t^V_{imv} \leq \overline{T} \, s_v \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, v \in \mathcal{V} \tag{5.16}$$

$$t^V_{imv} + (T_{ij} + M_1) \, x_{imjnv} \leq t^V_{jnv} + M_1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.17}$$

$$t^D_{jbd} + M_2 \, g_{jnbvd} \leq t^V_{jnv} + M_2 \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.18}$$

Constraints (5.19) - (5.21) determine operator arrival time in all nodes. Constraints (5.19) ensure that operators are returned to the depot within the planning time. Constraints (5.20) track the time of an operator when handling a rental car. Finally, constraints (5.21) track the time of an operator when transported by a service vehicle. M_3 and M_4 are big Ms, making the constraints non-restrictive when they not apply.

$$t^D_{iad} \leq \overline{T}\, w_d \qquad i \in \mathcal{N}, a \in \mathcal{M}_i, d \in \mathcal{D} \quad (5.19)$$

$$t^D_{iad} + (T_{ij} + M_3)\, h_{ijbd} \leq t^D_{jbd} + M_3 \qquad i \in \mathcal{N}^{EV}, a \in \mathcal{M}_i, j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j, d \in \mathcal{D} \quad (5.20)$$

$$t^V_{imv} + (T_{ij} + M_4)\, f_{imajnbvd} \leq t^D_{jbd} + M_4 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.21)$$

5.3.6 Final State Constraints

Constraints (5.22) and (5.23) assign the absolute value of deviations from the ideal state in each charging station node to the variable accounting for deviations.

$$y_j \geq S^0_j + \sum_{i \in \mathcal{N}^{EV}} \sum_{b \in \mathcal{M}_j} \sum_{d \in \mathcal{D}} h_{ijbd} - S^I_j \qquad j \in \mathcal{N}^{CS} \quad (5.22)$$

$$y_j \geq -S^0_j - \sum_{i \in \mathcal{N}^{EV}} \sum_{b \in \mathcal{M}_j} \sum_{d \in \mathcal{D}} h_{ijbd} + S^I_j \qquad j \in \mathcal{N}^{CS} \quad (5.23)$$

5.3.7 Non-negativity, Integer, and Binary Restrictions

Constraints (5.24) - (5.34) define non-negativity, integer, and binary restrictions on all variables of the problem.

$$x_{imjnv} \in \{0,1\} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{D} \quad (5.24)$$

$$f_{imajnbvd} \in \{0,1\} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.25)$$

$$q_{ivd} \in \{0,1\} \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.26)$$

$$g_{jnbvd} \in \{0,1\} \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.27)$$

$$h_{ijbd} \in \{0,1\} \qquad i \in \mathcal{N}^{EV}, j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j, d \in \mathcal{D} \quad (5.28)$$

$$t^V_{imv} \geq 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, v \in \mathcal{V} \quad (5.29)$$

$$t^D_{iad} \geq 0 \qquad i \in \mathcal{N}, a \in \mathcal{M}_i, d \in \mathcal{D} \quad (5.30)$$

$$z^H_i \in \{0,1\} \qquad i \in \mathcal{N}^{EV} \quad (5.31)$$

$$y_j \in \mathbb{Z}^+ \qquad j \in \mathcal{N}^{CS} \quad (5.32)$$

$$s_v \in \{0,1\} \qquad v \in \mathcal{V} \quad (5.33)$$

$$w_d \in \{0,1\} \qquad d \in \mathcal{D} \quad (5.34)$$

5.4 Performance Enhancing Constraints

In this section, performance enhancing constraints aiming to improve the computation time while keeping the same optimal solution are proposed. Sections 5.4.1 and 5.4.2 present a set of symmetry breaking constraints and valid inequalities, respectively. The effect of the performance enhancing constraints will be tested in Chapter 7.

5.4.1 Symmetry Breaking Constraints

A large number of symmetric solutions of the SFFEVCHP exist. This is a result of assuming a homogeneous fleet of service vehicles and operators, and because multiple visits to charging stations are allowed. Symmetric solutions are solutions that are equal in practice but mathematically different. Constraints (5.35) - (5.38) aim to reduce the number of symmetric solutions for service vehicles. Constraints (5.35) state that the service vehicle with lowest index must be utilized first. Constraints (5.36) enforce that the service vehicle with lowest index has the longest travel time. Finally, constraints (5.37) and (5.38) enforce chronological order of service vehicle visits.

$$s_{(v+1)} - s_v \leq 0 \qquad v \in \mathcal{V} \backslash \{|\mathcal{V}|\} \tag{5.35}$$

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N} \backslash \{i\}} \sum_{n \in \mathcal{M}_j} T_{ij} \left(x_{imjn(v+1)} - x_{imjnv} \right) \leq 0 \qquad v \in \mathcal{V} \backslash \{|\mathcal{V}|\} \tag{5.36}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \left(x_{i(m+1)jnv} - x_{imjnv} \right) \leq 0 \qquad i \in \mathcal{N}, \; m \in \mathcal{M}_i \backslash \{|\mathcal{M}_i|\}, v \in \mathcal{V} \tag{5.37}$$

$$t^V_{imv} + M_5 \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{i(m+1)jnv} \leq t^V_{i(m+1)v} + M_5 \qquad i \in \mathcal{N}, \; m \in \mathcal{M}_i \backslash \{|\mathcal{M}_i|\}, v \in \mathcal{V} \tag{5.38}$$

Constraints (5.39) - (5.41) aim to reduce the number of symmetric solutions for operators. Constraints (5.39) state that the operator with lowest index must be utilized first. Constraints (5.40) ensure that the operator with lowest index handles rental cars over longer distances. Finally, constraints (5.41) - (5.44) enforce chronological order of operator visits.

$$w_{(d+1)} - w_d \leq 0 \qquad d \in \mathcal{D} \backslash \{|\mathcal{D}|\} \tag{5.39}$$

$$\sum_{i \in \mathcal{N}^{EV}} \sum_{j \in \mathcal{N}^{CS}} \sum_{b \in \mathcal{M}_j} T_{ij} \left(h_{ijb(d+1)} - h_{ijbd} \right) \leq 0 \qquad d \in \mathcal{D} \backslash \{|\mathcal{D}|\} \tag{5.40}$$

$$\sum_{i \in \mathcal{N}^{EV}} (h_{ij(b+1)d} - h_{ijbd}) - \sum_{i \in \mathcal{N}^{EV}} \sum_{m \in \mathcal{M}_i} \sum_{a \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} \sum_{v \in V} f_{imajnbvd} \leq 0 \qquad j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j \backslash \{|\mathcal{M}_j|\}, d \in \mathcal{D} \tag{5.41}$$

$$t^D_{jbd} + M_6 \Big(\sum_{i \in \mathcal{N}^{EV}} h_{ij(b+1)d} + \sum_{i \in N} \sum_{m \in M_i} \sum_{a \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} f_{imajn(b+1)vd} \Big) \leq t^D_{j(b+1)d} + M_6 \qquad j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j \backslash \{|\mathcal{M}_j|\}, d \in \mathcal{D} \tag{5.42}$$

$$t^D_{jbd} + M_7 \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{a \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} f_{imajn(b+1)vd} \leq t^D_{j(b+1)d} + M_7 \qquad j \in \mathcal{N}, b \in \mathcal{M}_j \backslash \{|\mathcal{M}_j|\}, d \in \mathcal{D} \tag{5.43}$$

$$\sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{b \in \mathcal{M}_j} \sum_{v \in V} (f_{im(a+1)jnbvd} - f_{imajnbvd}) \leq 0 \qquad i \in \mathcal{N}^{CS}, a \in \mathcal{M}_i \backslash \{|\mathcal{M}_i|\}, d \in \mathcal{D} \tag{5.44}$$

5.4.2 Valid Inequalities

Introducing valid inequalities makes the LP relaxation of the problem tighter without removing feasible solutions. Constraints (5.45) state that rental cars only can be handled by operators in use. Furthermore, constraints (5.46) state that if a service vehicle traverse an arc, the service vehicle must be used. Constraints (5.47) - (5.49) declare that if a service vehicle transports, picks up, or drops off an operator, the service vehicle must be used, respectively. Similarly, constraints (5.50) - (5.52) ensure that if an operator is transported, picked up, and dropped off, the operator must be used.

$$h_{ijbd} \leq w_d \qquad i \in \mathcal{N}^{EV}, j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j, d \in \mathcal{D} \tag{5.45}$$

$$x_{imjnv} \leq s_v \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.46}$$

$$f_{imajnbvd} \leq s_v \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.47}$$

$$g_{jnbvd} \leq s_v \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.48}$$

$$q_{ivd} \leq s_v \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.49}$$

$$f_{imajnbvd} \leq w_d \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.50}$$

$$g_{jnbvd} \leq w_d \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.51)$$

$$q_{ivd} \leq w_d \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \quad (5.52)$$

Chapter 6

Hybrid Genetic Search with Adaptive Diversity Control for the SFFEVCHP

This chapter presents a heuristic proposed for the SFFEVCHP based on the Unified Hybrid Genetic Search (UHGS) framework developed by Vidal et al. (2014). UHGS has been ranked among the most promising metaheuristics in the survey by Koç et al. (2016). This survey compares metaheuristic algorithms proposed for vehicle routing problems between 1984 and 2015. UHGS is a general algorithm that can solve multiple variants of the VRP. Overall, the SFFEVCHP can be regarded as a VRP but the problem also includes synchronization of the routing of rental cars and the routing of service vehicles as well as the decision to handle or postpone a rental car.

The implementation of the heuristic draws on the Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) first presented by Vidal et al. (2012), which is an implementation of the UHGS framework. The original HGSADC has been modified and extended significantly to fit the SFFEVCHP. The HGSADC is a non-deterministic heuristic, meaning that it neither guarantees an optimal solution nor necessarily gives the same solution when run multiple times. The following chapter describes the specifics of the HGSADC proposed for the SFFEVCHP. First, an overview of the algorithm is presented in Section 6.1. Second, the representation of individuals, the motivation for solving parts of the problem as a dial-a-ride problem, and the evaluation of individuals are described in Sections 6.2, 6.3, and 6.4, respectively. Third, each module of the overall algorithm is described in detail in Sections 6.5 to 6.8. Finally, the novelty of the proposed algorithm is discussed in Section 6.9.

6.1 Overview of the Algorithm

Algorithm 1 shows an overview of the HGSADC proposed to solve the SFFEVCHP. The algorithm evolves a population of individuals, where an individual represents a solution to the SFFEVCHP. The population is divided into two disjoint subpopulations; a feasible subpopulation and an infeasible subpopulation. The feasible subpopulation consists of all individuals in the population representing feasible solutions. The metaheuristic literature indicates that allowing a controlled exploration of infeasible solutions may enhance the performance of the search (Vidal et al., 2012). Hence, the infeasible subpopulation contains all individuals representing solutions infeasible with respect to planning time or number of service vehicles used.

The algorithm breeds new individuals from the population as long as there have been improvements within the last I^{NI} iterations or the maximum running time limit T^{MAXRUN} is not reached. In each iteration, the algorithm picks two parent individuals from the population and combines them, yielding a new individual denoted an *offspring*. The offspring can be enhanced using an *education* procedure and, if infeasible, further enhanced using a *repair* procedure. The maximum population size is given by $\mu + \lambda$, where μ is the minimum population size and λ is the maximum number of offsprings that can be created before individuals are removed, i.e. the generation size. When the maximum population size is reached, the individuals with highest biased fitness, i.e. high cost and low diversity contribution, are removed until there are only μ individuals left in the population. This process is referred to as *survivor selection*. To prevent the algorithm from converging to a local optimum, a diversification procedure is performed if there has been no improvement for I^{DIV} iterations. The initial population is created using a construction heuristic and must be large enough to contribute sufficiently to the diversity of the population.

Algorithm 1 Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)

Initialize population	Section 6.5
while *Iterations without improvement* < I^{NI} and *time* < T^{MAXRUN} **do**	
Select parent individuals s_1 and s_2	Section 6.6
Generate offspring s_{new} from s_1 and s_2	
Educate offspring s_{new} with probability $\rho^{EDU}_{offspring}$	Section 6.7
if s_{new} is infeasible **then**	
Repair s_{new} with probability $\rho^{REP}_{offspring}$	Section 6.7
end if	
if s_{new} is still infeasible **then**	
Insert s_{new} into infeasible subpopulation	
else	
Insert s_{new} into feasible subpopulation	
end if	
if maximum subpopulation size $\mu + \lambda$ reached **then**	
Select survivors	Section 6.8.1
end if	
Adjust penalty parameters for violating feasibility condition	Section 6.8.2
if best individual not improved for I^{DIV} iterations **then**	
Diversify population	Section 6.8.3
end if	
end while	
Return best feasible individual	

6.2 Individual Representation

An individual describes the routes of all operators and service vehicles. The operator routes include assignment of operators to handle each rental car, postponement of handling or assignment of rental cars to charging stations, and the handling order of each operator. The routes of the service vehicles include assignment of transport requests by operators and the visit sequence of each service vehicle. An example individual of a small, simplified problem instance is shown in Figure 6.1 below to help facilitate the discussion.

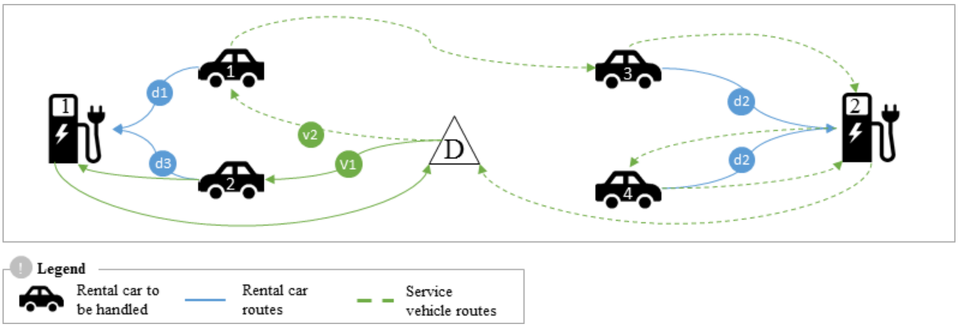

Figure 6.1: Example of an individual for a small example problem instance: Four cars in need of handling, two charging stations, two service vehicles, and three operators.

Each individual s in the population $\mathcal{S}$ is represented by five *chromosomes*. The first chromosome is the *rental car destination chromosome* $\alpha(s)$, determining the charging station to move a rental car. Alternatively, determining that the handling of the car is postponed. Each individual s consists of a *destination* $\alpha_i(s)$ for each rental car i.

The second chromosome is the *operator chromosome* $\beta(s)$, that for each rental car defines which operator that is going to perform the handling. Each individual consists of an *operator* $\beta_i(s)$ to handle each rental car i. The rental car destination and operator chromosomes for the individual in Figure 6.1 are shown in Table 6.1.

Table 6.1: Rental car destination and operator chromosomes of the example individual given in Figure 6.1.

Rental car i	1	2	3	4
$\alpha_i(s)$	1	1	2	2
$\beta_i(s)$	1	3	2	2

The third chromosome is the *handling sequence chromosome* $\gamma(s)$, that for each operator d defines the order to handle the rental cars assigned to the operator. Each individual consists of a *handling sequence* $\gamma_d(s)$ for each operator d. The handling sequence for the individual in Figure 6.1 are shown in Table 6.2.

Table 6.2: Handling sequence chromosome of example individual given in Figure 6.1.

Operator d	1	2	3
$\gamma_d(s)$	{1}	{3,4}	{2}

Taking the first three chromosomes as given, *transport requests* for the operators that need to be taken care of by the service vehicles are formulated. A transport request is formulated for each pick up of an operator. The transport request is represented by a node pair, the first node is the origin where the operator is picked up and the second node the destination where the operator is dropped off. Each transport request is denoted $\tau_r(s)$, indexed by *r* and the set of all transport requests is denoted $\mathcal{R}$. The transport requests for the individual presented in Figure 6.1 are shown in Table 6.3.

Table 6.3: Transport requests of the example individual given in Figure 6.1. The depot is denoted *D*, rental cars 1, 2, 3, and 4 *EV1*, *EV2*, *EV3*, and *EV4*, respectively, and charging stations 1 and 2 *CS1* and *CS2*, respectively.

Transport request *r*	1	2	3	4	5	6	7
Request $\tau_r(s)$	{*D,EV1*}	{*CS1,D*}	{*D,EV3*}	{*CS2,EV4*}	{*CS2,D*}	{*D,EV2*}	{*CS1,D*}

The transport request formulation is used to define the fourth chromosome, the *transport request assignment chromosome* $\delta(s)$, that assigns each transport request *r* to a service vehicle *v*. Each individual consists of a *transport request assignment* $\delta_r(s)$ for each transport request *r*. The transport request assignment chromosome for the example is presented in Table 6.4.

Table 6.4: Transport request assignment chromosome of example individual given in Figure 6.1.

Transport request *r*	1	2	3	4	5	6	7
$\delta_r(s)$	2	1	2	2	2	1	1

Finally, the last chromosome is the *route chromosome* $\varepsilon(s)$, that describes the route of each service vehicle. The route chromosome determines the order a service vehicle visits the nodes defined by the transport request assignment chromosome. Recall that an operator can be transported to an EV by one service vehicle, and picked up at the charging station by another. Each individual consists of a *route pattern* $\varepsilon_v(s)$ for each vehicle *v*. The route patterns for the given example are presented in Table 6.5.

Table 6.5: Route chromosome of example individual given in Figure 6.1. The depot is denoted *D*, rental cars 1, 2, 3, and 4 *EV1*, *EV2*, *EV3*, and *EV4*, respectively, and charging stations 1 and 2 *CS1* and *CS2*, respectively.

Service vehicle *v*	1	2
$\varepsilon_v(s)$	{*D,EV2,CS1,D*}	{*D,EV1,EV3,CS2,EV4,CS2,D*}

6.3 Solving the SFFEVCHP as a Dial-a-Ride Problem

If the chromosomes presented in Section 6.2 are determined in the same order as they are discussed, the problem of determining the transport request assignment and route chromosomes while considering the rental car destination, operator, and handling sequence chromosomes fixed is similar to a *dial-a-ride problem* (DARP). As described in Section 6.2, the first three chromosomes can be used to formulate transport requests. Each transport request is associated with the operator requesting transport by the variable $\eta_r(s)$, which is equal to *d* if operator *d* requires transport request *r*. In the DARP, customers can preorder trips and specify a time window to be picked up or dropped off. Cars are then routed to pick up customers at their desired location and transport them to their destination and multiple customers can be served simultaneously as long as there are available seats in the car. By specifying time windows for the formulated transport requests, solution methods used for the DARP can be used to create the transportation request assignment chromosome and the route chromosome in the HGSADC for the SFFEVCHP.

Using the travel times between nodes, time windows can be determined. The upper and lower limit for the time window of the origin node of transport request r is denoted $l_r^o(s)$ and $u_r^o(s)$ for individual s, respectively. Similarly, the upper and lower limit for the destination node is given by $l_r^d(s)$ and $u_r^d(s)$, respectively. The time windows are set by considering the minimum possible time required by the operator to either get to the origin node (lower limit) or finalize all handling after the destination node (upper limit) as described in Algorithm 2. The lower and upper limits are calculated by considering the arrival time of the operator in each node if a service vehicle drove together with the operator all the time, resulting in the operator being transported directly between all nodes he has to visit.

Algorithm 2 Determining time windows

```
for each operator d ∈ D do
    for each transport request r by operator d, r ∈ {r | η_r(s) = d, r ∈ ℛ(s)} do
        l_r^o(s) ← l_(r−1)^d(s) + T_{τ_(r−1)^d τ_r^o}, lower limit in the destination node of the last transport request
            plus the minimum travel time from the last transport request to the current
        l_r^d(s) ← l_r^o(s) + T_{τ_r^o τ_r^d}, lower limit in the origin node of the current transport request plus
            the minimum travel time to the destination node of the current transport request
    end do
    for each transport request r by operator d, r ∈ {r | η_r(s) = d, r ∈ ℛ(s)} (reverse direction) do
        u_r^d(s) ← u_(r+1)^o(s) − T_{τ_r^d τ_(r+1)^o}, upper limit of the origin node of the next transport request minus
            the minimum travel time from the current transport request to the next
        u_r^o(s) ← u_r^d(s) − T_{τ_r^o τ_r^d}, upper limit of the destination node in the current transport request minus
            the minimum travel time from the origin to destination node in the current transport request
    end do
end do
```

Finding the transport request assignment and route chromosomes by solving the subproblem as a DARP is done whenever new individuals are created in the HGSADC. However, the DARP itself is a hard problem to solve, as it is NP-hard (Healy and Moll, 1995). Hence, approximate solution methods are needed to solve large instances. A fast construction heuristic for the DARP is needed for the HGSADC. Low computation time is prioritized potentially at the expense of solution quality because the algorithm is executed many times. The static DARP as discussed here, as well as variations of the problem, are well studied in the literature. An extensive literature survey of model formulations and heuristic solution methods for DARP is presented by Cordeau and Laporte (2007). Although this survey is somewhat dated, it includes the majority of significant contributions to solution methods for the static DARP relevant for this thesis. More recent papers that present promising heuristics for the static DARP have been presented by Parragh and Schmid (2013), Kirchler and Wolfler Calvo (2013), Braekers et al. (2014), Osaba et al. (2015), Gschwind and Drexl (2016), and Masmoudi et al. (2017).

Cordeau and Laporte (2007) survey 16 papers on heuristics for the static DARP. As the SFFEVCHP commonly has 100 to 150 rental cars in need of handling at a given point in time, algorithms that are able to construct an initial solution for similar problem sizes in negligible time are desired. None of the full algorithms surveyed by Cordeau and Laporte (2007) achieve this, hence, only the construction phase of these algorithms are considered. Only five out of the 16 papers propose fast algorithms to construct an initial solution that are more sophisticated than random assignments. Toth and Vigo (1997), Aldaihani and Dessouky (2003), and Xiang et al. (2006) propose construction algorithms that sort transport requests by their time windows. The routes are created by assigning all transport requests possible to do without breaking any time windows to a route. For the remaining transport requests, it continues the same process with a new service vehicle, until all request have been assigned to a vehicle. This approach can be denoted *cluster first sweep second* (CFSS). Diana and Dessouky (2004) and Wong and Bell (2006) build on the same principles but also take the spatial distribution of the transport requests into account to decide the insertion order in an insertion heuristic. In the more recent papers not covered by Cordeau and Laporte (2007), Parragh and Schmid (2013) develop a hybrid column generation and large neighborhood search with an initial pool of columns generated by an insertion heuristic. Kirchler and Wolfler Calvo (2013) and Braekers et al. (2014) also employ a similar insertion heuristic to construct an initial solution. Osaba et al. (2015) and Gschwind and Drexl (2016) do not discuss the construction phase in detail. Masmoudi et al. (2017) introduce a hybrid genetic algorithm for the heterogeneous DARP, constructing the initial population using simple construction heuristics similar to Xiang et al. (2006) and randomization.

Because of a simple formulation and lower computational effort, the CFSS algorithm proposed by Xiang et al. (2006) is employed as a construction algorithm for the DARP subproblem of the individuals considered in the HGSADC. The construction heuristic is adapted to the SFFEVCHP and described in detail in Section 6.5. The constructed solution is improved by local search strategies discussed in Section 6.7.

6.4 Evaluation of Individuals

In evolutionary algorithms, evaluation of individuals is often based on the solution cost. This method promote the best individual (elitism), but does not take other factors such as the diversity of the population into account. A diverse population is important for GAs in order to avoid premature convergence to local optima and loss of information. The evaluation of individuals in the HGSADC is based on the *biased fitness* function presented by Vidal et al. (2012). The biased fitness function evaluates individuals based on their cost, how much they contribute to the diversity of the population, and how much they violate the constraints.

To evaluate the cost of an individual, let $\mathcal{A}(s)$ be the set of route patterns in individual $s \in \mathcal{S}$. Let c_{sa} be the cost of driving route $a \in \mathcal{A}(s)$, and C_s^E, C_s^{PH}, C_s^V, and C_s^D the cost of deviations from the ideal state, postponed handling, and use of service vehicles and operators in s, respectively. The individuals are allowed to violate the constraints on time used to perform handling and the number of service vehicles used. The *penalty costs* ϕ_{sa}^T and ϕ_s^V account for how much the time constraints are violated in route a and violations in number of service vehicles used in individual s, respectively. These are given by equations (6.1) and (6.2), where w^T is the *penalty parameter* per unit violation of the constraints on duration and t_{sa} is the duration of route r in individual s. w^V is the penalty parameter per unit violation of number of vehicles used by individual s, calculated by using the difference between V_s^{USED}, and available service vehicles $|\mathcal{V}|$. The *total cost* C_s of an individual s is calculated by equation (6.3).

$$\phi_{sa}^T = w^T max\{0, t_{sa} - \overline{T}\} \qquad s \in \mathcal{S}, a \in \mathcal{A}(s) \quad (6.1)$$

$$\phi_s^V = w^V max\{0, V_s^{USED} - |\mathcal{V}|\} \qquad s \in \mathcal{S} \quad (6.2)$$

$$C_s = \sum_{a \in \mathcal{A}(s)} (c_{sa} + \phi_{sa}^T) + \phi_s^V + C_s^E + C_s^{PH} + C_s^V + C_s^D \qquad s \in \mathcal{S} \quad (6.3)$$

The *diversity contribution* of each individual s is defined as the average distance to its closest neighbors. Let $\mathcal{N}_s^{CLO}$ be the set containing the n^{CLO} closest neighbors of s. The diversity contribution, $\Pi(s)$, can then be calculated as:

$$\Pi(s) = \frac{1}{n^{CLO}} \sum_{s' \in \mathcal{N}_s^{CLO}} \pi(s, s') \qquad s \in \mathcal{S} \quad (6.4)$$

where $\pi(s, s')$ is the normalized Hamming distance between individual s and s', based on the Hamming distance first presented in Hamming (1950). Here, we let the normalized Hamming distance be the number of different charging station assignments and the different handling assignments, i.e. the difference between destination assignment $\alpha_i(s)$ and $\alpha_i(s')$ and the handling assignment $\beta_i(s)$ and $\beta_i(s')$. The normalized Hamming distance is expressed as

$$\pi(s, s') = \frac{1}{2|\mathcal{N}^{EV}|} \sum_{i \in \mathcal{N}^{EV}} \Big(\mathbf{1}(\alpha_i(s) \neq \alpha_i(s')) + \mathbf{1}(\beta_i(s) \neq \beta_i(s'))\Big) \qquad s \in \mathcal{S} \quad (6.5)$$

where **1**(*cond*) = 1 if condition *cond* is true and 0 otherwise. Every individual is ranked based on its total cost and its diversity contribution. Let $Rank^C(s)$ and $Rank^D(s)$ be the rank of individual s in terms of total cost and diversity contribution, respectively. The individual with the lowest total cost will have $Rank^C(s) = 1$, and the individual with the highest total cost will have $Rank^C(s) = |\mathcal{S}|$. Equally, the individual s with highest diversity contribution will have $Rank^D(s) = 1$. Finally, the biased fitness, given by equation

(6.6), is calculated using the ranks. n^{ELI} is the number of elite individuals to survive to the next generation. If n^{ELI} equals 0, the cost and diversity ranks are given equal weight and if n^{ELI} equals $|\mathcal{S}|$, the rank is set based on the cost rank only. Hence, the composition of the total population $\mathcal{S}$ is influenced by how diversity is valued relative to the total cost because survivor selection is done based on the biased fitness.

$$BF(s) = Rank^C(s) + \Big(1 - \frac{n^{ELI}}{|\mathcal{S}|}\Big) Rank^D(s) \qquad s \in \mathcal{S} \tag{6.6}$$

6.5 Constructing the Initial Population

The initial population is constructed by creating μK^{INIT} individuals and assigning each individual to the appropriate subpopulation, where μ is the minimum number of individuals in each subpopulation. K^{INIT} should be set so that the number of individuals created ensure sufficient diversity in the population. However, setting K^{INIT} too large will increase the computational burden with marginal solution improvement as the survivor selection step of the algorithm restricts the size of each subpopulation to $\mu + \lambda$.

The initial population is created by the construction heuristic described in Algorithm 3. An individual *s* is created in four steps. Steps 1 to 3 create chromosomes $\alpha(s), \beta(s)$, and $\gamma(s)$, respectively, and Step 4 creates the remaining chromosomes $\delta(s)$ and $\varepsilon(s)$ by solving a DARP. In the first step, each rental car *i* is assigned a destination $\alpha_i(s)$. The destination is chosen semi-randomly with the closest charging station to each car having the largest probability of being chosen, the second closest second highest probability, and so on. Charging stations are chosen among the n^{CS} closest and the probabilities of all these plus the probability of postponing handling will sum to one ensuring that the rental car is assigned to a station or postponed. n^{CS} is chosen to be five in this algorithm. The list of possible destinations $G_i(s)$ is updated to only include charging stations with available charging slots. Furthermore, only charging stations within the range reachable with the given battery level of each rental car are included.

To guide how the remaining chromosomes are set, a pseudo time for each operator is used to avoid solutions with large infeasibilities in the total time constraints. Since only a small part of the problem is determined after the first step of Algorithm 3, the destination of each rental car is used to estimate the total duration of the handling. The travel time between *i* and *j* is given by T_{ij}. However, this time only accounts for the time spent while the rental car is handled. In addition to this, the operator must be transported to the rental car and picked up at the charging station. The operator may have to wait before being picked up and visit other rental cars or charging stations while riding in the service vehicle to pick up or drop off other operators. To account for this, the handling time is multiplied by a constant K^{pseudo}, which is greater than 1. Hence, the pseudo time can be expressed as:

$$t_d^{pseudo}(s) = \sum_{i \in N^{EV} | \beta_i(s) = d} K^{pseudo} T_{i\alpha_i(s)} \qquad s \in \mathcal{S} \tag{6.7}$$

Step 2 assigns a random handling assignment pattern to each rental car. With probability ρ^{assign} the pattern is chosen with priority on using a low number of operators as presented in Algorithm 3.1. Rental cars are assigned to operators with a greedy algorithm, adding rental cars to the operator as long as the pseudo time of the operator does not exceed the planning time $\overline{T}$. When the planning time is exceeded for one operator, rental cars are added to the next operator in the same fashion. Alternatively, operators are assigned with priority on reducing the distance travelled by service vehicles, presented in Algorithm 3.2. This is done by attempting to assign rental cars to operators so that cars handled to the same charging station are handled by different operators to allow service vehicles to only visit charging stations once. To do this, a list $H_i(s)$ of rental cars with destination i is created for each charging stations. For each charging station i, the rental cars in $H_i(s)$ are assigned to different operators. Only if the number of operators is limited, multiple rental cars will be handled to the same charging station by the same operator. ρ^{assign} is set to 0.5 to make sure the two methods are used approximately as many times.

The third step of the algorithm sets the handling order for the cars assigned to each operator. Until all cars have been included in the sequence, a new car is added to the end of the sequence. The car closest to the position of the operator after the previous handling is added with a probability ρ^{seq}, otherwise a random car is added.

Using the time windows, the origin and destination nodes of the transport requests are sorted, lowest upper limit first, in a list *L(s)*. The transport requests are split into origin and destination nodes because a service vehicle assigned to that request does not necessarily drive directly from the origin to the destination, other nodes can be visited in-between. Requests that are in conflict, i.e. not possible to fulfill with the given time windows on the same route, are stored in a conflict table *C(s)*. Using *L(s)* and *C(s)*, routes are created using the sweep heuristic proposed by Xiang et al. (2006), described in Algorithm 3.3. The algorithm iterates through the list *L(s)* adding unvisited nodes that are not in conflict with any of the nodes already in the route. Furthermore, destination nodes are added to the route if the origin node already is in the route. After all elements of *L(s)* are searched, a new route is created and all unvisited nodes in *L(s)* are searched and added by the criteria described above. The resulting assignment of transport requests to service vehicles and service vehicle routes are stored in the transport request assignment and route chromosomes, respectively.

6.6 Parent Selection and Crossover

Crossover is the process where the chromosomes of two parent individuals, s_1 and s_2, are combined into a new individual, s_{new}, denoted an *offspring*. Each parent is selected by a binary tournament, i.e. randomly picking two individuals from the entire population and choosing the one with best biased fitness as the parent, as proposed by Vidal et al. (2012). The four-stepped crossover operator is described in Algorithm 4. In the first step (Step 0), the genes to inherit from each parent are decided. This is done by randomly dividing the set of rental cars in three disjoint sets: Λ_1, Λ_2, and Λ_{mix} containing rental cars inheriting patterns from s_1, s_2, and both, respectively.

The next step (Step 1) inherits data from s_1. The destination and operator for all EVs in Λ_1 are copied directly from s_1 to s_{new}. Two random cut-off points υ_1 and υ_2,

Algorithm 3 Construction heuristic

individualsCreated $s \leftarrow 0$
while $s < \mu K^{INIT}$ **do**

STEP 1: SELECT SEMI-RANDOM DESTINATION PATTERN
Create sorted list $G_i(s)$ with the closest charging stations to rental car *i*
$CS_i^{cap}(s) \leftarrow$ Number of available charging slots at charging station i
for each rental car $i \in N^{EV}$ **do**
Choose charging station, *with available charging slot*, *g* in $G_i(s)$ with probability $\rho_g = \frac{1}{|G_i(s)|}$
$\alpha_i(s) \leftarrow g$
$CS_g^{Cap}(s) \leftarrow CS_g^{Cap}(s) - 1$
end do

STEP 2: SELECT RANDOM HANDLING ASSIGNMENT PATTERN
with probability ρ^{assign} **do**
Apply algorithm 3.1 to create handling assignment pattern with low operator cost
else do
Apply algorithm 3.2 to create handling assignment pattern with low travel cost
end do

STEP 3: SELECT SEMI-RANDOM HANDLING SEQUENCE PATTERN
for each operator $d \in D$ **do**
Create set of rental cars that are handled by each operator, $\mathcal{F}_d(s) = \{i | \beta_i(s) = d\}$
while $\mathcal{F}_d(s) \neq \emptyset$ **do**
with probability ρ^{seq} **do**
add rental car $i \in \mathcal{F}_d(s)$ closest to position of operator *d* after previous handling to $\gamma_d(s)$
else do
add random rental car $i \in \mathcal{F}_d(s)$ to $\gamma_d(s)$
end do
Remove i from $\mathcal{F}_d(s)$
end do
end do

STEP 4: SOLVE THE DIAL-A-RIDE PROBLEM WITH THE THREE FIRST CHROMOSOMES AS INPUT
Formulate transport requests and determine time windows using Algorithm 2
Create list *L(s)*, the node visit sequence sorted by the end time of the time window to serve all transport requests
Create conflict table *C(s)* of the transport requests with conflicting time windows
Create initial service vehicle routes using Algorithm 3.3, use routes to set $\delta_r(s)$ and $\varepsilon_v(s)$

Educate generated individual with probability $\rho_{construct}^{EDU}$
if generated individual is infeasible **then**
Repair individual with probability $\rho_{construct}^{REP}$
end if

individualsCreated $s \leftarrow s + 1$
end while

Algorithm 3.1 Handling assignment with low operator cost

$d \leftarrow 1$
Create sorted list $H(s)$ of rental cars that are handled, shortest handling time first
while $H(s) \neq \emptyset$ **do**
    EV $\leftarrow$ first element of $H(s)$
    if *pseudo time of* $d \leq \overline{T}$ *when EV is assigned to* d **then**
        $\beta_{EV}(s) \leftarrow d$
        Update pseudo time and remove *EV* from $H(s)$
    else if $(d + 1 \leq |D|)$ **then**
        $d \leftarrow d + 1$
    else
        Set $\alpha_i(s)$ to postpone for the remaining rental cars *i* in $H(s)$
        $H(s) \leftarrow \emptyset$
    end if
end do

Algorithm 3.2 Handling assignment with low travel cost

for each charging station $i \in N^{CS}$ **do**
Create set of rental cars being handled to charging station $H_i(s)$
    $d \leftarrow 1$
    while $H_i(s) \neq \emptyset$ **do**
        EV $\leftarrow$ random rental car from H_i(s)
        $\beta_{EV}(s) \leftarrow d$
        Remove *EV* from H_i(s)
        $d \leftarrow d + 1$
        if $d > |D|$ **then**
            $d \leftarrow 1$
        end if
    end do
end do

$\upsilon_1 \leq \upsilon_2$, are picked for the set Λ_{mix}, and the destination and the operator for the EVs in the sequence between these cut-off points are copied from s_1 to s_{new}. Furthermore, the handling sequence are copied directly from s_1 to s_{new}.

In Step 2, data is inherited from s_2. For all the remaining EVs in Λ_2 and Λ_{mix}, the destination is copied to s_{new} if capacity constraints on the charging stations are not violated. If the capacity constraints are violated, the rental car will instead be assigned to the closest charging station with available charging slots. The operator is copied directly. The handling sequence are copied directly from s_2 to s_{new}, except for the EVs already in $\gamma_d(s_{new})$. This ensures that all rental cars are handled without conflict between operators. An improvement heuristic minimizing the travel distance of the operator is then applied to improve the handling sequence patterns.

Finally, in Step 3, transport requests and service vehicle routes are constructed using Step 4 from the construction heuristic (Algorithm 3.1). Due to the design of the crossover operator, offspring individual s_{new} is feasible except in the time constraints and number of service vehicles used, which are allowed to violate. The crossover procedure used for the two first chromosomes of the SFFEVCHP is strongly inspired by the *periodic crossover*

Algorithm 3.3 Route construction heuristic (Xiang et al., 2006)

for each unvisited vertex *i* in list $L(s)$ **do**
  if vertex *i* is a pick up site **then**
    Add vertex *i* as the first pick up site in a new route
    for each unvisited vertex *j* after vertex *i* in list $L(s)$ **do**
      if vertex *j* is a pick up site and does not *conflict* with any request already in this route or vertex *j* is a delivery site and its corresponding pick up site is already in this route **then**
        Add vertex *j* to the tail of this route
      end if
    end do
  end if
end do

with insertions (PIX) dedicated to periodic routing problems and designed to transmit good sequences of visits. The PIX is proposed by Vidal et al. (2012).

6.7 Education

The education phase aims to decrease the total cost of an individual. This is achieved by attempting to improve the handling sequence, transport request assignment, and route chromosomes. As different rental car destination and operator chromosomes are evaluated as a part of the overall HGSADC, these are not altered in the education module. Simple improvement operators are sought in order to be able to run a large number of improvement iterations with little computational effort. The education module also includes a procedure to make infeasible individuals feasible. In this section, solution improvement is discussed in Section 6.7.1 and the repair procedure is presented in Section 6.7.2

6.7.1 Solution Improvement

Neighbors are defined by a *neighborhood operator* based on Braekers et al. (2014). A relocate operator removes a transport request from its current position in a route and attempts to insert the transport request in either another position in the same route or in a different route. The operator is illustrated in Figure 6.2. Transport requests can be inserted in positions that requires modification of the handling sequence chromosome. This happens if the modified routes force an operator to visit the rental cars in a different order than the order defined in the handling sequence chromosome. A change in this chromosome also requires the transport requests to be modified. This is done so that the transport requests satisfy the flow of operators as described in Section 6.2. Furthermore, if an improving inter-route move is found, the transport request assignment chromosome is modified so that it captures that a new service vehicle handles the transport request.

A *first improvement* strategy is implemented, meaning that the first improvement found is accepted and the search for better solutions continues by considering the next transport request. First improvement is chosen because it has been shown that there is little difference between *best improvement* and first improvement (Breedam, 2001). When all

Algorithm 4 Crossover operator

STEP 0: INHERITANCE RULE
Pick two random numbers between 0 and $|N^{EV}|$ according to a uniform distribution. Let n_1 and n_2 be the smallest and the largest of these numbers, respectively
Randomly select n_1 rental cars to form the set Λ_1
Randomly select $n_2 - n_1$ remaining rental cars to form the set Λ_2
The remaining $|N^{EV}| - n_2$ rental cars make up the set Λ_{mix}

STEP 1: INHERIT DATA FROM s_1
for each EV i belonging to the set Λ_1 **do**
  Copy the destination $\alpha_i(s_1)$ to $\alpha_i(s_{new})$ and the operator $\beta_i(s_1)$ to $\beta_i(s_{new})$
end for
Pick two random cut-off points υ_1 and υ_2 dividing the set Λ_{mix}
for each EV i in the subset between υ_1 and υ_2 **do**
  Copy the destination $\alpha_i(s_1)$ to $\alpha_i(s_{new})$ and the operator $\beta_i(s_1)$ to $\beta_i(s_{new})$
end for
Copy handling sequence $\gamma_d(s_1)$ to $\gamma_d(s_{new})$ for all drivers and EVs so far inherited from s_1

STEP 2: INHERIT DATA FROM s_2
for each EV $i \in \Lambda_2 \cup \Lambda_{mix}$ **do**
  if $\alpha_i(s_{new}) = \emptyset$ and destination assignment not violates capacity at charging station $\alpha_i(s_2)$ **do**
    Copy the destination $\alpha_i(s_2)$ to $\alpha_i(s_{new})$
    Copy the handling assignment $\beta_i(s_2)$ to $\beta_i(s_{new})$
  else if $\alpha_i(s_{new}) = \emptyset$ **do**
    Assign EV i to the closest available charging station or postpone
    if EV i not postponed **do**
      Copy the operator $\beta_i(s_2)$ to $\beta_i(s_{new})$
    end if
  end if
end do
Copy the handling sequence from s_2 to s_{new} for all drivers and EVs inherited form s_2
Apply improvement heuristic to improve handling sequence pattern

STEP 3: ROUTE SERVICE VEHICLES
Apply step 4 from construction heuristic (Algorithm 3) to formulate transport requests and route service vehicles

transport requests in the current routes are searched through, the search terminates. The education procedure is run as long as improvements are found.

6.7.2 Repair

Individuals that are feasible after education is performed are referred to as *naturally feasible individuals*. If an individual is infeasible, the individual is *repaired* with probability ρ^{REP} attempting to make it feasible. This is done by multiplying the penalty parameters by ten and running the education procedure again. If the individual still is infeasible, the penalty parameters are multiplied by 100 and the education procedure executed. If the individual still is infeasible, a module forcing the individual to become feasible is employed.

The force feasibility module consists of two parts. The first part repairs individuals that are using to many service vehicles and the second individuals that exceeds the maximum time limit. If too many service vehicles are used, the module searches through all routes

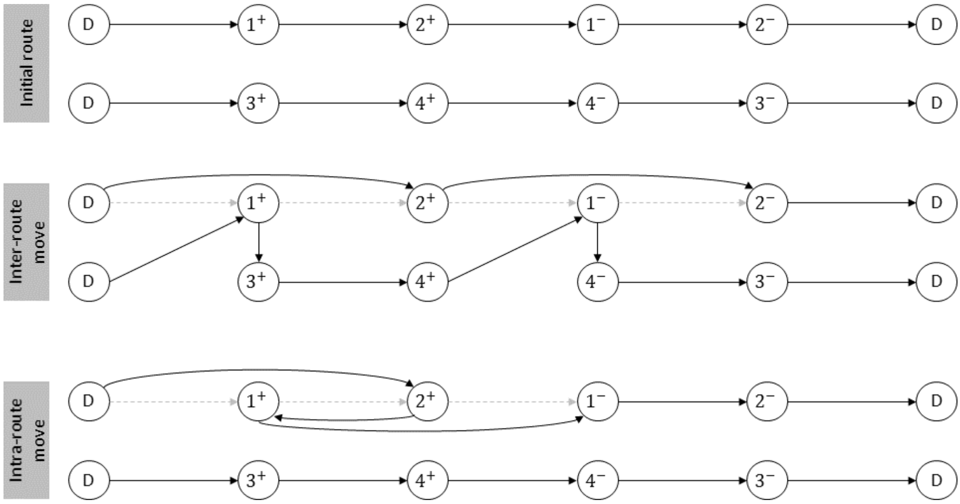

Figure 6.2: Illustration of relocate-operator, inter and intra-route moves allowed. 1^+ and 1^- denote pick up and drop off nodes of transport request 1, respectively. Equivalent notation applies for transport requests 1 to 4.

to find the vehicle that handles the fewest transport requests. Then, all the rental cars corresponding to these transport requests are postponed. The postponed rental cars are removed from the handling sequence chromosome of the relevant operators and the DARP is re-solved with the updated chromosomes to determine the transport request assignment and route chromosomes. This procedure is repeated until enough rental cars are postponed so that the service vehicle limit is no longer exceeded. If an individual is exceeding the maximum time limit constraint, all routes are searched through to find the route with the longest duration. Then, the rental car corresponding to the last transport request in the route is postponed. Similar to the first part of repair, the handling sequence chromosome is updated and the DARP re-solved. The procedure is repeated until the individual no longer exceeds the maximum handling time. Note that even though repair guarantees feasibility, the procedure is not run for all individuals. Hence, infeasible solutions are still present.

6.8 Population Management

Three population management schemes are employed to improve the performance of the genetic search algorithm. These are *survivor selection*, *penalty parameter adjustment*, and *diversification* discussed in Sections 6.8.1, 6.8.2, and 6.8.3, respectively. The purpose of the proposed schemes is to promote convergence towards high quality individuals, to maintain a certain balance between feasible and infeasible individuals, and to ensure the diversity of the population.

6.8.1 Survivor Selection

Survivor selection is performed to increase the quality of the population by removing the worst quality individuals based on the biased fitness. Survivor selection is executed on a population whenever the number of individuals in the population reaches its maximum limit $\mu + \lambda$. Individuals are removed until there are μ individuals left.

6.8.2 Penalty Parameter Adjustment

The penalty parameters ω^p, where $p \in \{T, V\}$, for violating the planning time and maximum number of service vehicles constraints are updated every 100 iterations with the goal of attaining a desirable share of feasible individuals. Looking at the current population, the proportion of individuals feasible in each of the constraints is evaluated and compared to the desirable proportion. The procedure for adjusting the penalty parameters is described in Algorithm 5. The share of individuals feasible in the time constraints and service vehicle constraints are denoted ζ^T and ζ^V, respectively. Furthermore, the desired ratio is denoted ζ^{REF}. Finally, $\xi^{UP} \geq 1$ and $\xi^{DOWN} \leq 1$ are adjustment factors for the penalties.

Algorithm 5 Penalty Parameter Adjustment

```
for p ∈ {T, V} do
    if ζ^p ≤ ζ^REF − 0.05 then
        ω^p = ω^p ξ^UP
    else if ζ^p ≥ ζ^REF + 0.05 then
        ω^p = ω^p ξ^DOWN
    end if
end for
```

6.8.3 Diversification

The diversification procedure is executed to prevent the algorithm from converging to a local optima. If no improvement is made to the best individual in I^{DIV} iterations, two thirds of the worst individuals are removed from each subpopulation. Then, μK^{DIV} new individuals are generated using the construction heuristic described in Section 6.5.

6.9 Novelty of the HGSADC

This chapter describes a novel application of the UHGS framework developed by Vidal et al. (2014). The UHGS framework is a component-based heuristic framework for multi-attribute vehicle routing problems (MVRPs), that has demonstrated solid performance. Nevertheless, with synchronization of the routing of operators and service vehicles including both spatial and temporal interdependencies, the SFFEVCHP represents a new and complex problem type. Hence, new chromosomes are proposed to represent the individuals and the GA operators have been adapted to fit this new problem type. A novel construction heuristic has been developed, combining semi-random assigning and heuristics used for DARPs. Components have been developed for education, diversification, and population management, strongly inspired by the HGSADC presented in Vidal et al. (2012).

The concepts of penalty costs for infeasibilities, ranking of individuals and inclusion of diversity contribution rank are all used by Vidal et al. (2012). These concepts are also used in the HGSADC for the SFFEVCHP, but the operators are changed to fit the representation of the individuals. Vidal et al. (2014) allow investigation of infeasible individuals during the search. In the HGSADC for the SFFEVCHP, infeasibility is only allowed in two constraints; planning time and number of service vehicles used. We believe that the complexity of the problem can make it hard to return to the feasible search space if many constraints are violated. Infeasibilities are handled in the same fashion as in Vidal et al. (2012) by penalizing violations in the biased fitness function. To demonstrate the capabilities of the proposed algorithm, extensive testing is performed in the computational study in Chapter 7.

Chapter 7

Computational Study

In this chapter, the Mixed Integer Program (MIP) presented in Chapter 5 and the HGSADC described in Chapter 6 is thoroughly tested. A short description of the test instances is provided in Section 7.1. Furthermore, the MIP and the HGSADC are calibrated in Section 7.2 and Section 7.3, respectively. Lastly, the final model and algorithm configuration are used to solve multiple instances to demonstrate and analyze the final performance in Section 7.4.2. The hardware and software used in the computational study is summarized in Table 7.1.

Table 7.1: Details of computer and solver used in the computational study.

Processor:	Intel(R) Core(TM) i7-6700 CPU 3.40GHz
RAM:	32 GB
Operating system:	Windows 10 Education 64-bit
Xpress-IVE version:	1.24.08 64 bit
Xpress optimizer version:	28.01.04
Mosel version:	3.10.0
Java version:	8
Maximum computation time:	3600 seconds

7.1 Test Instances

Test instances are created based on the data of a real CSO in Milan. An artificial depot is placed in the city center of Milan, and all rental cars in need of handling within 30 minutes drive form the depot is considered. Rental cars are drawn randomly from the set of considered cars, depending on the size of the test file. An overview of the test instances and their parameters are shown in Table 7.2. The letters a, b, and c are used to distinguish between test files of equal size. Different test instances are used for calibration and performance testing to avoid overfitting the model and algorithm to the data, i.e. fitting the model to only perform well on the given data set.

Travel times are retrieved from Google maps and assumed equal for both service vehicles and rental cars. Travel times are computed between all pairs of rental cars and

Table 7.2: Overview of the key parameters of the constructed test instances

Instance	# cars to be handled	# charging stations	# service vehicles	# operators	Planning period duration (min)
4_2	4	2	1	4	120
6_3	6	3	2	6	120
8_4	8	4	3	8	120
15_5	15	5	3	12	120
30_10	30	10	6	24	120
60_20	60	20	10	40	120
100_35	100	35	14	56	120
125_40	125	40	16	64	120
150_45	150	45	18	72	120
175_50	175	50	20	80	120
200_55	200	55	22	88	120

charging stations, as well as the depot. The ideal state of the system is set so that the number of rental cars is equal in all charging stations. The initial state in each charging station is a random number such that the total number of rental cars conform with the number of rental cars in the ideal state. The cost parameters used in the implementation of the model are estimated and meant to represent the relative size of the costs involved (e.g. how much does a minute of travel cost compare to having deviations from the ideal state in a node). To perform the calculations, Euro is used as currency. An overview of the estimations can be seen in Table 7.3.

Table 7.3: Cost parameters and estimated values.

Cost parameter	Deviation cost, C_i^E $(i \in N^{CS})$	Travel cost per minute, C_{ij}^T	Postponement cost, C^{PH}	operator cost, C^D	Vehicle cost, C^V
Estimated value	10	0.1	25 / 50	20	20

It is assumed that an average trip in a given area yields a revenue of 10 Euros for the CSO. Further, a 25 percent profit margin on the revenue is assumed, yielding an average profit of 2.50 Euros per trip. Deviations from the ideal state results in under-utilization of the rental cars. We assume that on average a rental car will lose four trips per deviation. Therefore the average deviation cost is set to be 10 Euros. The deviation cost C_i^E is assumed to be equal for all areas. The model does however allow different deviation costs in different areas. If handling of a rental car is postponed, the rental car will be unavailable for users in the period following the planning period. In addition, we assume that it results in lower customer satisfaction as fewer rental cars are available. Therefore, the cost of postponing, C_i^{PH}, is assumed to be 25 Euros for the instances with less than 15 cars in need of handling. The cost of postponed handling is set equal for all rental cars. For the larger instances, C^{PH} is set to 50 Euros. This is because a higher value is needed to promote that the HGSADC-algorithm handles the majority of the cars instead of postponing. In real life, the cost parameters discussed here are dependent on how frequent handling of the carsharing system is performed and individual characteristics of each CSO.

All costs associated with the service vehicle is included in the travel cost. Maintenance cost is included in the travel cost to capture that it will increase with the kilometers driven. However, only a small fraction of the maintenance costs will be attributed to each minute travelled. If we assume an average speed during transport of 30 kilometers per hour in urban areas, a car will travel half a kilometer per minute. Further, we assume a fuel cost of one Euro and fifty cents per ten kilometers, and add 50 cents per ten kilometer travelled for maintenance and other costs. Combining these assumptions yield a travel cost of ten cents per minute travelled. The unit cost is multiplied by the travel times to arrive at the travel cost matrix C_{ij}^T. It is assumed that service vehicles are gasoline cars, but in reality service vehicles might as well be EVs. An hourly total employee cost is assumed to be 10 Euros per hour per operator. The only fixed cost associated with service vehicles not included in the travel cost is the cost of the operator of the service vehicle. Hence, the cost of operators and service vehicles are 20, as the planning period is set to 120 minutes.

7.2 Mathematical Model Configuration

The MIP presented in Chapter 5 has been implemented in Xpress IVE version 1.24.08 using the Mosel programming language. In Section 7.2.1 and 7.2.2 the big Ms and the number of possible visits in the model are briefly explained, respectively. Further, in Section 7.2.3 different combinations of performance enhancing constraints are tested in order to arrive at a final model formulation.

7.2.1 Big Ms

To make the formulation as tight as possible, the big Ms must be set as small as possible. As all big Ms are used to enforce time constraints, the largest difference between two time variables is the planning time of the problem. This is specified in equations (7.1) - (**??**).

$$M_1 = max\{t_{jnv}^V - t_{imv}^V\} = \overline{T} \qquad i \in N, m \in M_i, j \in N, n \in M_i, v \in V \tag{7.1}$$

$$M_2 = max\{t_{jnv}^V - t_{jbd}^D\} = \overline{T} \qquad j \in N^{CS}, n \in M_j, b \in M_j, v \in V, d \in D \tag{7.2}$$

$$M_3 = max\{t_{jbd}^D - t_{iad}^D\} = \overline{T} \qquad i \in N^{EV}, a \in M_i, j \in N^{CS}, b \in M_j, d \in D \tag{7.3}$$

$$M_4 = max\{t_{jbd}^D - t_{imv}^V\} = \overline{T} \qquad i \in N, m \in M_i, a \in M_i, j \in N, v \in V, d \in D \tag{7.4}$$

$$M_5 = max\{t_{im+1v}^V - t_{imv}^V\} = \overline{T} \qquad i \in N, m \in M_i, v \in V \tag{7.5}$$

$$M_6 = max\{t_{jb+1d}^D - t_{jbd}^D\} = \overline{T} \qquad j \in N, b \in M_j, d \in D \tag{7.6}$$

$$M_7 = max\{t_{jb+1d}^D - t_{jbd}^D\} = \overline{T} \qquad j \in N, b \in M_j, d \in D \tag{7.7}$$

7.2.2 Number of Possible Visits

In the MIP the set M_i is defined for each node i, representing possible visits to a node i for both service vehicles and operators. Hence, the maximum possible visits to a node i is $|M_i|$ for both service vehicles and operators. The project report by Folkestad and Hansen (2016) found that setting the number of visits equal to the lower bound plus one ensures the best trade off between solution quality and computational time. Hence, number of possible visits is set equal to the lower bound plus one for all charging stations. Each rental car can only be visited once, so both the maximum and minimum number of visits are set to 1 for all rental cars.

7.2.3 Final Model Formulation

Results from Folkestad and Hansen (2016) show that valid inequalities and the simplest symmetry breaking constraints yield significant improvements when added to the model. Thus, constraints (5.35), (5.39), and (5.45) - (5.52) are included in the final model formulation. These constraints are referred to as base constraints. In addition to the base constraints, combinations of the remaining performance enhancing constraints are tested in order to arrive at the final model formulation. An overview of the different combinations tested are given in Table 7.4.

Table 7.4: Configurations with different combinations of performance enhancing constraints tested.

Configuration	Base Constraints	Added Constraints
0	No performance enhancing constraints	
1		-
2	(5.35), (5.39), (5.45) - (5.52)	(5.37), (5.38), (5.41)-(5.44)
3		(5.36), (5.40)
4		(5.36) - (5.38), (5.40), (5.41)-(5.44)

The computation time and the gap at the end of runs for the tested configurations are shown in Table 7.5. Looking at the table it is evident that for small instances, none of the combinations of performance enhancing constraints improve the performance of the model. Configuration 1, however, performs better than Configuration 0 on the largest instances. For both model configurations the lower bound on each of the 8_4 instances are equal, but Configuration 1 finds solutions with lower gaps. On the two smaller instance sizes the gaps are equal as the optimal solution is found in all cases, while the computation time for Configuration 1 increases significantly compared to Configuration 0. It is likely that the performance enhancing constraints performs better when test instances gets larger, while at small test instances they complicate the problem unnecessary. Hence, Configuration 1 is chosen as the final model formulation as good solutions to large test instances are preferred.

7.3 HGSADC Configuration

In this section the parameters of the HGSADC is calibrated and a final configuration of the HGSADC is determined. First, the test methodology is discussed in Section 7.3.1. Then, the target ratio of feasible individuals, population and generation size, stopping criteria, penalty parameters, pseudo time, education and repair, number of individuals created in the construction heuristic and diversity step, and proportion of elite individuals are calibrated in Sections 7.3.2 to 7.3.9, respectively. Finally, concluding remarks on the parameter calibration are presented in Section 7.3.10.

Table 7.5: Computational time and gap at the end of MIP runs for combinations of performance enhancing constraints. Average is compared to the model without performance enhancing constraints (Configuration 0).

Instance	**Configuration 1**		**Configuration 2**	
	Time (s)	Gap %	Time (s)	Gap %
4_2_a	0.52	0	0.28	0
4_2_b	1.33	0	1.67	0
4_2_c	0.66	0	0.61	0
Average	0.8	0	0.9	0
Average change	36.4%	0%	39.1%	0%
6_3_a	2087.56	0	1890.36	0
6_3_b	122.2	0	116.67	0
6_3_c	6.8	0	9.48	0
Average	738.9	0	672.2	0
Average change	109.5%	0%	90.6%	0%
8_4_a	>3600	9.4	>3600	39.3
8_4_b	>3600	7.3	>3600	14
8_4_c	>3600	67.4	>3600	177.8
Average	N/A	28.0	N/A	77.0
Average change	N/A	-16.1%	N/A	130.6%

Instance	**Configuration 3**		**Configuration 4**	
	Time (s)	Gap %	Time (s)	Gap %
4_2_a	0.33	0	0.41	0
4_2_b	2.06	0	1.81	0
4_2_c	0.53	0	0.95	0
Average	1.0	0	1.1	0
Average change	58.7%	0%	72.3%	0%
6_3_a	2436.11	0	2412.94	0
6_3_b	290.76	0	134.54	0
6_3_c	13.73	0	11.58	0
Average	913.5	0.0	853.0	0.0
Average change	159.1%	0%	141.9%	0%
8_4_a	>3600	38.6	>3600	46.3
8_4_b	>3600	15.4	>3600	24.5
8_4_c	>3600	159.2	>3600	191.2
Average	N/A	71.1	N/A	87.3
Average change	N/A	112.8%	N/A	161.5%

Best improvement marked with green.

7.3.1 HGSADC Parameter Calibration Methodology

An overview of the parameters used in the HGSADC for the SFFEVCHP are shown in Table 7.6 along with their chosen values. ζ^{REF}, μ, λ, η^{DIV}, I^{NI}, K^{PSEUDO}, ρ^{EDU}, ρ^{REP}, K^{INIT}, K^{DIV}, and η^{ELI} are the parameters that affect the search of the algorithm the most, and are therefore thoroughly tested. The penalty parameters of this implementation are different than in Vidal et al. (2012), thus these are tested as well. Each parameter is tested on five different instances. Since the HGSADC is non-deterministic, the parameters are tested five times for each instance. The average computation time and the average gap to the best known solution of the given instance are reported. The best known solution is the optimal solution found by the MIP for the 6_3 instance and the best solution found using the HGSADC for the remaining instances. High performance on the largest test instances are weighted the most, as these best resemble the real world usage of the algorithm.

Table 7.6: Overview of the parameters used in the HGSADC and their values

Parameter	Value	Description
μ	35	Minimum population size
λ	100	Generation size
I^{NI}	10,000	Max. number of iterations without improvement
η^{DIV}	0.2	Proportion of I^{NI} , such that $I^{DIV} = \eta^{DIV} \times I^{NI}$
η^{ELI}	0.5	Proportion of elite individuals, $n^{ELI} = \eta^{ELI} \times \lvert\mathcal{S}\rvert$
η^{CLO}	0.2	Proportion of individuals considered in diversity contribution, such that $n^{CLO} = \eta^{CLO} \times \mu$
K^{INIT}	20	Construction heuristic size factor
K^{DIV}	20	Diversification size factor
$\rho^{EDU}_{construct}$	0.75	Probability of education in construction heuristic
$\rho^{REP}_{construct}$	0.25	Probability of repair in construction heuristic
$\rho^{EDU}_{crossover}$	0.5	Probability of education in crossover
$\rho^{REP}_{crossover}$	0.5	Probability of repair in crossover
ζ^{REF}	0.6	Desired ratio of feasible individuals
w^{T}	2	Duration violation penalty
w^{V}	0.5	Number of vehicles violation penalty
ξ^{UP}	1.25	Penalty adjustment factor, up
ξ^{DOWN}	0.75	Penalty adjustment factor, down
T^{MAXRUN}	3,600	Maximum running time (seconds)

Based on Vidal et al. (2012) and preliminary testing, a base configuration has been defined. The base parameter values are presented in Appendix B. Different values for each parameter are tested individually, keeping the rest of the parameters fixed to the base values. The parameters are tested in the order presented in this section. Once a parameter value is chosen, the remaining tests are performed with all prior parameter values set to the chosen values. This is done to avoid ending up with a set of chosen parameters that work well when the rest of the parameters are at the base values, but perform badly when combined. The parameter tuning is described in detail in the following subsections.

Thorough parameter tuning has not been performed on all parameters. ξ^{UP} and ξ^{DOWN} are constants set by Vidal et al. (2012). The values set by Vidal et al. (2012) were adjusted somewhat due to observed improved performance during preliminary testing. η^{CLO} is not expected to affect performance of the algorithm significantly and is therefore set to the same as in Vidal et al. (2012). Finally, the maximum running time, T^{MAX}, is set to the same as for the MIP, 3600 seconds.

7.3.2 Calibrating the Target Ratio of Feasible Individuals

ζ^{REF} is the target ratio of feasible individuals in the population used in the penalty parameter adjustment, described in Section 6.8.2. A high target ratio results in higher penalties for infeasibility, hence guiding the population towards a higher proportion of feasible individuals. Table 7.7 shows the average running times and gaps for different values of ζ^{REF}. The performance of all values of ζ^{REF} are comparable, but $\zeta^{REF} = 0.6$ and $\zeta^{REF} = 0.8$ have the best average performance. Of those, $\zeta^{REF} = 0.6$ performs better on the two largest test instances and is therefore chosen as the target ratio of feasible individuals.

Table 7.7: The average running time and gap from the best known objective value for different values of ζ^{REF}. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	$\zeta^{REF} = 0.6$		$\zeta^{REF} = 0.8$		$\zeta^{REF} = 1.0$	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	4.6	0.2	4.6	0.3	4.2	0.3
15_5	26.8	0.0	25.4	0.1	29.8	0.1
30_10	84.8	9.3	86.0	8.4	52.4	11.5
60_20	158.0	5.6	149.8	5.7	128.4	5.6
100_35	134.8	6.0	147.0	6.3	122.4	5.6
Average	81.8	4.2	82.6	4.2	67.4	4.6

Best and second best average performance for each instance marked with dark and light green, respectively.

7.3.3 Calibrating the Minimum Population and the Generation Size

The minimum population size μ and the generation size λ are expected to have a high correlation since both adjust the size of the population. Hence, these parameters are calibrated together. Table 7.8 shows the average running time and gap for different combinations of μ and λ. In addition to best overall performance on the two largest test instances, $\lambda = 100$ and $\mu = 35$ give the best average performance and are thus chosen.

7.3.4 Calibrating the Stopping and the Diversification Criterion

Recall that I^{NI} is the maximum number of iterations the HGSADC can run without any improvements on the best individual. Reducing I^{NI} would decrease the number of iterations and the computational time. Since the HGSADC is non-deterministic, reducing I^{NI} might result in deterioration of the objective value. If the best individual does not improve for $I^{DIV} = \eta^{DIV} \times I^{NI}$ iterations, the diversification procedure is called. As I^{DIV} is dependent on I^{NI}, I^{NI} and η^{DIV} are tested together.

Table 7.9 shows the average running time and gap for different combinations of I^{NI} and η^{DIV}. Averagely, the combination $I^{NI} = 10000$, $\eta^{DIV} = 0.2$ and $I^{NI} = 7500$, $\eta^{DIV} = 0.6$ performs best. The former combination performs better on the large instances and is therefore chosen. The combination $I^{NI} = 10000$, $\eta^{DIV} = 0.2$ implies that the diversification procedure is called five times after the best individual is found before the algorithm terminates. This process introduces a significant amount of new genetic material, which revives the search. There is a trade off between average gap and computational time when choosing this combination. The computation times are however considered acceptable for all configurations.

Figure 7.1 presents the best found solution plotted against iteration count of a run of the algorithm on instance 100_35 with $I^{NI} = 10000$, $\eta^{DIV} = 0.2$. After 1750 iterations, an individual with gap of 5.6 percent is found. The algorithm searches for an improving individual for 8300 iterations before a slightly improved individual is found after 10040 iterations. After 17830 iterations multiple significantly better individuals are found until the best individual in this run is found after 22340 iterations with 4.5 percent gap. This clearly illustrates that setting I^{NI} greater than 7500 leads to better performance in some runs of the model.

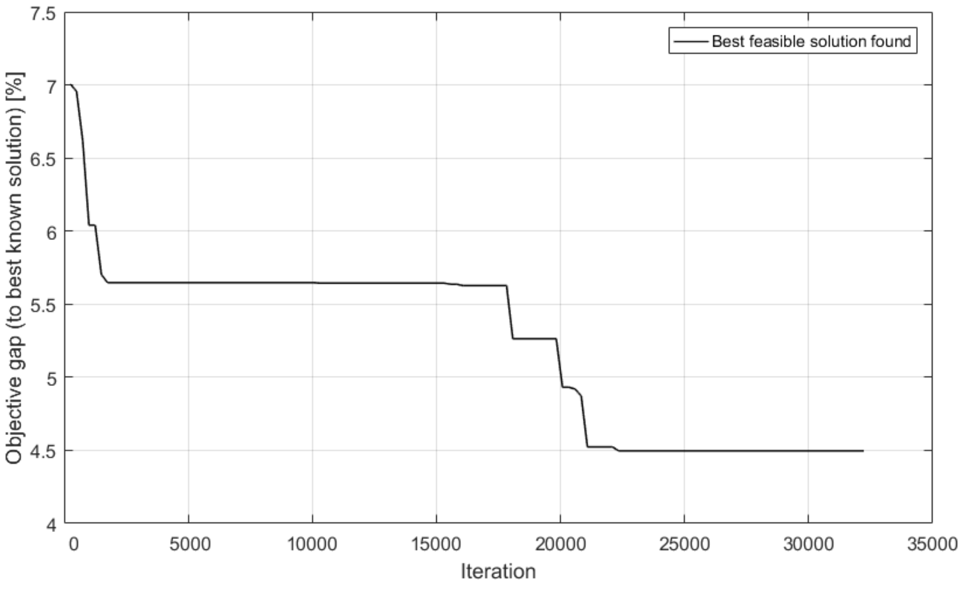

Figure 7.1: Plot of best solution found at different iteration numbers for a single run of the algorithm on the 100_35 instance with $I^{NI} = 10000$, $\eta^{DIV} = 0.2$.

7.3.5 Calibrating the Starting Values of the Penalty Parameters

The values of the penalty parameters ω^T and ω^V are adjusted dynamically by the algorithm in order to maintain the desired ratio between feasible and infeasible individuals. Tests are run with ω^T and ω^V equal to a 0.5, 2.0, 3.5, and 10.0. The tests revealed that the

Table 7.8: The average running time and gap from the best known objective value for different values of μ and λ. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	μ	$\lambda = 50$		$\lambda = 75$		$\lambda = 100$	
		Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	15	1.2	0.4	2.0	0.4	2.8	0.3
	25	1.8	0.3	2.4	0.3	3.2	0.3
	35	2.4	0.3	2.0	0.4	3.0	0.4
15_5	15	11.2	0.2	17.2	0.1	28.4	0.1
	25	8.4	0.5	13.6	0.2	31.6	0.1
	35	11.4	0.3	20.2	0.1	27.0	0.2
30_10	15	39.4	13.3	60.4	8.4	68.4	12.5
	25	44.0	9.5	89.0	9.6	61.0	10.0
	35	46.0	11.3	70.6	9.2	122.6	8.8
60_20	15	38.0	9.7	61.4	8.3	71.4	7.6
	25	34.2	7.3	69.0	6.5	84.6	7.4
	35	72.2	6.3	68.8	8.0	131.6	6.0
100_35	15	46.2	10.5	86.0	8.7	109.8	6.8
	25	70.0	6.9	123.8	6.6	154.0	6.2
	35	127.8	6.7	158.4	6.8	150.6	5.9

Best and second best average performance for each instance marked with dark and light green, respectively.

Table 7.9: The average running time and gap from the best known objective value for different values of I^{NI} and η^{DIV}. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	η^{DIV}	$I^{NI} = 5000$		$I^{NI} = 7500$		$I^{NI} = 10000$	
		Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	0.2	2.4	0.2	2.2	0.3	3.0	0.4
	0.4	1.6	0.4	2.0	0.2	3.0	0.3
	0.6	2.0	0.3	2.0	0.4	3.4	0.3
15_5	0.2	25.2	0.2	25.4	0.2	42.4	0.1
	0.4	17.2	0.2	20.2	0.1	29.6	0.1
	0.6	18.8	0.1	24.2	0.2	24.0	0.2
30_10	0.2	48.0	7.8	60.2	10.1	98.0	8.1
	0.4	36.6	12.0	83.4	7.7	89.2	8.6
	0.6	52.8	12.9	112.2	11.5	73.4	8.8
60_20	0.2	79.8	6.0	135.4	5.8	166.2	5.5
	0.4	88.2	5.8	94.2	6.0	74.8	6.9
	0.6	72.8	6.8	77.4	6.8	139.4	5.6
100_35	0.2	168.8	6.4	188.8	6.0	222.4	6.0
	0.4	111.0	6.2	218.0	6.2	245.6	6.7
	0.6	101.6	6.4	146.2	5.1	191.6	6.7

Best and second best average performance for each instance marked with dark and light green, respectively.

algorithm is robust against changes in these parameters. The performance of the algorithm is independent of the value chosen for ω^T and performs slightly better when ω^V is low. Therefore, $\omega^T = 2.0$ and $\omega^V = 0.5$ are chosen.

Figure 7.2 shows how the penalty parameters and the number of feasible and infeasible individuals develop throughout a search on the 100_35 instance. From the penalty parameter plot it is clear that the solutions found by the algorithm are not constrained by the number of available service vehicles as the service vehicle penalty quickly converges towards zero. Furthermore, we observe that the time constraint penalty parameter varies greatly but that the ratio of feasible individuals in the population is kept in the area of 60 percent, as defined in Section 7.3.2. Hence, the adaptive diversity control is successful in maintaining the desired ratio of feasible individuals. In addition, this demonstrates that the value of the penalty parameters are able to change rapidly, reducing the impact of the starting values.

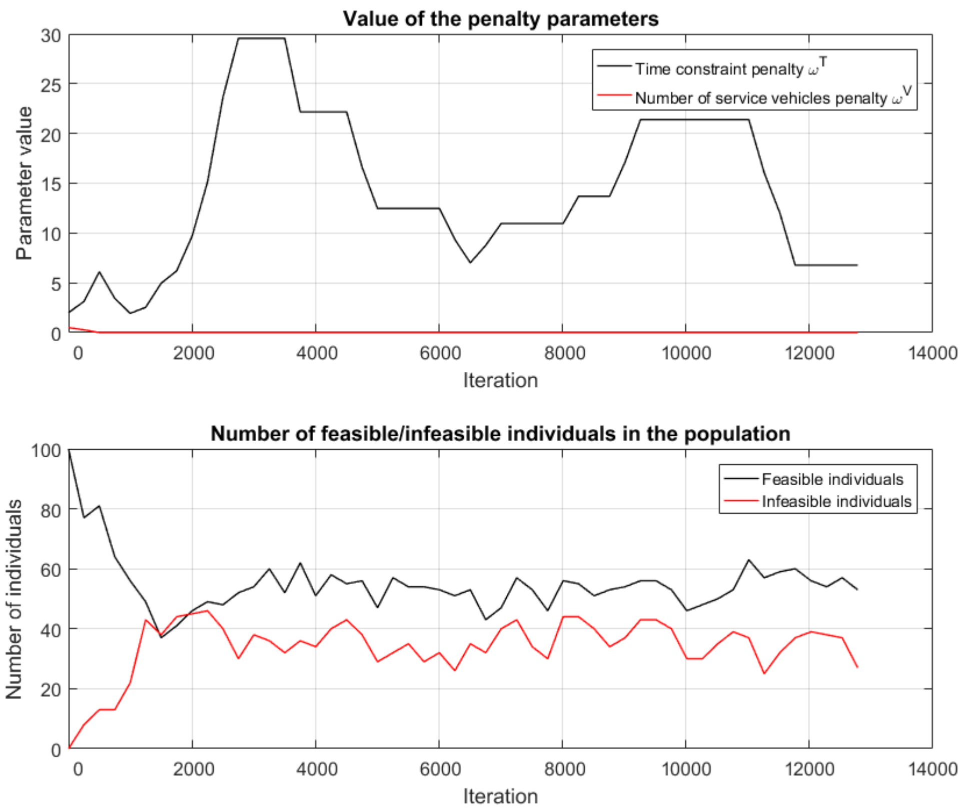

Figure 7.2: Plot of the penalty parameters and the number of feasible and infeasible individuals in the population of a run on instance 100_35. $\zeta^{REF} = 0.6, \omega^T = 2.0, \omega^V = 0.5$

7.3.6 Calibrating the Pseudo Time in the Construction Heuristic

Recall that K^{PSEUDO} is used in the construction heuristic to control how many rental cars that are assigned to an operator for handling. Preliminary testing showed that the feasibility of solutions generated by the construction heuristic varies greatly for different choices of K^{PSEUDO}. This is because K^{PSEUDO} in essence affects how aggressive the operator chromosome is determined. A high K^{PSEUDO} results in fewer rental cars assigned to each operator, making it more likely to complete all handling activities within the maximum time constraint. As the appropriate value of K^{PSEUDO} also varied significantly with the input data, K^{PSEUDO} is modified throughout the construction algorithm. Further preliminary testing reveals setting $K^{PSEUDO} = 1.5$ and increase it by 1 five times in the construction algorithm gives both feasible individuals while maintaining individuals with an aggressive operator chromosome. At the same time, setting K^{PSEUDO} dynamically contributes to the diversity of the generated population. Henceforth, K^{PSEUDO} is increased by one each time a fifth of the initial population is created.

7.3.7 Calibrating the Rate of Education and Repair

The education and repair procedures are strongly related as repair is using the education procedure with adjusted infeasibility penalty parameters to attempt to make solutions feasible. Hence, they are tested together. First, the rates to run education and repair when constructing the initial population are tested. Then, the rates to run education and repair on offsprings created in the genetic algorithm are tested. Preliminary testing reveals that solution quality increases when education and repair is performed. Furthermore, testing shows that repair cannot be run too often because it would lead to too many individuals being forced feasible and thus a suboptimal proportion of rental cars being postponed.

Table 7.10 shows the results of running education and repair with varying rates in the construction heuristic. The rates of education and repair after an offspring is created, $\rho^{EDU}_{crossover}$ and $\rho^{REP}_{crossover}$, are kept constant at 0.5 in this test. The results clearly demonstrate that $\rho^{EDU}_{construct} = 0.75$ produces solutions with the lowest average gap. The results for $\rho^{REP}_{construct}$ = to 0.25 and 0.5 are comparable on instances 6_3, 15_5, and 60_20. However, on instance 30_10 and 100_35, $\rho^{REP}_{construct} = 0.25$ performs best. Because the performance of $\rho^{REP}_{construct} = 0.25$ is equal or better than 0.5 for all instances, 0.25 is chosen.

In addition to the rates of education and repair in the construction heuristic, separate tests are performed with varying rates to run education and repair after an offspring is created. The results are presented in Table 7.11. None of the tested configurations were able to outperform the base configuration of $\rho^{EDU}_{crossover} = 0.5$, $\rho^{REP}_{crossover} = 0.5$. These parameter values are thus chosen.

7.3.8 Calibrating the Number of Individuals Created in the Initial and Diversification Population

K^{INIT} and K^{DIV} are multiplied by the minimum population size μ to determine how many individuals that are generated during the construction of initial solutions and diversification steps of the HGSADC, respectively. In these tests, K^{INIT} and K^{DIV} are kept

Table 7.10: The average running time and gap from the best known objective value for different values of $\rho^{EDU}_{construct}$ and $\rho^{REP}_{construct}$. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	ρ^{REP}_{cons}	$\rho^{EDU}_{cons} = 0.25$		$\rho^{EDU}_{cons} = 0.5$		$\rho^{EDU}_{cons} = 0.75$		$\rho^{EDU}_{cons} = 1.0$	
		Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	0.25	3.6	0.2	3.0	0.4	3.0	0.4	3	0.4
	0.5	3.0	0.3	3.0	0.4	3.0	0.4	3.4	0.3
15_5	0.25	33.6	0.2	35.2	0.1	35.2	0.1	30.6	0.2
	0.5	30.2	0.1	42.4	0.1	32.0	0.1	47.4	0.1
30_10	0.25	78.8	9.7	88.8	8.2	421.2	8.6	65.0	10.1
	0.5	81.4	8.8	98.0	8.1	335.4	9.0	79.8	8.4
60_20	0.25	234.2	5.5	148.8	7.2	257.0	5.2	154.6	5.6
	0.5	113.6	5.8	166.2	5.5	177.2	5.3	230.2	6.1
100_35	0.25	290.2	5.9	208.2	5.9	353.4	4.6	155.4	7.7
	0.5	287.4	5.7	222.4	6.0	413.0	5.1	329.8	5.3

Best and second best average improvement for each instance marked with dark and light green, respectively.

Table 7.11: The average running time and gap from the best known objective value for different values of $\rho^{EDU}_{crossover}$ and $\rho^{REP}_{crossover}$. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	ρ^{REP}_{cross}	$\rho^{EDU}_{cross} = 0.25$		$\rho^{EDU}_{cross} = 0.5$		$\rho^{EDU}_{cross} = 0.75$		$\rho^{EDU}_{cross} = 1.0$	
		Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	0.25	4.0	0.3	3.4	0.4	3.8	0.4	4.4	0.3
	0.5	3.0	0.4	3.0	0.4	3.2	0.4	4.0	0.3
15_5	0.25	25.4	0.3	33.6	0.2	32.8	0.2	39.4	0.1
	0.5	27.8	0.1	35.2	0.1	27.6	0.2	33.6	0.9
30_10	0.25	87.0	9.4	134.6	9.5	96.8	11.8	108.4	8.4
	0.5	134.0	8.7	421.2	8.6	63.2	11.4	202.8	6.0
60_20	0.25	105.4	7.0	97.4	6.5	159.4	6.9	171.4	8.2
	0.5	239.2	7.2	257.0	5.2	194.8	6.7	129	6.0
100_35	0.25	207.2	6.3	218.0	5.5	226.8	6.4	280.0	6.0
	0.5	245.8	6.2	353.4	4.6	343.8	7.0	243.2	7.6

Best and second best average improvement for each instance marked with dark and light green, respectively.

equal. Four values are tested, K^{INIT} and K^{DIV} equal to four, ten, 20, and 30. Larger values than 30 are not tested because preliminary testing revealed that the computation times increase significantly for the large instances and the construction algorithm struggles to find such a high number of unique individuals for the small instances. The results clearly show that larger K^{INIT} and K^{DIV} improve the solution quality of the algorithm. This is natural as more solutions are evaluated. Values 20 and 30 produce comparable results, with K^{INIT} and $K^{DIV} = 20$ performing slightly better for the 100_35 instance. Furthermore, the average computation time of 1722.4 seconds for K^{INIT} and $K^{DIV} = 30$ is becoming undesirable. K^{INIT} and K^{DIV} equal to 20 stand out as the best trade off between solution quality and computation time and are thus chosen.

Table 7.12: The average running time and gap from the best known objective value for different values of K^{INIT}. All tests are run with K^{DIV} = K^{INIT}. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	$K^{INIT} = 4$		$K^{INIT} = 10$		$K^{INIT} = 20$		$K^{INIT} = 30$	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	3.0	0.4	4.4	0.2	11.4	0.2	29.2	0.2
15_5	35.2	0.1	30.2	0.0	43.4	0.0	43.6	0.1
30_10	421.2	8.6	86.8	10.3	136.4	8.4	172	8.6
60_20	257.0	5.2	140.8	6.4	169.4	5.1	206.8	4.8
100_35	353.4	4.6	283.2	5.0	596.2	4.2	1722.4	4.6
Average	214.0	3.8	109.1	4.4	191.4	3.6	434.8	3.6

Best and second best average performance for each instance marked with dark and light green, respectively.

7.3.9 Calibrating the Proportion of Elite Individuals

Table 7.13 shows the average of the running time and objective gap for the five instances with η^{ELI} set to 0.25, 0.5, and 0.75. η^{ELI} is a parameter that influences the number of elite individuals to survive to the next generation. A higher value of η^{ELI} increases the weight of the total cost rank relative to the diversity rank in the biased fitness function described in Section 6.4. $\eta^{ELI} = 0.5$ and $\eta^{ELI} = 0.75$ clearly outperforms $\eta^{ELI} = 0.25$ on instances 30_10, 60_20, and 100_35. $\eta^{ELI} = 0.5$ and $\eta^{ELI} = 0.75$ have comparable performance on instances 6_3, 15_5, and 30_10. However, $\eta^{ELI} = 0.5$ performs significantly better on instances 60_20 and 100_35. Although $\eta^{ELI} = 0.5$ has somewhat longer computation time, the performance gains are large enough to defend added computational effort. Hence, $\eta^{ELI} = 0.5$ is chosen.

7.3.10 Final Remarks on the HGSADC Calibration

The parameter calibration of the HGSADC consists of 230 runs of the algorithm for each of the five training instances. In addition, a large number of algorithm runs was performed in preliminary testing. The values tested for each parameter are based on the paper by Vidal et al. (2012) and our preliminary testing. We believe that the parameter values

Table 7.13: The average running time and gap from the best known objective value for different values of η^{ELI}. The average gap from the best known objective value is calculated as the gap between the average objective value and the lowest known objective value for the instance. All averages are calculated from five runs.

Instance	$\eta^{ELI} = 0.25$		$\eta^{ELI} = 0.5$		$\eta^{ELI} = 0.75$	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3	18.4	0.2	13.0	0.2	11.4	0.2
15_5	57.0	0.3	53.2	0.0	43.4	0.0
30_10	128.8	10.8	127.6	10.3	136.4	8.4
60_20	243.0	6.5	450.8	2.5	169.4	5.1
100_35	435.8	5.9	764.8	2.7	596.2	4.2
Average	176.6	4.7	281.9	3.1	191.4	3.6

Best and second best average performance for each instance marked with dark and light green, respectively.

chosen in this chapter are parameters that will cause the algorithm to produce high quality solutions in reasonable computation time. However, better configurations may exist, as it is intractable to test all values and combinations of the parameters. As the parameter tuning is done incrementally, final configuration performance on the training instances can be found in Table 7.13 where $\eta^{ELI} = 0.5$. The chosen configuration produces solutions with less than 2.7 percent gap for all instances except instance 30_10. The relatively large gap and variation in objective value for this instance are believed to be caused by the characteristics of the input data.

The objective value gap is affected by the deviation and postponement costs in the objective function of the SFFEVCHP. These are artificial costs introduced to evaluate the trade off between the cost of unmet demand due to a disadvantageous distribution of rental cars and the cost of transporting operators to charge and reposition the rental cars. The artificial costs are high compared to the real costs. E.g. for the 100_35 instance, one additional postponement would increase the total cost with approximately one percent. As a result, the algorithm may report large gaps between solutions that in real life would be very similar.

Several chosen parameter values deviate from the values chosen as the basis of our testing and those used by Vidal et al. (2012). The minimum population size μ, generation size λ, max number of iterations without improvement I^{NI}, proportion of elite individuals η^{ELI}, and the proportion of individuals considered in the diversity contribution η^{CLO} have the same or nearly the same values as Vidal et al. (2012). The target ratio of feasible individuals is set significantly higher, $\zeta^{REF} = 0.6$, compared to 0.4 in Vidal et al. (2012). This is likely because it is hard to make infeasible individuals feasible without making large alterations to the chromosomes. Hence, it is troublesome to produce high quality feasible individuals from infeasible individuals in the SFFEVCHP.

Furthermore, the education and repair rates differ. First, our testing revealed that different rates in the construction and offspring creation phase are beneficial. In the construction phase, $\rho^{EDU}_{construct} = 0.75$ and $\rho^{REP}_{construct} = 0.25$ proved to be the best configuration, both 0.25 lower than the values of Vidal et al. (2012). The lower education rate is likely because a large number of individuals are generated in the construction phase and thus no significant improvement was observed by having a higher rate. The lower repair rate is probably due to the fact that the repair module guarantees feasibility by postponing handling of rental cars if necessary. If performed too often it will lead to overly pessimistic solutions, reducing the probability of finding high quality solutions when creating offsprings. In the offspring generation phase, the repair rate is equivalent to that found by Vidal et al. (2012) but the education rate found here is lower. We believe this is because the changes made in the handling sequence chromosome by the education module may not merge well with other individuals when new offsprings are created at later stages in the algorithm.

In addition to the aforementioned parameters, the diversification rate found here, $\eta^{DIV} = 0.2$, is lower than 0.4 found by Vidal et al. (2012). This is likely because the education and repair procedures increase the homogeneity of the population. Thus, a higher rate of inserting new genetic material is performance improving. Also, the number of individuals created in the construction and diversification stages are considerably higher, $K^{INIT} = K^{DIV} = 20$, than 4 used by Vidal et al. (2012). We strongly believe that this is caused by the dynamic setting of K^{PSEUDO} throughout the construction heuristic. In effect a fifth of the generated population is created with a given K^{PSEUDO}, equating the generation size multiple found by Vidal et al. (2012) for each value of K^{PSEUDO}.

7.4 Results

In this section, the results of running the final configurations of the MIP and the HGSADC are presented, compared, and discussed. First, the maximum solvable instance size is explored and the MIP and the HGSADC are compared in Section 7.4.1. Then, results from running the HGSADC on a large number of instances are reported and discussed in Section 7.4.2

7.4.1 Performance of the MIP and HGSADC

Figure 7.3 shows how the computation times of the MIP and the HGSADC are affected by the instance size. The MIP can only solve small instances to optimality within a computation time of an hour. Two out of three instances with six rental cars in need of handling are solved to optimality in 662 and 32 seconds, respectively. The remaining instances are stopped after running an hour with gaps at the end of the computation time ranging from four to 270 percent. Hence, the MIP cannot reliably produce high quality solutions for instances with more than six rental cars in need of handling. The plot clearly demonstrates that the HGSADC scales significantly better with increasing instance size. The algorithm can solve instances with 200 rental cars in need of handling with acceptable computation time and objective value stability. Several of the algorithm runs on the tested instances are using the entire computation time of an hour. The runs lasting an hour do however produce solutions with less than two percent gap from the best known solution, so the algorithm may be applied to even larger instances. In this thesis, the ability to solve instances with 100 to 150 rental cars reliably was sought. Hence, solving of larger instances has not been tested.

Furthermore, Figure 7.3 reveals that the HGSADC for the SFFEVCHP scales approximately linearly with the number of rental cars in need of handling. The instances with 175

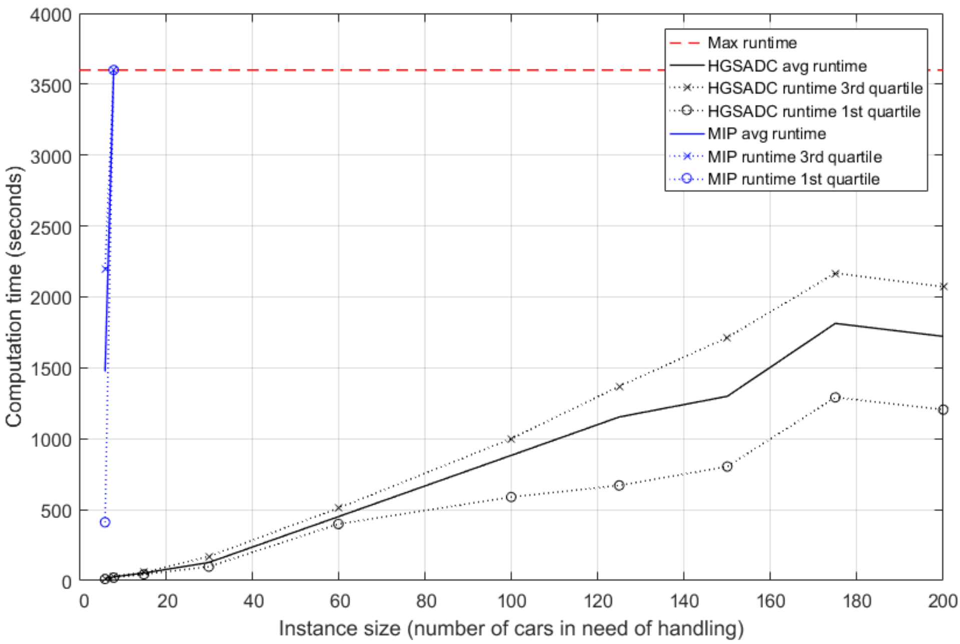

Figure 7.3: Plot of the average run time and quartiles of the MIP and the genetic algorithm (HGSADC).

rental cars in need of handling deviates from the linear trend, requiring longer computation time. Especially one of the three tested data files require longer time, with an average run time of 2,137 seconds. This is believed to be because the data file has almost as many available charging slots as rental cars in need of handling (161 available slots), resulting in a large feasible solution space to explore. This does however demonstrate that the algorithm manages input data with this characteristic as the gaps for this instance remains below 2.7 percent for all ten runs of the algorithm.

The HGSADC consists of many modules making it a relatively complex algorithm to design and implement. To rationalize the added complexity, it is essential that the algorithm provide a significant improvement in solution quality and/or computation time. To provide evidence of the value of the HGSADC, table 7.14 compares the results of running the MIP and the HGSADC with different modules on the same instances. Each instance is run five times when solved with the HGSADC due to randomization. The average computational times and gaps are reported. The first column reports the results of running the MIP for the solvable instances. In the second and third column, the results of running only the construction module of the algorithm without and with education and repair, respectively, are presented. In the latter case, the construction heuristic is run with the parameter values decided in Section 7.3 and the best solution found at the end of the construction heuristic is returned. Then, the fourth column presents the results of running the algorithm without education and repair in neither the construction algorithm nor after an offspring is created. Finally, the results of the full algorithm with all parameters as described in 7.3 are shown in the last column.

Table 7.14: The average running time and gap from the best known objective value reported for runs of the MIP, the construction heuristic without education and repair, the construction heuristic with education and repair, the HGSADC without education and repair, and the HGSADC with education and repair, respectively.

Instance	MIP[1]		CH[2]		CH + E/R[3]		HGSADC[4]		HGSADC+E/R[5]	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
6_3*	32.1	0.0	>1	0.1	3.6	0.0	7	0.0	22.8	0.0
8_4	3600	9.2	>1	18.6	>1	11.0	26.2	2.8	26.4	0.7
30_10	N/A	N/A	N/A	N/A	2.0	18.5	66.2	19.2	127.6	10.3
100_35	N/A	N/A	N/A	N/A	18.2	7.1	N/A	N/A	764.8	2.7

*Proven optimal.

1: Implementation of the MIP.

2: The construction heuristic without education and repair.

3: The construction heuristic with education and repair as configured in Section 7.3.

4: The HGSADC without education and repair in both the construction heuristic and the crossover.

5: The HGSADC with all configurations from 7.3.

The MIP finds the optimal solution for the instance with six rental cars and three charging stations, but is unable to find optimal solution for the instance with eight rental cars and four charging stations within the maximum computation time. Using only the construction heuristic without education and repair, solutions with an average gap of 0.1 percent and 18.6 percent are found for the two smallest instances. No feasible solutions can be found for the two remaining instances. When education and repair are introduced in the construction heuristic, the optimal solution is found for the smallest instance and feasible solutions are found for all test instances. Comparing the third column and the fifth column clearly demonstrates the value of the hybrid genetic algorithm. The construction heuristic is not able to find feasible solutions for the 30_10 instance, but using the HGSADC feasible solutions with an average gap of 19.2 percent from the best known solution is found. Furthermore, the average gap for the 6_3 and the 8_4 instance are reduced to 0.0 and 2.3 percent, respectively. Finally, the last column illustrates that the HGSADC including all modules clearly outperforms both the MIP and the other modules of the HGSADC.

7.4.2 Final Results

To demonstrate the capabilities of the proposed algorithm, it has been tested on 15 problem instances. To avoid overfitting of the parameters to the data, none of these data files were used in the parameter calibration in Section 7.3. Three instances for each size of 100, 125, 150, 175, and 200 rental cars in need of handling have been tested. The remaining characteristics of the data files are discussed in Section 7.1.

Each instance is run ten times and the average run time, average gap after ten minutes, average gap at end of the algorithm execution, and coefficients of variance of the gap and computation time are reported in Table 7.15. The average run time for the tested instances range from 693.5 to 2403.4 seconds. However, there is a relative large variation in run time indicated by the average coefficient of variance of the run time of 45.6 percent.

One of the runs of the 100_35 instances, two of the 125_40 instances, two of the 150_45 instances, two of the 175_50, and three of the 200_55 instances ran for 3600 seconds, the maximum run time. However, all of these runs found a solution with less than 4.1 percent gap after ten minutes and less than 3.5 percent gap at the end of the execution, making the solutions usable for most practical purposes. There are several factors contributing to the large variation in run time. The most important reason is simply randomization. The search may progress in a manner such that many, smaller improvements are found so that the maximum number of iterations without improvements is never reached. Another observation that can be made is that the average gap to the best known solution and the coefficients of variance of the gap and run time decrease with increasing instance size. This indicates that the algorithm performs consistently for all problem sizes.

The gap after ten minutes (600 seconds) is reported because we assume that running the algorithm for a maximum of ten minutes is desirable in real life scenarios. We show that the average gap to the best found solution after ten minutes is 1.9 percent with no averages above 2.7 percent. Furthermore, the average coefficient of variation of the objective value after ten minutes is equal to the value at the end of the algorithm execution at 0.9 percent. This demonstrates that the algorithm is able to produce acceptable solutions reliably within ten minutes. The average gap at the end of the algorithm execution is 1.3 percent. For the largest instances with 200 rental cars, these numbers are even lower, with average gap of 1.0 percent and coefficient of variation of 0.6 percent. These results are a clear indicator of the capabilities of the algorithm to produce consistent, high quality solutions for realistic carsharing systems.

Table 7.15: Final results of running the HGSADC on 15 instances with 100 to 200 rental cars in need of handling.

Instance	Avg. time (s)	Avg. gap % after 600s	Avg. gap %	Coeff. of Var. gap %	Coeff. of Var. time %
100_35_a	1212.1	2.6	1.9	0.9	76.4
100_35_b	693.5	1.6	1.5	1.3	32.7
100_35_c	743.5	1.3	1.2	0.9	29.3
125_40_a	1185.8	2.7	2.0	1.6	85.5
125_40_b	1302.6	2.5	1.9	1.2	68.2
125_40_c	972.7	1.7	1.3	0.6	49.1
150_45_a	1774.9	2.6	1.4	1.0	48.8
150_45_b	760.7	1.0	1.0	0.8	26.0
150_45_c	1362.9	2.1	1.3	0.9	47.4
175_50_a	1453.7	1.8	1.2	1.0	32.2
175_50_b	1849.1	2.9	1.7	1.1	51.1
175_50_c	2137.4	1.8	0.6	0.5	31.2
200_55_a	1496.9	1.5	1.0	0.4	26.4
200_55_b	1265.5	1.0	0.6	0.5	36.7
200_55_c	2403.4	2.1	1.2	0.8	42.8
Average	1374.3	1.9	1.3	0.9	45.6

To investigate the stability of the algorithm further, 100 runs on the 100_35_c instance have been executed. The results of these runs are presented in the histograms in Figure 7.4. As can be seen from the plot, the mean objective value gap to the best known solution is 2.0 percent. 58 out of the 100 solutions found have equal or smaller gap than the mean. Of the 42 solutions with objective value gap above the mean, 39 are below 4.0 percent. The remaining three solutions have gaps of 4.3, 5.1, and 5.1 percent, respectively. The mean run time of the algorithm is 655.7 seconds. 61 of the 100 algorithm executions completed in less or equal run time as the mean. 91 of the runs completed in less than 1000 seconds. Of the remaining nine algorithm runs, eight completed in less than 1314 seconds and one outlier required 1970 seconds to complete. Based on these results we conclude that the algorithm is largely able to produce solutions with stable quality within a reasonable time for most executions of the algorithm. If compared to the capabilities and results of the MIP-model, the HGSADC represents a significant improvement because realistic problem sizes can be solved with acceptable gaps. In addition, the results demonstrate that the UHGS framework is suitable for problems with complex synchronization constraints, and spatial and temporal interdependencies. Further economic implications and sensitivity on input data will be studied in Chapter 8.

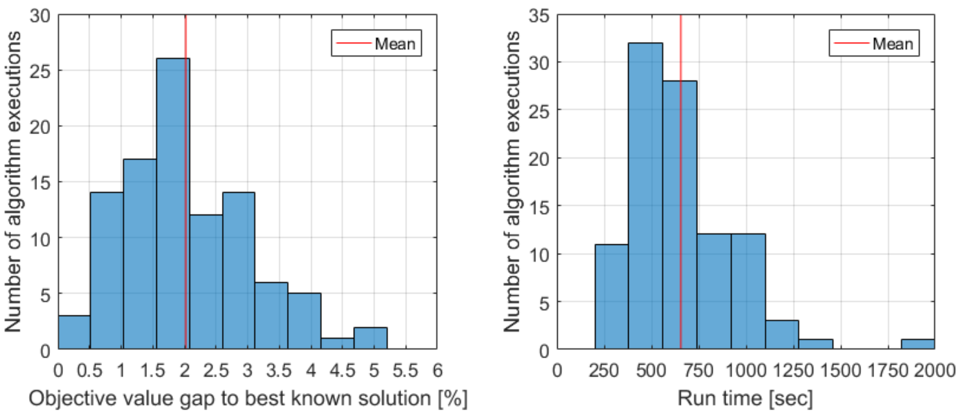

Figure 7.4: Histogram of objective value gap to the best known solution and run time from running the 100_35_c instance 100 times.

Chapter 8

Practical Use of the Model and Solution Algorithm

This chapter discusses the practical use of the HGSADC for the SFFEVCHP. The goal is to demonstrate how the presented model and algorithm can be used as decision support for real life CSOs. Section 8.1 presents operational insights obtained by altering different parameters central to the problem. Moreover, Section 8.2 discuss the general features and the usefulness of the model.

8.1 Operational Insights

This section takes a closer look on some operational insights deducted from varying input data and configurations of the HGSADC. First, the value of combining handling and repositioning is discussed in Section 8.1.1. Furthermore, the value of added operators and service vehicles, and the value of added handling time is discussed in Section 8.1.2 and Section 8.1.3, respectively.

8.1.1 The Value of Repositioning

An important assumption of the SFFEVCHP is the benefit of performing repositioning simultaneously as the fleet of rental cars is handled. As discussed in Chapter 4, repositioning is necessary to maintain a profitable carsharing system. At the same time, repositioning is a costly procedure requiring operators and transport of operators. A central hypothesis of this thesis is that combining necessary daily operations like handling with repositioning will increase the operational costs of the CSO marginally, while harvesting the full benefits of repositioning. Furthermore, this represents more realistic repositioning conditions for CSOs.

To investigate the effect of repositioning, two different configurations are compared. In the first configuration, all cars are either handled to the closest charging station or postponed. The first configuration is meant to represent the handling procedure without considering repositioning while the second configuration is the full HGSADC as described in

Chapter 6. An instance with 100 rental cars to handle has been run five times for each configuration and the average costs, number of deviations, and postponements are reported in Table 8.1. The average change when repositioning is considered is reported in the fourth column of the table, compared to when repositioning is not considered.

Table 8.1: Comparison of costs, and number of deviations and postponements when repositioning is omitted and when repositioning is performed. The numbers are the average of five runs with the HGSADC. The test instance has 100 rental cars in need of handling. The average change when repositioning considered is reported compared to when repositioning is not considered.

	No repositioning	**With repositioning**	**Change %**
Number of postponed cars	46.6	42.6	-9.4
Number of deviations	84.6	79.0	-7.1
Travel cost [Euro]	112.5	116.5	3.2
Service vehicle cost [Euro]	256.0	276	7.3
Operator cost [Euro]	1008.0	1076.0	6.3
Total cost [Euro]	4552.5	4388.2	-3.7

Improvements marked with green.

The table shows that the operational costs increase when repositioning is considered. The travel cost increases by 3.2 percent as the rental cars are moved to charging stations farther away. In addition, the cost of service vehicles and operators increase by 7.3 and 6.3 percent, respectively. This implies that when repositioning is considered, more staff is prioritized in order to achieve a more favorable distribution of rental cars in the system. Nevertheless, the total cost of the system decreases with 3.7 percent when repositioning is considered. The reduction in total costs can be attributed to the decreased number of postponements and deviations. When both postponement and deviations are considered, the marginal benefit of handling increases resulting in more handled rental cars. Furthermore, when considering more than the closest charging station, the deviation cost decreases. As the total cost (objective function) includes lost profits when deviations are present and cars are postponed, the total cost to a large degree captures the profit effect of the repositioning operations. Hence, the 3.7 percent decrease in total costs can be directly transferred to gross profit margin improvement, thereby representing a significant improvement of the economic viability for the CSO.

The gains for a specific CSO will depend on its valuation of postponements and deviations. The numbers reported here should therefore be used as an indication on the potential benefits of combining handling and repositioning. We believe that the total benefit of performing handling and repositioning simultaneously are even greater than stated in Table 8.1. If no repositioning is performed alongside with handling, a separate handling procedure should be performed. This would result in a significant increase in operator and service vehicle costs, as more staff would be required to perform this procedure. Furthermore, the travel costs would increase. Hence, modeling a carsharing system as the SFFEVCHP increases the operational costs of a CSO marginally, but gives the full benefits of repositioning.

Some CSOs might desire to apply the HGSADC without repositioning. Handling without repositioning might closer resemble these CSOs' operational mode, or the deviation cost in these systems may be too cumbersome to derive. When the HGSADC is run with no repositioning, the computational time is reduced by 61.2 percent and the stability of the solutions increase due to a smaller search space. This implies that the HGSADC can be highly valuable even if repositioning is not done in conjunction with charging.

8.1.2 The Value of Added Operators and Service Vehicles

In the SFFEVCHP, a given number of operators and service vehicles are available at the depot. By varying these numbers, the marginal benefit of added operators and service vehicles can be found. Figure 8.1 shows the plot of the objective value, number of rental cars handled, number of vehicles used, and number of operators used by the HGSADC solution when the number of available service vehicles is varied. The plot is an average of five runs and the test instance has 100 rental cars in need of handling. The number of operators is increased linearly with the number of service vehicles. We have assumed a service vehicle capacity of four operators, thus keeping a one to four ratio of service vehicles and operators is reasonable. As expected, the objective value decreases as the number of operators and service vehicles increase. The improvement in the objective function can mainly be attributed to the increased number of handled rental cars. The marginal increase in number of rental cars handled are 1.6 for each added service vehicle. The benefit of handling more rental cars exceeds the added costs of service vehicles and operators, and the increased travel costs. This results in an average marginal net benefit of 62.5 Euros for each additional service vehicle when increasing from two to fourteen service vehicles.

The marginal benefit reported here is only valid for CSOs with equal operational costs and valuation of postponement and deviation. By running the algorithm using their own cost estimates, CSOs can find the marginal benefits applicable for their system. Nevertheless, a clear insight from Figure 8.1 is that the number of available operators and service vehicles used for handling and repositioning has a significant effect on the profitability of the system. When no more rental cars can be handled due to e.g. shortage of charging slots, the marginal added benefit of an added service vehicle will equal zero. By running the algorithm on historic data sets representative for typical conditions in the system and with the desired planning time, the strategically optimal number of operators to hire and service vehicles to invest in can be derived.

8.1.3 The Value of Added Handling Time

The length of the planning period is an important decision for the CSO. The length of the planning period affects the number of operators and service vehicles needed to perform the handling and the proportion of cars that are postponed. An increased planning period requires fewer operators and service vehicles to handle the same amount of rental cars, and gives the possibility to handle more cars. On the other hand, an increased planning period could result in higher deviation and postponement costs as the time until next handling increases. This implies that the system will have an unfavorable distribution of cars for a longer period and that cars not handled will cause the CSO to lose more trips, likely to also affect customer satisfaction negatively. Furthermore, the salary of an operator per planning period increases as planning time increases, offsetting some of the cost savings of having fewer operators.

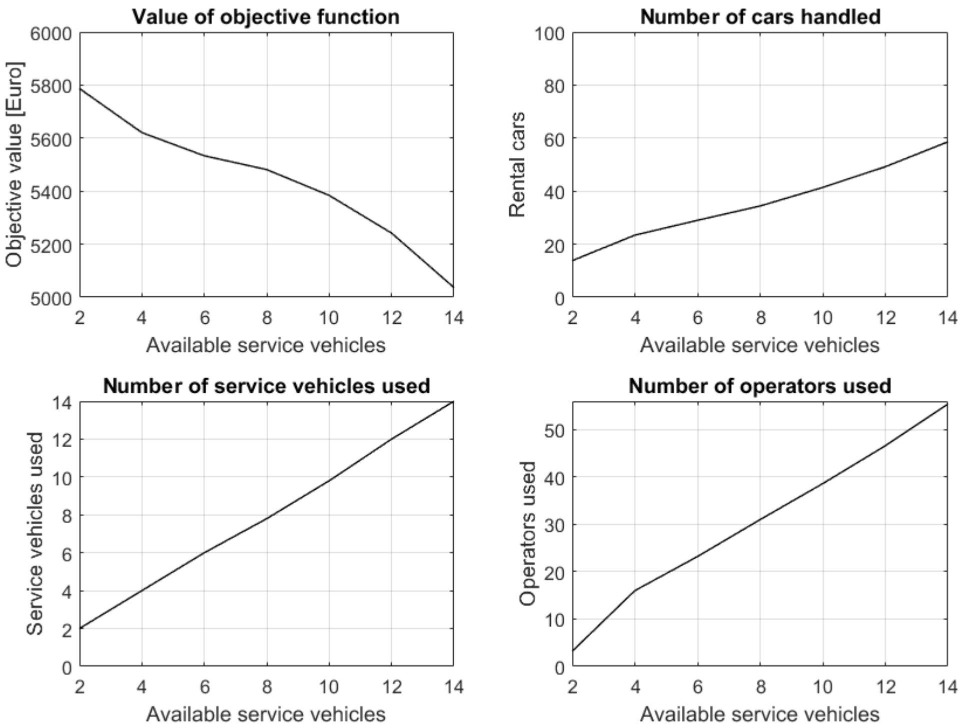

Figure 8.1: The plots of the average objective value, number of rental cars handled, number of vehicles used, and number of operators used when the number of available operators and service vehicles are varied. The number of operators increases linearly by a factor of four as the number of service vehicles increases. There are 100 rental cars in need of handling.

Figure 8.2 shows the plot of the objective value, number of rental cars handled, service vehicles used, and operators used when the planning period is increased. The plot is an average of five runs and the test instance has 100 rental cars in need of handling. As the figure shows, the number of handled rental cars are stable and the number of service vehicles and operators used decrease when the handling time increases. Nevertheless, the total cost increase due to the increased deviation and postponement cost. This implies that shorter handling times are favorable for an effective utilization of the system and improved customer satisfaction. The stable number of rental cars handled indicate that the optimal number of cars to handle is governed by the trade off between repositioning costs and postponement costs, not strictly limited by the planning period duration. This trade off will depend on each CSO's valuation of lost demand and deviations.

An important decision for the CSO is the trade off between a larger pool of operators and investments in service vehicles that enable shorter planning times and utilization of the system. Using real data and cost estimates from the CSO, the HGSADC can be used to analyze this trade off for different scenarios and hence work as a decision support for the CSO.

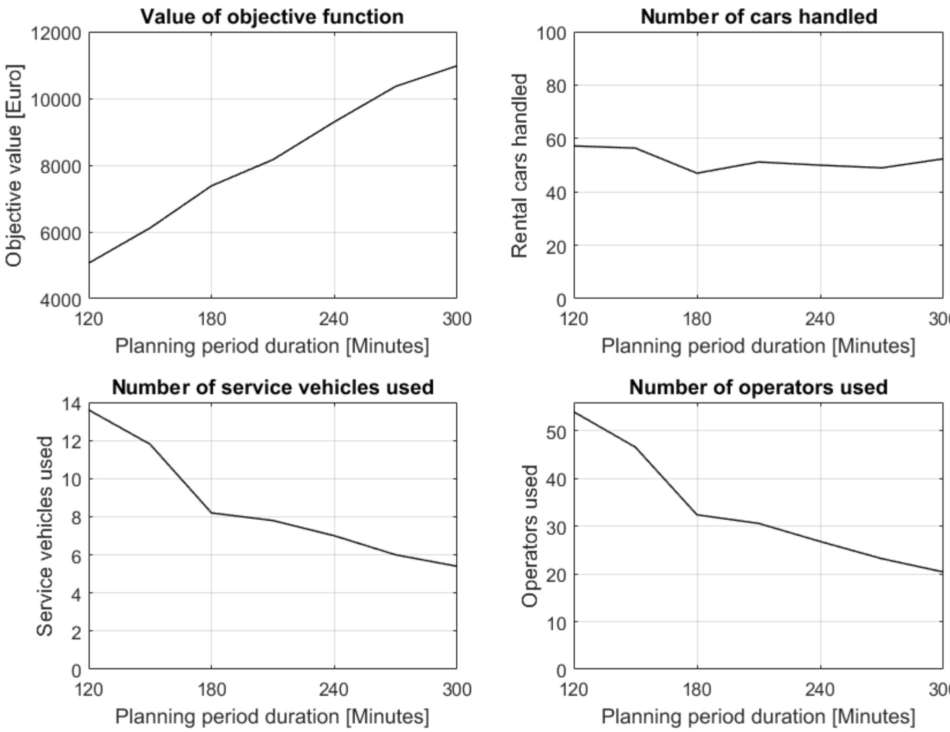

Figure 8.2: The plots of the average objective value, number of rental cars handled, number of vehicles used, and number of operators used when the planning period is increased. There are 100 rental cars in need of handling.

8.2 Benefits of the Proposed Model

In this section, the benefits of the proposed model and algorithm are discussed. In Section 8.2.1 the main advantages of the proposed model are described, while Section 8.2.2 elaborates on how the model can be implemented as a decision support tool for a CSO.

8.2.1 Advantages of the Proposed Modeling Approach

An important element of the SFFEVCHP is the total cost perspective, including routing of operators and service vehicles, as well as the cost of postponing rental cars and deviations from the ideal number of rental cars in an area. The integrated nature of the decisions on handling, routing, and repositioning makes it necessary to formulate an integrated model to take the required trade offs into account. Hence, the model proposed is well suited to be used as decision support for CSOs.

Another strength of the formulation is the division of the business area into smaller areas surrounding the charging stations. By varying the size of the areas, the model can indirectly factor in the flexibility of users, as described by Correia et al. (2014) and discussed in Chapter 4. This can have positive effects on the profitability of the system. Increasing the size of the areas results in demand data being aggregated, and thus counting in that users are willing to walk longer to pick up a rental car. In addition, larger areas lead to a smaller problem size thus making the SFFEVCHP easier to solve. Finally, the forecasting of demand will have increased likelihood of being correct as aggregation reduces the variance.

8.2.2 Applying the Model to Real Life Carsharing Systems

Modern carsharing systems often rely on mobile applications that allow customers to find available cars and make reservations. This is particularly relevant for free-floating systems, as it is crucial that users are able to locate cars. Therefore, the CSO must have GPS tracking of the rental cars. Furthermore, an app-based service allows companies to gather large amounts of data that can be used in the optimization of its handling and repositioning operations.

By utilizing this source of data, the number of cars in need of handling and the initial state of the system can be extracted in real time. Furthermore, travelling times can be calculated based on e.g. Google Maps data in real time, including traffic, or be based on historical data. If the operator performs handling and repositioning throughout the day, historic data for travel times can be defined for different times during the day. Finally, cost parameters can be estimated using historic data of demand, as well as using the CSO's operational experience, salary levels, and asset-related costs. Based on historical data, the operator can define the ideal state for different times during a day to incorporate recurring trends in the demand pattern, e.g. rush hour traffic into the city in the morning and out of the city in the evening.

As a result of these factors, the algorithm can handle real world scenarios well by intelligently collecting and pre-process the input data. If cars that require handling are made unavailable in the booking system, the HGSADC can easily be applied to handling and repositioning during the day, without modifications. These operations will then be performed based on information about handling requirements, traffic, and states available when the algorithm execution is started. If the model is integrated in a program that collects and prepares data based on real time and historic data, we believe it can contribute to a significant increase in the efficiency of the handling and repositioning operations of the CSO. Furthermore, it is expected that CSOs with a large number of rental cars will realize the biggest benefit of employing the algorithm. A high number of cars is likely to increase the density of cars in the business area, enabling the service vehicles to drop off more operators per distance driven. In addition, more operators can be picked up at the same time when a charging station is visited. Taking advantage of the economics of scale can be troublesome if planning is done semi-manually but the HGSADC is highly suitable to exploit these effects.

Chapter 9

Concluding Remarks

In this chapter, we summarize the major findings of this thesis and discuss attractive future research areas. First, a conclusion is presented in Section 9.1 and finally, future research opportunities are discussed in Section 9.2.

9.1 Conclusion

This thesis presents a mathematical formulation and a Genetic Algorithm (GA) for the Static Free-Floating Electric Vehicle Carsharing Handling Problem (SFFEVCHP). The studied problem is concerned with charging and repositioning of rental cars in a free-floating carsharing system of electric vehicles. Discussions with carsharing organizations revealed that repositioning is costly but desirable to balance the system to better serve demand. Many companies already have a fleet of service vehicles and staff to move rental cars to charging stations. Henceforth, considering repositioning to improve the distribution of cars in the system while moving cars to charging stations shows potential to realize the benefits of repositioning without a large increase in operational costs. Ultimately, this will improve the profits and the economic viability of carsharing systems.

A novel model that minimizes the total costs of moving rental cars to charging stations is proposed. In addition, the model assigns costs to rental cars that have an unfavorable location in the business area and allows rental cars to be moved to improve the distribution while being handled. The model finds optimal routes for service vehicles transporting operators to handle cars and determines to which charging station to move each rental car. This integrated approach, taking the trade off between repositioning costs and deviations into account, while handling the fleet of rental cars has, to our knowledge, not been presented before. The model is formulated as a Mixed Integer Program (MIP) and solved by Xpress. The problem is however computationally cumbersome to solve, resulting in intractable computation times. Hence, a GA based on the Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) presented by Vidal et al. (2012) is developed to solve realistic problem sizes.

The MIP is able to solve the problem to optimality for instances with six rental cars in need of charging in less than 3600 seconds. The computation time scales exponentially with the number of rental cars in need of charging. Real problem sizes are in the range of 100 to 150 rental cars in need of charging making the MIP unsuitable. The HGSADC is able to find the optimal solution for all instances tested solvable by the MIP. Furthermore, the computation time of the HGSADC scales linearly with the number of rental cars in need of charging and is capable of solving instances with up to 200 rental cars yielding seemingly high quality solutions in an average computation time of less than 2400 seconds. The stability of the algorithm is acceptable for practical purposes with an average gap to the best known solution of 1.3 percent and an objective value coefficient of variance of 0.9 percent. More importantly for practical applications, the average of the gaps to the best known solutions after 600 seconds for all instances tested is 1.9 percent with a 0.9 percent coefficient of variance. Furthermore, these values indicate larger variations than what is true in practice. This is caused by the presence of semi-artificial costs in the objective function resulting in rather large changes in the objective value with small changes in the solution chromosomes.

The economic benefit for a carsharing organization implementing the proposed algorithm is twofold. First, the algorithm is largely capable of solving the essential problem of routing operators to move rental cars to charging stations. This procedure is necessary to maintain a functioning service. We have not succeeded in obtaining comparable solutions from existing companies to compare to the solutions produced by the algorithm. However, as the route planning typically is performed semi-manually we believe that significant efficiency gains are possible. Secondly, the element of repositioning rental cars to improve the distribution in the system represents an added value for the organizations. When comparing solutions from the HGSADC with solutions produced by the algorithm when prohibiting repositioning, a net reduction in the total cost of 3.7 percent is obtained. As the total cost function also includes lost profits when the distribution of rental cars is unfavorable, the total cost approximately captures the profit effect of the daily operations. As a result, the observed improvements can be transferred directly to the gross profit margin of the companies.

The HGSADC developed for the SFFEVCHP demonstrate the performance of the Unified Hybrid Genetic Search (UHGS) framework developed by Vidal et al. (2014) on routing problems with complex synchronization constraints. The SFFEVCHP consists of two closely linked routing problems, one for the routing of rental cars to charging stations and one for routing service vehicles transporting operators to rental cars and from charging stations. As the drop off time and location of an operator affects the time and location of his/hers pick up, spatial and temporal interdependencies emerge. The work in this thesis outline the merit of genetic algorithms for solving this complex problem type. In conclusion, solving the SFFEVCHP with the HGSADC produces high quality solutions within reasonable computation time for realistic problem sizes. We consider this a significant contribution to creating efficient and profitable carsharing systems.

9.2 Future Research Opportunities

The SFFEVCHP presented in this thesis is a novel problem type designed to model real world carsharing systems studied. We regard the MIP and the HGSADC as an initial attempt in modeling and solving this problem type for realistic problem sizes. We believe that the developed algorithm will perform well if combined with a modern data collection system and far outperform manual or partial optimization schemes currently available. Nevertheless, this section highlights further research opportunities we believe to be important to improve the algorithm developed and more generally create efficient, profitable carsharing systems. First, high level research areas are discussed in Section 9.2.1. Then, possible extensions of the proposed model are introduced in Section 9.2.2. Finally, possible improvements to the HGSADC algorithm are presented in Section 9.2.3

9.2.1 High Level Research Topics

The pricing of the carsharing service is the major instrument to influence how the users behave in the system. If prices are set dynamically it is e.g. possible to reduce the imbalances in the system drastically by making disfavorable trips more expensive and trips that improve the distribution of rental cars in the system cheaper. Furthermore, the users can be incentivized to leave rental cars at charging stations or perform other minor maintenance. These efforts may reduce or eliminate the need of interventions by the CSO to charge and redistribute cars. Hence, research on the pricing problem can have large effects on the economic viability of the systems.

Another research opportunity is optimizing the charging of EVs taking power grid balancing into account. Large CSOs with a fleet of EVs represent a large power storage capacity. By managing charging and connection to the power grid of the rental cars, the carsharing systems can play a role in balancing the power grid. This is especially attractive with the presence of volatile power sources such as wind and solar in the grid. Research on how to leverage this opportunity could provide a new revenue source for CSOs while at the same time further increase the utilization and environmental benefits of the rental car fleet.

9.2.2 Extensions of the Presented Model

Several extensions that can improve the performance of the model when applied to a real world system are identified. First, extending the model to account for the dynamic problem is considered. The SFFEVCHP models a static system. In practice, this means that the CSO must rely on the data available when planning starts to make handling and repositioning decisions. As demand, distribution of cars and traffic vary throughout the day, a model and algorithm considering these factors to re-solve the problem regularly with updated data throughout the planning period can improve the value of the model. E.g., the decision of where to move cars after handling could be re-solved every time an operator arrives at a rental car so that the car is repositioned based on the current distribution, demand and traffic. This could either be achieved by using real time data or expectations based on historic data.

Furthermore, the current model formulation require that every operator either is transported by a service vehicle, or handles a car. Allowing operators to be transported by cars being handled by other operators may reduce the need for service vehicles, hence reducing the costs of handling and repositioning. E.g., a service vehicle could drop off three operators by a rental car. One of the operators drive the rental car, drops off the two other operators at other rental cars and places the car at a charging station. This extension may yield significant gains in operational efficiency for the CSO. However, we expect that the routing problem will become substantially more complex as it must identify possible routes for rental cars including drop offs. In addition, it must take into account the trade off between rental cars travelling a longer distance to drop off operators and travel costs of service vehicles.

Another extension of the model is to include the possibility to perform repositioning of rental cars even though they are not in need of charging. In conjunction with the aforementioned extension to allow operators to be transported by rental cars handled by other operators, this may lead to better utilization of the cars in the system with a marginal cost increase. The current algorithm can be used in this manner by including more cars in the input data not necessarily in need of charging and adjust the postponement costs so that cars in need of charging have significantly higher cost than those that do not. This may include intelligent ways to select which cars to include in the data file to avoid getting overly large instances. Other approaches to this extension are also possible and implementing a pure repositioning element can lead to a higher utilized fleet.

Finally, the presented model and algorithm include a single depot. In real life systems however, multiple depots can be beneficial. There are mainly two factors contributing to this. First, the driving distance from the depot to the rental cars and charging stations can become long. Secondly, employees may have to travel a long distance to and from the depot wasting travel time that can be used on repositioning of rental cars. When traffic is taken into account, the importance of these effects increase. Thus, extending the model and algorithm to allow multiple depots can be of great value for companies. The MIP can easily be extended to allow multiple depots. The HGSADC however must be considerably modified to allow this. It is possible to develop a cluster first route second algorithm where rental cars are clustered to a depot and a SFFEVCHP is solved for each depot. The performance of such an algorithm is however highly dependent on the clustering scheme. As a result, incorporating multiple depots into the current algorithm is likely to perform better.

9.2.3 Improving the HGSADC for the SFFEVCHP

We believe that the improvement of some modules of the HGSADC as well as adding an intensification step may improve the solution quality and stability of the proposed algorithm. A possible area of improvement is the education module. The proposed education module attempts to improve the individual by rearranging the handling sequence, transport request assignment and route chromosomes. However, the rental car destination and operator chromosomes are not modified. Improvement of these chromosomes will likely contribute to finding high quality solutions faster. Particularly, we believe that attempting to insert postponed rental cars into existing operator sequences or adding more operators to handle postponed cars merits future research.

In addition to the education module, further development of the crossover module may improve the algorithm performance. The main contribution would be to employ a crossover operator capable of transferring more information about the route chromosome to the offsprings. Currently, the rental car destination, operator, and handling sequence chromosomes are inherited, but the DARP is solved for each offspring to form the transport request assignment and route chromosomes. We believe that including genetic material from the parent individuals for these chromosomes can further improve solution quality and stability.

Finally, we believe the notion of adding an intensification step to the algorithm deserves attention. As the dial-a-ride problem formed by the three first chromosomes is a complex problem to solve in itself, better solution methods of the DARP could be employed to further improve promising individuals. This can be done by employing a solution method capable of producing high quality solutions of the DARP in reasonable time. E.g., Masmoudi et al. (2017) present a genetic algorithm with operators designed for the DARP that shows promising results. Intensifying promising solutions, either at some interval during the HGSADC iterations or after the HGSADC terminates can have a positive effect on the solution quality. Nevertheless, as the DARP is complex to solve computation time is a significant concern and the trade off between the allowed running time of the HGSADC and the intensification module must be investigated.

Bibliography

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

Belk, R., 2014. You are what you can access: Sharing and collaborative consumption online. Journal of Business Research 67 (8), 1595–1600.

Boyaci, B., Zografos, K., Geroliminis, N., 2015. An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research 240 (3), 718–733.

Boyaci, B., Zografos, K. G., Geroliminis, N., 2017. An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. Transportation Research Part B: Methodological 95, 214–237.

Braekers, K., Caris, A., Janssens, G. K., sep 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.

Breedam, A. V., 2001. Comparing descent heuristics and metaheuristics for the vehicle routing problem. Computers & Operations Research 28 (4), 289–315.

Bruglieri, M., Pezzella, F., Pisacane, O., 2017. Heuristic algorithms for the operator-based relocation problem in one-way electric carsharing systems. Discrete Optimization 23, 56–80.

Cepolina, E. M., Farina, A., 2012. A new shared vehicle system for urban areas. Transportation Research Part C: Emerging Technologies 21 (1), 230–243.

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

Correia, G. H. d. A., Antunes, A. P., 2012. Optimization approach to depot location and trip selection in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review 48 (1), 233–247.

Correia, G. H. D. A., Jorge, D. R., Antunes, D. M., 2014. The Added Value of Accounting For Users' Flexibility and Information on the Potential of a Station-Based One-Way Car-Sharing System: An Application in Lisbon, Portugal. Journal of Intelligent Transportation Systems 18 (3), 299–308.

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

Fan, W., Machemehl, R., Lownes, N., 2008. Carsharing: Dynamic Decision-Making Problem for Vehicle Allocation. Transportation Research Record: Journal of the Transportation Research Board 2063, 97–104.

Folkestad, C. A., Hansen, N., 2016. Modeling the Static Free-Floating Carsharing Handling and Repositioning Problem. Tech. rep., Norwegian University of Science and Technology.

Frost and Sullivan, 2010. Analysis of the Market for Carsharing in North America. Tech. rep., Frost and Sullivan Research Service.

Frost and Sullivan, 2014. Strategic Insight of the Global Carsharing Market. Tech. rep., Frost and Sullivan Research Service.

Gschwind, T., Drexl, M., 2016. Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem.

Hamming, R. W., 1950. Error detecting and error correcting codes. Bell Labs Technical Journal 29 (2), 147–160.

Healy, P., Moll, R., may 1995. A new extension of local search applied to the Dial-A-Ride Problem. European Journal of Operational Research 83 (1), 83–104.

Herbawi, W., Knoll, M., Kaiser, M., Gruel, W., 2016. An evolutionary algorithm for the vehicle relocation problem in free floating carsharing. In: 2016 IEEE Congress on Evolutionary Computation, CEC 2016. pp. 2873–2879.

Jorge, D., Correia, G., Barnhart, C., 2014. Comparing Optimal Relocation Operations With Simulated Relocation Policies in One-Way Carsharing Systems. IEEE Transactions on Intelligent Transportation Systems 15 (4), 1667–1675.

Kaspi, M., Raviv, T., Tzur, M., 2014. Parking reservation policies in one-way vehicle sharing systems. Transportation Research Part B: Methodological 62, 35–50.

Kek, A., Cheu, R., Chor, M., 2006. Relocation Simulation Model for Multiple-Station Shared-Use Vehicle Systems. Transportation Research Record: Journal of the Transportation Research Board 1986, 81–88.

Kek, A. G. H., Cheu, R. L., Meng, Q., Fung, C. H., 2009. A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review 45 (1), 149–158.

Kirchler, D., Wolfler Calvo, R., oct 2013. A Granular Tabu Search algorithm for the Dial-a-Ride Problem. Transportation Research Part B: Methodological 56, 120–135.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2016. Thirty years of heterogeneous vehicle routing. European Journal of Operational Research 249 (1), 1–21.

Kuhne, K. S., Rickenberg, T. A., Breitner, M. H., 2016. An Optimization Model and a Decision Support System to Optimize Car Sharing Stations with Electric Vehicles. In: Lübbecke, M., Koster, A., Letmathe, P., Madlener, R., Peis, B., Walther, G. (Eds.), Operations Research Proceedings 2014: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), RWTH Aachen University, Germany, September 2-5, 2014. Springer International Publishing, Cham, pp. 313–320.

Li, X., Ma, J., Cui, J., Ghiasi, A., Zhou, F., 2016. Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. Transportation Research Part B: Methodological 88, 21–45.

Masmoudi, M. A., Braekers, K., Masmoudi, M., Dammak, A., 2017. A hybrid Genetic Algorithm for the Heterogeneous Dial-A-Ride Problem. Computers & Operations Research 81, 1–13.

Nair, R., Miller-Hooks, E., 2011. Fleet Management for Vehicle Sharing Operations. Transportation Science 45 (4), 524–540.

Nourinejad, M., Zhu, S., Bahrami, S., Roorda, M. J., 2015. Vehicle relocation and staff rebalancing in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review 81, 98–113.

Osaba, E., Onieva, E., Diaz, F., Carballedo, R., Lopez, P., Perallos, A., 2015. An Asymmetric Multiple Traveling Salesman Problem with Backhauls to solve a Dial-a-Ride problem.

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

Repoux, M., Boyaci, B., Geroliminis, N., 2015. Simulation and optimization of one-way car-sharing systems with variant relocation policies. In: Transportation Research Board 94th Annual Meeting.

Santos, G., Correia, G., 2015. A MIP Model to Optimize Real Time Maintenance and Relocation Operations in One-way Carsharing Systems. Transportation Research Procedia 10, 384–392.

Shaheen, S. A., Chan, N. D., Micheaux, H., 2015. One-way carsharing's evolution and operator perspectives from the Americas. Transportation 42 (3), 519–536.

Toth, P., Vigo, D., 1997. Heuristic Algorithms for the Handicapped Persons Transportation Problem. Transportation Science 31 (1), 60–71.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. Operations Research 60 (3), 611–624.

Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2014. A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research 234 (3), 658–673.

Weikl, S., Bogenberger, K., 2013. Relocation Strategies and Algorithms for Free-Floating Car Sharing Systems. IEEE Intelligent Transportation Systems Magazine 5 (4), 100–111.

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

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

Appendix A

Final Model Formulation

Notation

Sets	
$\mathcal{N}$	Set of all nodes
$\mathcal{N}^{CS}$	Set of all charging stations, $\mathcal{N}^{CS} \subset \mathcal{N}$
$\mathcal{N}^{EV}$	Set of rental cars in need of handling, $\mathcal{N}^{EV} \subset \mathcal{N}$
$\mathcal{M}_i$	Set of all possible visits to node i
$\mathcal{V}$	Set of all service vehicles
$\mathcal{D}$	Set of all operators
Indices	
i, j, k	Node *i, j, k* $\in \mathcal{N}$
a, b, c	Operator visit *a,b,c* to node *i*, $a, b, c \in \mathcal{M}_i$
m, n, o	Service vehicle visit *m,n,o* in node *i*, $m, n, o \in \mathcal{M}_i$
v	Service vehicle *v* $\in \mathcal{V}$
d	Operator *d* $\in \mathcal{D}$
Parameters	
N_j^{CSP}	Number of available charging slots at charging station *j*
C_j^E	Deviation cost in charging station j
C_{ij}^T	Travel cost between node i and j
C_i^{PH}	Cost of postponed handling of rental car in node i
C^V	Fixed service vehicle cost
C^D	Fixed operator cost
T_{ij}	Travel time between node i and j
T_i^{EV}	Max travel time for rental car in node i
$\overline{T}$	Time limit for the planning period
Q	Service vehicle capacity
S_j^0	Initial state at charging station j
S_j^I	Ideal state at charging station j

Variables

x_{imjnv}	1 if service vehicle *v* drives directly from visit (*i,m*) to visit (*j,n*), 0 otherwise
$f_{imajnbvd}$	1 if opertor *d* is transported from visit (*i,a*) to (*j,b*) by service vehicle *v* in visit (*i,m*) to (*j,n*), 0 otherwise
q_{ivd}	1 if operator *d* is dropped off in *i* by service vehicle *v*, 0 otherwise
g_{jnbvd}	1 if operator *d* is picked up in visit (*j,b*) by service vehicle *v* in visit (*j,n*), 0 otherwise
h_{ijbd}	1 if operator *d* handles rental car *i* to charging station visit (*j, b*), 0 otherwise
t^V_{imv}	Time of arrival to visit (*i,m*) for service vehicle *v*
t^D_{iad}	Time of arrival to visit (*i,a*) for operator *d*
z^H_i	1 if the handling of rental car *i* is postponed, 0 otherwise
y_j	Deviation from ideal state in node *j*
s_v	1 if service vehicle *v* is used, 0 otherwise
w_d	1 if operator *d* is used, 0 otherwise

Objective Function

$$\begin{aligned} min \sum_{j\in\mathcal{N}^{CS}} C^E_j\, y_j + \sum_{i\in\mathcal{N}} \sum_{m\in\mathcal{M}_i} \sum_{j\in\mathcal{N}} \sum_{n\in\mathcal{M}_j} \sum_{v\in\mathcal{V}} C^T_{ij}\, x_{imjnv} + \\ \sum_{i\in\mathcal{N}^{EV}} C^{PH}_i\, z^H_i + \sum_{v\in\mathcal{V}} C^V\, s_v + \sum_{d\in\mathcal{D}} C^D\, w_d \end{aligned} \tag{A.1}$$

Constraints

$$\sum_{j\in\mathcal{N}\setminus\{0\}} x_{01j1v} = s_v \qquad v \in \mathcal{V} \tag{A.2}$$

$$\sum_{j\in\mathcal{N}\setminus\{0\}} \sum_{m\in\mathcal{M}_j} x_{jm02v} = s_v \qquad v \in \mathcal{V} \tag{A.3}$$

$$\sum_{j\in\mathcal{N}\setminus\{0\}} \sum_{n\in\mathcal{M}_j} x_{imjnv} \leq s_v \qquad i \in N\setminus\{0\}, m \in \mathcal{M}_i, \; v \in \mathcal{V} \tag{A.4}$$

$$\sum_{i\in\mathcal{N}} \sum_{m\in\mathcal{M}_i} x_{imjnv} = \sum_{i\in\mathcal{N}} \sum_{m\in\mathcal{M}_i} x_{jnimv} \qquad j \in \mathcal{N}\setminus\{0\}, n \in \mathcal{M}_j, \; v \in \mathcal{V} \tag{A.5}$$

$$\sum_{i\in\mathcal{N}^{EV}} \sum_{b\in\mathcal{M}_j} \sum_{d\in\mathcal{D}} h_{ijbd} \leq N^{CSP}_j \qquad j \in \mathcal{N}^{CS} \tag{A.6}$$

$$\sum_{j\in\mathcal{N}^{CS}} \sum_{b\in\mathcal{M}_j} \sum_{d\in\mathcal{D}} h_{ijbd} + z^H_i = 1 \qquad i \in \mathcal{N}^{EV} \tag{A.7}$$

$$\sum_{b\in\mathcal{M}_j}\sum_{d\in\mathcal{D}} T_{ij}\, h_{ijbd} \le T_i^{EV} \qquad i\in\mathcal{N}^{EV}, j\in\mathcal{N}^{CS} \qquad \text{(A.8)}$$

$$\sum_{j\in\mathcal{N}^{CS}}\sum_{b\in\mathcal{M}_j} h_{ijbd} = \sum_{v\in V} q_{ivd} \qquad i\in\mathcal{N}^{EV}, d\in\mathcal{D} \qquad \text{(A.9)}$$

$$\sum_{i\in\mathcal{N}^{EV}} h_{ijbd} = \sum_{v\in\mathcal{V}}\sum_{n\in\mathcal{M}_j} g_{jnbvd} \qquad j\in\mathcal{N}^{CS}, b\in\mathcal{M}_j, d\in\mathcal{D} \qquad \text{(A.10)}$$

$$\sum_{i\in\mathcal{N}^{EV}} h_{ijbd} + \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{a\in\mathcal{M}_i}\sum_{n\in\mathcal{M}_j}\sum_{v\in\mathcal{V}} f_{imajnbvd} \le w_d \qquad j\in\mathcal{N}^{CS}, b\in\mathcal{M}_i, d\in\mathcal{D} \qquad \text{(A.11)}$$

$$\sum_{j\in\mathcal{N}\setminus\{0\}}\sum_{v\in V} f_{011j11vd} = w_d \qquad d\in\mathcal{D} \qquad \text{(A.12)}$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{a\in\mathcal{M}_i}\sum_{v\in\mathcal{V}} f_{ima022vd} = w_d \qquad d\in\mathcal{D} \qquad \text{(A.13)}$$

$$\sum_{k\in N}\sum_{o\in\mathcal{M}_k}\sum_{c\in\mathcal{M}_k} f_{jnbkocvd} = \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{a\in\mathcal{M}_i} f_{imajnbvd} + g_{jnbvd} - q_{jvd} \qquad j\in\mathcal{N}\setminus\{0\}, n\in\mathcal{M}_j, b\in\mathcal{M}_j, v\in\mathcal{V}, d\in\mathcal{D} \qquad \text{(A.14)}$$

$$\sum_{a\in\mathcal{M}_i}\sum_{b\in\mathcal{M}_j}\sum_{d\in\mathcal{D}} f_{imajnbvd} \le Q\, x_{imjnv} \qquad i\in\mathcal{N}, m\in\mathcal{M}_i, j\in\mathcal{N}, n\in\mathcal{M}_j, v\in\mathcal{V} \qquad \text{(A.15)}$$

$$t^V_{imv} \le \overline{T}\, s_v \qquad i\in\mathcal{N}, m\in\mathcal{M}_i, v\in\mathcal{V} \qquad \text{(A.16)}$$

$$t^V_{imv} + (T_{ij} + M_1)\, x_{imjnv} \le t^V_{jnv} + M_1 \qquad i\in\mathcal{N}, m\in\mathcal{M}_i, j\in\mathcal{N}, n\in\mathcal{M}_j, v\in\mathcal{V} \qquad \text{(A.17)}$$

$$t^D_{jbd} + M_2\, g_{jnbvd} \le t^V_{jnv} + M_2 \qquad j\in\mathcal{N}^{CS}, n\in\mathcal{M}_j, b\in\mathcal{M}_j, v\in\mathcal{V}, d\in\mathcal{D} \qquad \text{(A.18)}$$

$$t^D_{iad} \le \overline{T}\, w_d \qquad i\in\mathcal{N}, a\in\mathcal{M}_i, d\in\mathcal{D} \qquad \text{(A.19)}$$

$$t^D_{iad} + (T_{ij} + M_3)\, h_{ijbd} \le t^D_{jbd} + M_3 \qquad i\in\mathcal{N}^{EV}, a\in\mathcal{M}_i, j\in\mathcal{N}^{CS}, b\in\mathcal{M}_j, d\in\mathcal{D} \qquad \text{(A.20)}$$

$$t^V_{imv} + (T_{ij} + M_4)\, f_{imajnbvd} \leq t^D_{jbd} + M_4 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.21}$$

$$y_j \geq S^0_j + \sum_{i \in \mathcal{N}^{EV}} \sum_{b \in \mathcal{M}_j} \sum_{d \in \mathcal{D}} h_{ijbd} - S^I_j \qquad j \in \mathcal{N}^{CS} \tag{A.22}$$

$$y_j \geq -S^0_j - \sum_{i \in \mathcal{N}^{EV}} \sum_{b \in \mathcal{M}_j} \sum_{d \in \mathcal{D}} h_{ijbd} + S^I_j \qquad j \in \mathcal{N}^{CS} \tag{A.23}$$

$$s_{(v+1)} - s_v \leq 0 \qquad v \in \mathcal{V} \backslash \{|\mathcal{V}|\} \tag{A.24}$$

$$w_{(d+1)} - w_d \leq 0 \qquad d \in \mathcal{D} \backslash \{|\mathcal{D}|\} \tag{A.25}$$

$$h_{ijbd} \leq w_d \qquad i \in \mathcal{N}^{EV}, j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j, d \in \mathcal{D} \tag{A.26}$$

$$x_{imjnv} \leq s_v \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{A.27}$$

$$f_{imajnbvd} \leq s_v \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.28}$$

$$g_{jnbvd} \leq s_v \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.29}$$

$$q_{ivd} \leq s_v \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.30}$$

$$f_{imajnbvd} \leq w_d \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.31}$$

$$g_{jnbvd} \leq w_d \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.32}$$

$$q_{ivd} \leq w_d \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.33}$$

Non-negativity, Integer, and Binary Restrictions

$$x_{imjnv} \in \{0,1\} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{D} \tag{A.34}$$

$$f_{imajnbvd} \in \{0,1\} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, a \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.35}$$

$$q_{ivd} \in \{0,1\} \qquad i \in \mathcal{N}^{EV}, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.36}$$

$$g_{jnbvd} \in \{0,1\} \qquad j \in \mathcal{N}^{CS}, n \in \mathcal{M}_j, b \in \mathcal{M}_j, v \in \mathcal{V}, d \in \mathcal{D} \tag{A.37}$$

$$h_{ijbd} \in \{0,1\} \qquad i \in \mathcal{N}^{EV}, j \in \mathcal{N}^{CS}, b \in \mathcal{M}_j, d \in \mathcal{D} \tag{A.38}$$

$$t^V_{imv} \geq 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, v \in \mathcal{V} \tag{A.39}$$

$$t^D_{iad} \geq 0 \qquad i \in \mathcal{N}, a \in \mathcal{M}_i, d \in \mathcal{D} \tag{A.40}$$

$$z^H_i \in \{0,1\} \qquad i \in \mathcal{N}^{EV} \tag{A.41}$$

$$y_j \in \mathbb{Z}^+ \qquad j \in \mathcal{N}^{CS} \tag{A.42}$$

$$s_v \in \{0,1\} \qquad v \in \mathcal{V} \tag{A.43}$$

$$w_d \in \{0,1\} \qquad d \in \mathcal{D} \tag{A.44}$$

Appendix B

Base Configuration for Parameter Calibration of the HGSADC

Table B.1: Overview of the base parameters used in parameter calibration of the HGSADC

Parameter	Value	Description
μ	25	Minimum population size
λ	75	Generation size
I^{NI}	10,000	Max. number of iterations without improvement
η^{DIV}	0.20	Proportion of I^{NI} , such that $I^{DIV} = \eta^{DIV} \times I^{NI}$
η^{ELI}	0.75	Proportion of elite individuals such that $n^{ELI} = \eta^{ELI} \times \lvert \mathcal{S} \rvert$
η^{CLO}	0.2	Proportion of individuals considered in diversity contribution, such that $n^{CLO} = \eta^{CLO} \times \mu$
K^{INIT}	4	Construction heuristic size factor
K^{DIV}	4	Diversification size factor
$\rho^{EDU}_{construct}$	0.5	Probability of education in construction heuristic
$\rho^{REP}_{construct}$	0.5	Probability of repair in construction heuristic
$\rho^{EDU}_{crossover}$	0.5	Probability of education in crossover
$\rho^{REP}_{crossover}$	0.5	Probability of repair in crossover
ζ^{REF}	0.6	Desired ratio of feasible individuals
w^{T}	2.0	Duration violation penalty
w^{V}	2.0	Number of vehicles violation penalty
ξ^{UP}	1.25	Penalty adjustment factor, up
ξ^{DOWN}	0.75	Penalty adjustment factor, down
T^{MAXRUN}	3,600	Maximum running time (seconds)