RECHARGE STRATEGIES FOR THE ELECTRIC VEHICLE ROUTING PROBLEM WITH TIME WINDOWS IN DETERMINISTIC AND STOCHASTIC ENVIRONMENTS

by

Merve Keskin Özel

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Sabancı University,

December, 2018

RECHARGE STRATEGIES FOR THE ELECTRIC VEHICLE ROUTING PROBLEM WITH TIME WINDOWS IN DETERMINISTIC AND STOCHASTIC ENVIRONMENTS

APPROVED BY:

Prof. Dr. Bülent Çatay
(Thesis Supervisor)

Prof. Dr. Tonguç Ünlüyurt

Assoc. Prof. Dr. Deniz Aksen

Asst. Prof. Dr. Duygu Taş Küten

Assoc. Prof. Dr. Güvenç Şahin

DATE OF APPROVAL: 11/12/2018

RECHARGE STRATEGIES FOR THE ELECTRIC VEHICLE ROUTING PROBLEM WITH TIME WINDOWS IN DETERMINISTIC AND STOCHASTIC ENVIRONMENTS

Merve Keskin Özel

PhD Thesis, 2018

Thesis Advisor: Prof. Dr. Bülent Çatay

Keywords: electric vehicle routing, metaheuristics, green logistics

Due to increasing concerns about greenhouse gas emissions in recent years, many companies have had an interest in using alternative fuel vehicles in their fleets. Electric vehicles (EVs) are one of these vehicles and they have various advantages such as zero tailpipe emissions, low maintenance costs and low energy consumption. However, their acquisition costs are higher compared to the conventional vehicles and recharging the battery may take significant amount of time compared to the short fueling times. Hence, to overcome these challenges, logistics decisions have to be made effectively. The problem of planning EVs' activities has been introduced to the literature as the Electric Vehicle Routing Problem (EVRP), which is a special case of the classical VRP where the fleet consists of EVs. The difference between this problem and the classical VRP is that vehicles have batteries as the energy source and the battery is being discharged while the EV is traveling. Hence, the EVs may recharge their batteries at the recharging stations to continue their routes. These stations are located at distant locations and there are few of them compared to the common fuel stations. Recharging may be performed at any level of the battery and the recharging time increases with the recharge amount. In some stations, there may be different chargers which vary in terms of charging speed. For instance, fast chargers recharge the battery faster, but they incur higher cost. Furthermore, EVs may wait in the queue at the stations since there may be other EVs which arrive earlier and wait for

service. In this dissertation, we address four problems which consider these different features of the EVRP. First, we study the EVRP with Time Windows where the batteries can be recharged partially at the recharging stations. Second, we extend this problem where the recharging stations are equipped with multiple types of chargers which differ by recharging rates and unit recharging costs. Next, we consider a stochastic environment where an EV may wait in the queue before recharging due to other EVs that have arrived earlier at that station. The waiting times depend on the time of the visit during the day, i.e., they are longer in the rush hours. Furthermore, the recharging time is assumed to be a nonlinear function of the energy recharged. In the final problem, we consider random waiting times at the recharging stations. In this case, the EVs do not have information about the queue lengths of the stations before they arrive at. We propose Adaptive Large Neighborhood Search heuristics and matheuristics to solve these problems effectively.

ZAMAN PENCERELİ ELEKTRİKLİ ARAÇ ROTALAMA PROBLEMİ İÇİN DETERMİNİSTİK VE RASSAL ORTAMLARDA ŞARJ STRATEJİLERİ

Merve Keskin

Doktora Tezi, 2018

Tez Danışmanı: Prof. Dr. Bülent Çatay

Anahtar Kelimeler: elektrikli araç rotalama, metasezgisel, yeşil lojistik

Sera gazı salınımları ile ilgili son yıllarda artan endişeler nedeniyle birçok şirket, filosuna alternatif yakıtlar ile çalışan araçları dahil etmekle ilgilenmeye başlamıştır. Elektrikli araçlar (EA) da bu araçlardan biri olup, egzoz gazı salınımı olmaması, düşük bakım ve enerji maliyetleri gibi birçok avantaja sahiptir. Bunların yanında, satınalma maliyeti klasik araçlara göre yüksek olup, pilin şarj edilmesi, kısa yakıt doldurma süresine kıyasla oldukça uzun sürebilmektedir. Bu zorlukları aşabilmek için lojistik kararların etkin bir şekilde alınması gerekmektedir. Elektrikli araçların hareketlerinin planlanmasını içeren bu problem literatüre Elektrikli Araç Rotalama Problemi (EARP) olarak girmiştir ve özünde, filonun EA'lardan oluştuğu, klasik ARP'nin özel bir durumudur. Bu problem ile klasik ARP'nin farkı, araçların enerji kaynağı olarak, araç yolda ilerledikçe şarj seviyesi azalan bir pile sahip olmalarıdır. Bu nedenle, araçlar rotalarına devam edebilmek için şarj istasyonlarına uğrayıp pillerini şarj etmek zorunda kalabilirler. Bu istasyonlar uzak mesafelerde olup sayıları, yaygın olarak bulunan benzin istasyonlarına göre oldukça azdır. Şarj işlemi, pil herhangi bir şarj seviyesindeyken yapılabilmekte ve şarj süresi, şarj miktarına bağlı olarak artmaktadır. Bazı istasyonlar şarj hızı farklı olan şarj cihazlarına da sahip olabilir. Örneğin, hızlı şarj cihazları pili hızlı şarj ederken, birim şarj maliyetleri daha yüksektir. Bununla beraber, istasyonda daha erken gelmiş olan araçlar olması durumunda, yeni gelen bir EA şarj işleminden önce bir süre kuyrukta beklemek

zorunda kalabilir. Bu tezde, EARP'nin farklı özelliklerini ele alan dört problem incelenmiştir. İlk olarak şarj istasyonlarında, araçların pillerinin kısmi olarak şarj edildiği EARP ele alınmıştır. İkinci olarak şarj istasyonlarının, şarj hızı ve birim şarj maliyetleri farklı olan şarj ekipmanlarına sahip olduğu problem incelenmiştir. Üçüncü problem rassal bir ortamı incelemekte olup, bir EA'nın şarj işleminden önce, o istasyona daha erken gelmiş olan başka araçlar nedeniyle bir süre kuyrukta bekleyebildiği durumu ele almaktadır. Bu bekleme süreleri, gün içinde istasyonun ziyaret saatine göre değişkenlik göstermektedir. Örneğin, trafiğin yoğun olduğu saatlerde bekleme süresi daha uzundur. Ayrıca, şarj süresinin şarj miktarının doğrusal olmayan bir fonksiyonu olduğu varsayılmıştır. Son problem ise, şarj istasyonlarında rassal bekleme sürelerini ele almaktadır. Bu durumda araçların, ilgili istasyona gitmedikleri sürece oradaki bekleme süresi hakkında bilgileri yoktur. Bu problemlerin etkin bir şekilde çözümü için Uyarlanabilir Geniş Komşuluk Arama Yöntemi sezgisel ve mat-sezgisel yöntemleri geliştirilmiştir.

to my mother

Acknowledgments

First of all, I would like to express my deepest gratitude to my advisor Prof. Bülent Çatay for his guidance, support and encouragement at all times. I have learnt many things from him and being his student has been an honor and pleasure. Without his help, I would not have even begun this PhD study.

I am much grateful to Professors Tonguç Ünlüyurt, Nilay Noyan, Kerem Bülbül, Güvenç Şahin, Barış Balcıoğlu, İlker Birbil, Kemal Kılıç and Gündüz Ulusoy for all they taught me. Being their student and learning from them have been a privilege for me.

I would like to thank my thesis committee members Prof. Tonguç Ünlüyurt and Prof. Deniz Aksen for their valuable advises and insightful comments throughout my study. They have always supported me with their positive energy.

It was a pleasure to be a visiting student at CIRRELT. I would like to thank Prof. Gilbert Laporte for his hospitality. The third part of this study was improved with his guidance, advices and feedbacks.

I would like to thank Prof. Duygu Taş for her contributions. Without her ideas and advises, I would not have developed the last part of this study.

Many thanks to my friends Siamak Naderi Varandi, Sina Rastani and Amin Ahmadi for always supporting me. We have shared not only the Logistics Lab but also many joyful moments together.

I would like to thank my friends Çağrı Koç and Şeyma Koç for their support in my Montréal visit. We had great days together and they made me feel that I am at home.

Many thanks to Sinem Aydın and Banu Akıncı for their help in solving bureaucratic problems. And thanks to Osman Rahmi Fıçıcı, the problems with the computers I have been using were solved instantly.

I would like to express my special thanks to Berk Özel who has always shared all the positive and negative experiences with me. He has been much understanding to me at all times and he has always encouraged and guided me to do the right.

Last but not least, I would like to express my appreciation to my family. I have always felt their support. Without their help, I would not have finished this study.

I would like to thank Scientific and Technological Research Council of Turkey TÜBİTAK for supporting me to participate in a conference within this study.

Contents

List of Figures ... xi
List of Tables ... xii
List of Algorithms ... xiv
1. Introduction ... 1
1.1. Overview of Electric Vehicle Technology ... 2
1.2. The Problem of Routing Electric Vehicles ... 4
1.3. Thesis Organization ... 4
2. Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows
2.1. Introduction ... 8
2.2. Related Literature ... 9
2.3. Problem Description and Model Formulation ... 12
2.4. Solution Methodology ... 15
2.4.1. Overview of the Proposed ALNS Approach ... 15
2.4.1.1. Initial Solution Construction ... 15
2.4.1.2. ALNS Procedure ... 16
2.4.2. Removal Algorithms ... 17
2.4.2.1. Customer Removal ... 17
2.4.2.2. Station Removal ... 19
2.4.3. Insertion Algorithms ... 21
2.4.3.1. Customer Insertion ... 21
2.4.3.2. Station Insertion ... 22
2.5. Computational Study ... 24
2.5.1. Parameter Tuning ... 25
2.5.2. Numerical Results for EVRPTW Instances ... 26
2.5.3. Experiments on EVRPTW Instances Using PR Scheme ... 28
2.5.3.1. Numerical Results for Small-Size Instances ... 28
2.5.3.2. Numerical Results for Large-Size Instances ... 30
2.6. Conclusions and Future Research ... 32

3. A Matheuristic Method for the Electric Vehicle Routing Problem with Time Windows and Fast Chargers

3.1. Introduction 34
3.2. Related Literature 36
3.3. Problem Description and Model Formulation 37
3.3.1. Problem Description 37
3.3.2. Problem Formulation 40
3.3.2.1. Model 1 41
3.3.2.2. Model 2 43
3.3.2.3. Evaluation of the Two Models 44
3.4. Description of the Matheuristic 44
3.4.1. Removal Heuristics 45
3.4.1.1. Customer Removal 45
3.4.1.2. Station Removal 46
3.4.2. Insertion Heuristics 46
3.4.2.1. Customer Insertion 46
3.4.2.2. Station Insertion 47
3.4.3. Constructing the Initial Solution 47
3.4.4. Route Enhancement 47
3.4.5. Reducing the Number of Vehicles 50
3.5. Experimental Design and Numerical Results 51
3.5.1. Results for Large Instances 51
3.5.1.1. Analysis of Different Configurations 51
3.5.1.2. Effect of Multiple Chargers 55
3.5.2. Results for Small Instances 57
3.5.3. Results for FORT instances of Felipe et al. (2014) 58
3.6. Conclusions 60

4. Electric Vehicle Routing Problem with Soft Time Windows and Time-Dependent Waiting Times at Recharging Stations

4.1. Introduction 62
4.2. Related Literature 63
4.3. Problem Description and Formulation 65
4.3.1. Time-Dependent Waiting Time Functions 65
4.3.2. Mathematical Formulation 69
4.4. Solution Methodology 74

4.4.1. Adaptive Large Neighborhood Search 75
4.4.1.1. Initial Solution 75
4.4.1.2. Customer Removal 75
4.4.1.3. Customer Insertion 76
4.4.1.4. Station Insertion... 76
4.4.2. Fixed Sequence Route Optimization... 77
4.4.2.1. Mathematical Model 77
4.4.2.2. Preprocessing on Decision Variables for the Fixed Route Formulation 80
4.5. Computational Experiments 81
4.5.1. Experimental Design.. 81
4.5.2. Results on Small Instances... 84
4.5.3. Results on Large Instances... 84
4.5.3.1. The Impact of Waiting on Total Cost and Its Components....... 84
4.5.3.2. The Impact of Waiting at the Recharging Stations on the Decisions Made in Different Time Intervals 88
4.5.3.3. Sensitivity of the Solutions to the Late Arrival Penalty 89
4.5.3.4. Computational Times 91
4.5.3.5. Computations on R2 Type Instances 92
4.6. Conclusions 92
5. Electric Vehicle Routing Problem with Time Windows and Stochastic Waiting Times at Recharging Stations
5.1. Introduction 94
5.2. Related Literature 95
5.3. Problem Description and Formulation 97
5.3.1. Formulation of the First Stage Problem 98
5.3.2. Recourse Action 100
5.3.3. Formulation of the Second Stage Problem 103
5.3.4. Modeling the Waiting Times 105
5.4. Solution Methodology 106
5.4.1. Stochastic Simulation for Computing the Expected Values and Probabilities 106
5.4.2. Destroy Operators 106
5.4.2.1. Customer Removal Operators 106
5.4.2.2. Station Removal Operators.. 107

5.4.3. Repair Operators ... 107
5.4.3.1. Customer Insertion Operators ... 107
5.4.3.2. Station Insertion Operators ... 109
5.4.4. Initial Solution ... 109
5.4.5. Solving the Second Stage Problem ... 110
5.4.6. Waiting Time Adjustment ... 110
5.5. Computational Study ... 111
5.5.1. Problem Settings ... 112
5.5.2. Parameter Tuning ... 113
5.5.3. Results ... 114
5.5.3.1. Sensitivity of Results to Utilization Levels at the Stations ... 114
5.5.3.2. Effect of Parameter N_w ... 116
5.6. Conclusions ... 118
6. Conclusions ... 120
Appendix A ... 123
Appendix B ... 125
Appendix C ... 126
Bibliography ... 128

List of Figures

2.1. An illustrative example 12
2.2. Illustration of Greedy Route Removal 18
2.3. Example of time-window infeasibility after SR 20
2.4. Example of battery infeasibility after SR 21
2.5. An improved route after SR and SI procedure 23
3.1. Route plans when each recharging station is equipped with (a) only normal chargers, (b) normal, fast and super-fast chargers 39
3.2. Set of recharging stations between customers i and $i+1$ 49
3.3. Station insertion between two nodes 50
3.4. Percentage of computational effort required by ALNS vs. CPLEX 54
4.1. Arrival rate of EVs at station i as a function of time 68
4.2. Piecewise linear approximation of the arrival rate as a function of time 68
4.3. Piecewise linear approximation of waiting time in the queue as a function of time ... 69
4.4. Piecewise linear approximation for the charging function 70
4.5. Elimination of dominated stations 77
4.6. Average waiting times for each scenario 83
4.7. Average number of recharges for each data set 86
4.8. Comparison of different cost components for different waiting schemes 87
4.9. Distribution of the cost components for no-wait and TD-St-L scenarios 88
4.10. Temporal analysis of recharging decisions and costs 90
5.1. Illustration of the recourse action and the resulting solution after the recourse 101
5.2. A route in which the EV arrives at the depot later than its due date 102
5.3. First stage solution and the second stage solutions of two different scenarios 102

List of Tables

2.1. Average results for EVRPTW obtained with fixed parameters 26
2.2. Comparison with the best-known solutions of EVRPTW instances 27
2.3. Comparison of results obtained with CPLEX and ALNS on the small-size instances 29
2.4. EVRPTW-PR results for different recharge strategies 31
3.1. Demand and time-window data for the example illustrated in Figure 3.1.................. 38
3.2. Mathematical notation 40
3.3. Comparison of the two models 44
3.4. Comparison of results obtained with different configurations 52
3.5. Average run times of different configurations (in minutes) 53
3.6. Comparison of results obtained by multiple fast charging vs. single normal charging 56
3.7. Comparison of results on small size instances 58
3.8. Comparison of average results with Felipe et al. (2014) on FORT instances 59
4.1. Meanings of the objective functions in Table 4.2.. 64
4.2. EVRP literature review 66
4.3. Mathematical notation 71
4.4. Mathematical notation 77
4.5. Average waiting time ($\overline{W}$) parameters for each scenario... 83
4.6. Results on small size instances 85
4.7. Impact of different waiting schemes on total cost 86
4.8. Comparison of results for different late arrival penalty values 90
4.9. Computation times for various waiting schemes (in minutes) 91
4.10. Sensitivity of solution quality and run time to number of iterations 91
4.11. Comparison of results for R1 and R2 instances... 92
5.1. Notation for the first-stage problem... 98
5.2. Notation for the second-stage problem 103
5.3. Parameter tuning 114
5.4. Comparison of best deterministic and stochastic solutions 115
5.5. Results for low utilization levels at stations 116
5.6. Results for high utilization levels at stations 117
5.7. The results with and without adjusting waiting times 118

A.1. Notation and description of the parameters 123
A.2. Parameter tuning 124
C.1. Comparison of results on 100-customer instances of Felipe et al. (2014) 126
C.2. Comparison of results on 200-customer instances of Felipe et al. (2014) 127
C.3. Comparison of results on 400-customer instances of Felipe et al. (2014) 127

List of Algorithms

2.1. Initial solution construction 16

2.2. ALNS algorithm... 24

4.1. General Framework of the Matheuristic 80

5.1. Simulation to calculate the probability that the next customer is infeasible................ 107

5.2. Simulation to calculate the probability that the route is feasible 108

5.3. Simulation to calculate expected recourse cost and construct second stage solution.. 109

5.4. General structure of the proposed metaheuristic 112

B.1. Pseudocode of the proposed matheuristic 125

Chapter 1

Introduction

Transportation systems account for about 20-25% of global energy consumption and CO_2 emissions. Road transport is a major contributor with 75% share (White Paper on Transport, 2011). 95% of the world's transportation energy comes from fossil fuels, mainly gasoline and diesel. In the US, about 28% of total greenhouse gas (GHG) emissions in 2016 are transport related (www.epa.gov). 74% of the domestic freight in 2012 is moved by trucks and the freight volume is expected to grow by 39% in 2040 (Bureau of Transportation Statistics, 2014). Transport accounts for 63% of fuel consumption and 29% of all CO_2 emissions in the EU. Freight transport activity is predicted to grow by around 80% in 2050 compared to 2005 (ec.europa.eu).

Transportation will continue to be a major and still growing source of GHGs. Hence, governments are considering new environmental measures and targets for reducing emissions and fuel resource consumptions. The US Administration aims at cutting the overall GHG emissions 17% below 2005 levels by 2020 and has recently established the toughest fuel economy standards for internal combustion engine vehicles (ICEVs) in the US history (www.state.gov). The EU targets 80–95% reduction of GHGs below 1990 levels by 2050, where a reduction of at least 60% is expected from the transport sector. The European Commission aims at reducing the transport-related GHG emissions to around 20% below their 2008 level by 2030. The use of conventional vehicles will be reduced by 50% in urban transport by 2030 and phased out by 2050. City logistics in major European urban centers will be CO_2-free by 2030 (White Paper on Transport, 2011).

The targets set by governments and the new regulations imposed encourage the usage of alternative fuel vehicles (AFV) such as solar, electric, biodiesel, LNG, CNG vehicles. Many municipalities, government agencies, non-profit organizations, and private companies are converting their fleets to include AFVs, either to reduce their environmental impact voluntarily or to meet new environmental regulations (Erdoğan and Miller-Hooks, 2012). Consequently, the improvements in the EV technology have gained momentum in parallel with the growing environmental concerns in societies.

This thesis aims to develop models and solution algorithms for different vehicle routing problems in which EVs are used.

1.1. Overview of Electric Vehicle Technology

EVs move with electric propulsion and can provide emission-free urban transportation. They can be classified as battery electric vehicles (BEV), hybrid electric vehicles (HEV), and fuel-cell electric vehicles (FCEV) such as electric trains, airplanes, boats, motorcycles, scooters, and spacecrafts (Chan, 2002). Within the routing context, we refer to EV as a commercial road vehicle such as a lorry or van. A BEV has only one or more electric motors and uses the power generated by the on-board battery for propulsion (Electrification Coalition, 2013). As reported in Pollet et al. (2012), the advantages of BEVs are lack of tailpipe emissions, high efficiency and low operating noise while they have some disadvantages such as low achievable driving range and low energy density causing long times for recharging the battery. Number of moving parts in BEVs are much less than of ICEVs and do not require regular oil changes (Feng and Figliozzi, 2013). Also due to the regenerative breaking, brake wear is used less which brings less maintenance costs (Lee et al., 2013). Nesterova et al. (2013) stated that a single charge for freight BEVs provides a range varying from 100 to 150 kilometers.

HEVs are further classified according to their powertrain architecture as parallel, series, series-parallel and complex (Chan, 2007). A plug-in hybrid electric vehicle (PHEV) is an HEV which utilizes a rechargeable battery and is also equipped with both electric motor and internal combustion engine (ICE). In series type of vehicles, internal combustion engine (ICE) is used to power a generator and the propulsion comes from the electric motor while in the parallel type, both the ICE and the electric motor are used in the propulsion (Chan, 2007). The main advantage of PHEVs is the ability to move using fuel when it runs out of battery power.

In the FCEV, the electricity is produced by a fuel cell via a chemical reaction which uses hydrogen as the input and produces water as the output. Then, the electricity is used to charge a battery or power the electric motor (Chan, 2007). den Boer et al. (2013) reported that fuel cells can convert approximately 50% of hydrogen's energy to electricity and they have a durability of 10,000 operating hours. Those factors are the main drawbacks of FCEVs.

The rechargeable battery is the critical component of EVs. The main types of batteries include lead acid batteries, nickel metal hybrid batteries and lithium-ion batteries (Chan, 2007). Lithium-ion batteries are the most widely used type since they have high energy density, high power density, long battery life and low memory effect compared to other alternatives (den Boer et al., 2013).

There are different ways for recharging EVs such as conductive charging, inductive charging and battery swapping. The most common method is conductive charging which is done by a cable and a vehicle connector. In inductive charging, the power is transferred to the battery magnetically via an on-board charger without needing any cables connectors (Yılmaz and Krein, 2013). Stationary inductive charging is used when the vehicle is stopped while in-road inductive charging can be performed even if the vehicle is moving (den Boer et al., 2013). Battery swapping includes changing the empty battery with a fully charged one in a battery swapping station. Using catenary wires is another charging option where the vehicles can be recharged using a pantograph device which slides along the electric wires and transfer the energy. It can be useful for public electric buses (CALSTART, 2013).

The battery recharging times are dependent to the battery type, charging equipment and charging level. Yılmaz and Krein (2013) classifies the charging levels into three categories: level 1 (1.4 kW to 1.9 kW), level 2 (4 kW to 19.2 kW) and level 3 (50 kW to 100 kW). The last is also called as fast/quick charging. The charge durations are linear with respect to time at the first phase of charging which corresponds to almost full battery while the second phase is non-linear and can take hours to obtain a fully charged battery (Bruglieri et al., 2015).

Although EVs enable low-emission logistics services, operating an EV fleet has several drawbacks such as: (i) low energy density of batteries compared to the fuel of combustion engine vehicles; (ii) limited number of public charging stations; and (iii) long recharging times (Touati-Moungla and Jost, 2011). Battery swap may remedy the last; however, swapping raises additional issues in battery design and compatibility, battery degradation, ownership, and swap station infrastructure. Under these limitations, routing an EV fleet arises as a challenging combinatorial optimization problem in the Vehicle Routing Problem (VRP) literature. The problems studied in this thesis are motivated by the fact that the use of EVs are becoming more and more common and making the logistics decisions in an effective way is also becoming essential. The chapters are organized in a way that at the beginning the very basic problem is studied and in the subsequent chapters, the problem studied in that part is an extension to the previous one. In this thesis, we refer to EV as a commercial road BEV such as a lorry or van.

A fleet of EVs can be used in a variety of transport needs such as public transportation, home deliveries from grocery stores, postal deliveries and courier services, distribution operations in different sectors.

1.2. The Problem of Routing Electric Vehicles

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) was introduced by Schneider et al. (2014) as an extension to the Green Vehicle Routing Problem (GVRP) of Erdoğan and Miller-Hooks (2012). GVRP concerns “green” vehicles which run with biodiesel, liquid natural gas, or CNG, and have a limited driving range. Hence, the vehicles may need refueling along their route. Refueling is fast; however, the stations for these fuels are scarce. EVRPTW is a variant of the classical VRPTW where the fleet consists of EVs that may need to visit stations to have their batteries recharged in order to continue their route, as in GVRP. On the other hand, the recharging operation may take a significant amount of time, especially when compared to relatively short refueling times of liquid fuels. Recharging may take place at any battery level and the recharge time is proportional to the amount charged. After the recharge, the battery is assumed to be full. The number of stations is usually few and the stations are dispersed in distant locations, which increases the difficulty of the problem.

1.3. Thesis Organization

Chapter 2 studies a variant of the EVRPTW introduced by Schneider et al. (2014) where partial recharging is allowed instead of recharging the battery up to the capacity at the recharging stations which is more practical in the real world due to shorter recharging duration. This relaxation brings advantages in terms of meeting time windows of the customers. In full charging scheme, sometimes it is not necessary to recharge the battery fully since the recharging point is close to the return point. Similarly, the EV may recharge a small amount at a station in order to catch the time windows of a specific customer and recharge more at a station visited afterwards. In this way, more customers may be merged in fewer number of vehicles and fleet size may be decreased. Furthermore, in some cases, even if the fleet size does not change, it is possible to save from the total distance. The primary objective of the problem is minimizing the fleet size and the secondary objective is minimizing the total distance. We formulate this problem as a 0-1 mixed integer linear program and develop an Adaptive Large Neighborhood Search (ALNS) algorithm to solve it efficiently. We apply several removal and insertion mechanisms by selecting them dynamically and adaptively based on their past performances, including new mechanisms specifically designed for EVRPTW and EVRPTW-PR. These new mechanisms include the removal of the stations independently or along with the preceding or succeeding customers and the insertion of the stations with determining the charge amount based on the recharging decisions. We test the performance of ALNS by using benchmark instances from the recent literature. The computational results show that the proposed method is effective in finding high quality solutions and the partial recharging option may significantly improve the routing decisions. This study was published in Transportation Research Part C: Emerging Technologies as "*Partial recharge strategies for the electric vehicle routing problem with time windows*" by Merve Keskin and Bülent Çatay. This chapter introduced partial recharging to the EVRP literature and presented benchmark results.

Chapter 3 analyzes the case in which the recharging stations have multiple types of chargers. They differ in their recharging rates and unit costs of energy. There are three types of chargers, namely slow, fast and super-fast. As expected, slow charger is the cheapest one and the super-fast charger recharges the battery with the most expensive cost. The advantage of using fast chargers is saving time and being able to catch the time windows of the customers which cannot be reached otherwise due to long recharging times. In this way, customers can be merged and total travelled distance or sometimes the fleet size may be reduced. The primary objective is minimizing the number of vehicles as in the previous problem, and the secondary objective is minimizing the total energy cost. Here we are minimizing the total energy cost since the chargers have different costs per energy recharged. We formulated this problem as a mixed integer linear program and solved the small instances using CPLEX. To solve the larger problems, we develop a matheuristic approach which couples the ALNS approach with a mixed integer linear programming model. Our ALNS is equipped with various destroy-repair algorithms to efficiently explore the neighborhoods and uses CPLEX to strengthen the routes obtained. We carried out extensive experiments to investigate the benefits of fast recharges and test the performance of our algorithm using benchmark instances from the literature. The results show the effectiveness of the proposed matheuristic and demonstrate the benefits of fast chargers on the fleet size and energy costs. This study was published in Computers & Operations Research as "*A Matheuristic Method for the Electric Vehicle Routing Problem with Time Windows and Fast Chargers*" by Merve Keskin and Bülent Çatay. Although there were studies considering multiple chargers, this chapter was the first which also addresses the time windows for the customers and the depot.

Chapter 4 relaxes the assumption that EVs receive service right after they arrive at a recharging station. If the stations are public, then the EVs cannot have a control on the scheduling of the stations and they may face queues. It means, they may wait before they recharge their batteries.

Furthermore, these waiting times may differ from one time interval to another within the day due to the rush hours. Hence, the EVs have to make routing plans according to these waiting times because otherwise they may wait too long and be late for the customers. In this study, the planning horizon is split into a set of time intervals and for each interval, different waiting times are assigned to the stations. It is assumed that the stations have M/G/1 queueing system and we make use of the average waiting time equations to generate waiting times. We further assume that the customers and the depot have soft time windows. If the EV arrives at a customer earlier than the service beginning time, it has to wait until that time, but if it arrives later than the late service time, a penalty, proportional to the lateness, is paid. For the depot, this penalty is paid as overtime wage to the driver. A regular wage is also paid to the driver on unit time basis. The problem is to find routes such that total cost of energy, penalty, driver regular and overtime wages, and operating EVs is minimized. We formulate the problem as a mixed integer linear program and solve small instances with CPLEX. For the larger instances, we develop a matheuristic which is a combination of Adaptive Large Neighborhood Search and of the solution of a mixed integer linear program. We perform experiments on benchmark instances. Our results show the impact of waiting times on routing decisions. This study was carried out with the help of Prof. Gilbert Laporte when the author was visiting student at CIRRELT (Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport) and is submitted to Computers & Operations Research as "*Electric Vehicle Routing Problem with Soft Time Windows and Time-Dependent Waiting Times at Recharging Stations*" by Merve Keskin, Gilbert Laporte and Bülent Çatay. The contribution of this chapter is addressing time-dependent waiting times at the stations which was not studied before.

Chapter 5 generalizes the problem studied in the previous chapter and considers stochastic waiting times at the recharging stations. Here the waiting times in the queues are not approximated with the average values, but they are random variables. Similar to the previous study, we assume that the stations have M/G/1 queueing system. The problem is modeled as a two-stage stochastic program with recourse. In the first stage, an a priori plan is made using the expected waiting times. Then, each time an EV arrives at a station, the random waiting time realizes. The customers and the depot have hard time windows and a failure occurs if an EV arrives at a customer or at the depot after their service finish time. In this case, a recourse action should be taken to correct the initial solution and make it feasible with the realized waiting time. The randomness of the waiting times is modeled using a set of scenarios and to calculate the probabilities and expected values, stochastic simulation is used. To solve the problem, an ALNS algorithm is proposed with some well-known operators from the literature as well as newly introduced mechanisms. Results show that waiting times are essential in planning and using expected waiting times does not always yield good solutions. This chapter introduced presence of random waiting times at the stations which was not addressed before.

Finally, the last chapter concludes the thesis and gives an outlook on future directions of research.

Chapter 2

Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows

2.1. Introduction

EVs move with electric propulsion and can be used in a variety of transport needs such as public transportation, home deliveries from grocery stores, postal deliveries and courier services, and distribution operations in different sectors. Although EVs enable zero- or low-emission logistics services, operating an EV fleet has several drawbacks such as: (i) low energy density of batteries compared to the fuel of combustion engined vehicles; (ii) limited number of public charging stations; and (iii) long recharging times (Touati-Moungla and Jost, 2011). Battery swap may remedy the last; however, swapping raises additional issues in battery design and compatibility, battery degradation, ownership, and swap station infrastructure. Under these limitations, routing an EV fleet arises as a challenging combinatorial optimization problem in the Vehicle Routing Problem (VRP) literature.

In this chapter, we relax the full recharging (FR) restriction and allow partial recharging (PR) which is more practical in the real world due to shorter recharging duration. When the vehicle visits a station near the end of its route, FR may not be needed for the vehicle to return to the depot. A similar situation may exist between two consecutive recharges. Saving from recharging time may allow the vehicle to catch the time window of an otherwise unvisited customer, thus, may improve the solution.

In the PR scheme, the recharge quantity is associated with a continuous decision variable. We refer to this problem as EVRPTW and Partial Recharges (EVRPTW-PR) and formulate it as 0-1 mixed integer linear program. Note that determining the recharge quantities brings significant difficulties to the problem. Since the problem is intractable for large instances, we propose an ALNS approach to solve it efficiently. ALNS is based on the destroy-and-repair framework where at each iteration the existing feasible solution is destroyed by removing some customers and recharging stations from their routes and then repaired by inserting the removed customers to the solution along with stations when recharging is necessary. Several removal and insertion algorithms are applied by selecting them dynamically and adaptively based on their past performances**.** The new solution is accepted according to the Simulated Annealing criterion. Our approach combines the removal and insertion mechanisms presented in Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007) and Demir et al. (2012) with some new mechanisms designed specifically for EVRPTW and EVRPTW-PR. Our computational tests show that the proposed ALNS is effective in finding good quality solutions and improves some of the best-known solutions in the literature. Furthermore, our results reveal that the PR scheme may substantially improve the routing decisions.

The contributions of this study can be summarized as follows:

- We extend EVRPTW to a PR scheme, which is more general and practical, and present the mathematical programming formulation of the problem.
- We propose an effective ALNS method to solve the EVRPTW and EVRPTW-PR. The proposed method introduces new removal and insertion mechanisms to tackle the more complex problem structure of VRPs where the fleet consists of EVs.
- We validate the performance of the proposed method using the EVRPTW instances of Schneider et al. (2014) and improve the best-known solutions of four problems.
- We show that the PR scheme may improve the solutions obtained with the FR scheme substantially.

The remainder of the chapter is organized as follows: Section 2.2 reviews the related studies in the literature. Section 2.3 describes the problem and formulates the mathematical model. The proposed ALNS method is presented in Section 2.4. Section 2.5 provides the computational study and discusses the results. Finally, concluding remarks and future research directions are given in Section 2.6.

2.2. Related Literature

There are relatively few studies on route optimization of AFVs. Artmeirer et al. (2010) study this problem within a graph-theoretic context and propose extensions to general shortest path algorithms that address the problem of energy-optimal routing. They formalize energy-efficient routing in the presence of rechargeable batteries as a special case of the constrained shortest path problem and present an adaption of a general shortest path algorithm that respects the given constraints. Wang and Shen (2007) develop a model that minimizes the number of tours and total deadhead time hierarchically. The driving range of the vehicle is limited but the charging durations, time windows and vehicle capacities are not considered. A multiple ant colony algorithm is proposed to solve the problem.

Conrad and Figliozzi (2011) introduce the Recharging VRP (RVRP), a new variant of the VRP where the EVs are allowed to recharge at selected customer locations. The model has dual objectives: the primary objective minimizes the number of routes or vehicles whereas the secondary objective minimizes the total costs associated with the travel distance, service time and vehicle recharging. The latter is a penalty cost incurred at each recharge. The EV is charged while servicing the customer and the charging time is constant. The battery level departing from a customer depends on the choice of full charge or partial charging. In the partial charge case the battery is charged to a specified level such as 80% of battery capacity. Conrad and Figliozzi (2011) use an iterative construction and improvement procedure to solve this problem but do not provide its details.

Wang and Cheu (2012) investigate the operations of an electric taxi fleet. Their model minimizes the total distance travelled under the recharging constraints and maximum route time. The battery is consumed at a given rate per distance and can be replenished at the recharging stations. Charging times are constant and after charging the battery becomes full. They construct an initial solution using one of the nearest-neighbor, sweep and earliest time window insertion heuristics and improve it using Tabu Search (TS). They also suggest three different recharging plans which provide different driving ranges and compare the results against the full charging scheme.

Omidvar and R. Tavakkoli-Moghaddam (2012) tackle an AFV routing problem with time-windows and propose a mathematical model that minimizes total costs related to vehicles, distance travelled, travel time and emissions. The refueling times are assumed to be constant. They use the Simulated Annealing (SA) and Genetic Algorithm (GA) approaches and compare their performances. Worley et al. (2012) address the problem of locating recharging stations and designing EV routes simultaneously. The objective is to minimize the sum of the travel costs, recharging costs, and costs of locating recharging stations. A solution method is not proposed and left as future work.

Erdoğan and Miller-Hooks (2012) consider the routing of AFVs within the GVRP context and formulate the mathematical model. The model aims at minimizing the total distance travelled where the length of the routes is restricted. Fuel is consumed with a given rate per traveled distance and can be replenished at the alternative fuel stations. Refueling times are assumed to be fixed and after refueling the tank becomes full. The model does not involve time windows and vehicle capacity constraints. They propose two heuristics to solve the GVRP. The first is a Modified Clarke and Wright Savings (MCWS) algorithm which creates routes by establishing feasibility through the insertion of AFSs, merging feasible routes according to savings values, and removing redundant AFSs. The second is a Density-Based Clustering Algorithm (DBCA) based on the cluster-first and route-second approach. DBCA forms clusters of customers such that every vertex within a given radius contains at least a predefined number of neighbors. Subsequently, the MCWS algorithm is applied to the identified clusters. To test the performance of these two heuristics, they design two sets of problem instances. The first consists of 40 small-sized instances with 20 customers while the second involves 12 instances with up to 500 customers.

Recently, Felipe et al. (2014) extend GVRP for EVs by allowing partial recharges using multiple technologies, i.e. using different power options. As in GVRP, the problem does not involve time windows but EVs have capacity and total route duration limits. The authors formulate the mathematical programming model and present constructive and deterministic local search algorithms as well as a metaheuristic extension based on an SA framework. The computational tests on both randomly generated and GVRP data show that using partial recharge strategies and providing multiple recharge technologies can achieve cost and energy savings and ensure feasibility in some instances.

Schneider et al. (2014) develop a hybrid metaheuristic that combines the Variable Neighborhood Search (VNS) algorithm with TS for solving EVRPTW. They test the performance of the proposed method on benchmark instances of GVRP and Multi-Depot VRP with Inter-Depot Routes. They also generate new instances for EVRPTW by modifying Solomon (1987) data and report their results. Desaulniers et al. (2014) tackle the same problem by considering four recharging strategies which are the combinations of 2 cases. In the first case, the EVs are allowed to recharge only once (single) or multiple times (multiple). In the second case, batteries are recharged partially (PR) or fully (FR). Hence, they analyze the strategies single-FR, single-PR, multiple-FR, multiple-PR and attempt to solve the problems optimally using branch-price-and-cut algorithms. Goeke and Schneider (2015) extend EVRPTW to the routing of a mixed fleet of EVs and internal combustion engine (ICE) vehicles. Their objective function consists of an energy consumption function of speed, gradient, and cargo load distribution, and they propose an ALNS approach to solve it. Hiermann et al. (2015) address the Fleet Size and Mix Vehicle Routing Problem with Time Windows where the fleet consists of EVs. They also implement an ALNS algorithm equipped with local search and labeling procedures.

2.3. Problem Description and Model Formulation

EVRPTW-PR concerns a set of customers with known demands, delivery time windows, and service durations. The deliveries are performed by a homogeneous fleet of EVs with fixed loading capacities and limited cruising ranges. While the vehicle is traveling, the battery charge level decreases proportionally with the distance traversed and the vehicle may need to visit a recharging station in order to continue its route. The battery is recharged at any quantity and the duration of the recharge depends on the initial state of battery charge. The vehicle departs from the depot fully charged and may arrive at/depart from a station with any SoC and it returns to depot with an empty battery if it has been recharged once during its route. If the EV does not visit any stations, it may still arrive at the depot with an empty battery if the total distance traveled is equal to the battery capacity. Otherwise, the arrival SoC at the depot will have a positive value.

Figure 2.1 illustrates an example involving ten customers (C1-C10), four stations (S1-S4), and the depot (D) which can also be used for recharging. The percentage values along the routes show the battery SoC when the vehicle arrives at a customer or a station and when it departs from the station after having its battery recharged. EV1 services C1 and C2, returns to the depot without any recharging. EV2 visits S1 after servicing C4 and has its battery recharged before visiting C5 and C3. On the other hand, EV3 is recharged once in S4 and twice in S3. Note that a station can be visited multiple times by the same (see S3) or different vehicles and each station is not necessarily visited (see S2). In what follows, we provide the mathematical model for EVRPTW-PR following the notation and formulation of Schneider et al. (2014).

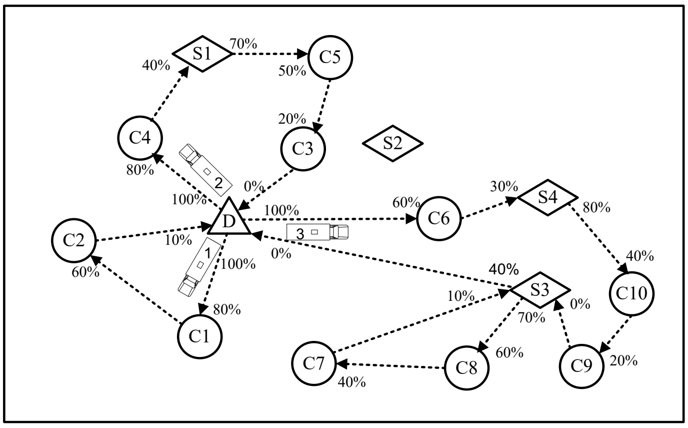


Figure 2.1. An illustrative example

Let $V = \{1, \dots, N\}$ denote the set of customers and F denote the set of recharging stations. Since a recharging station may be visited more than once depending on the route structure, we create F'which is the set of dummy vertices generated to permit several visits to each vertex in the set F. Vertices 0 and $N + 1$ denote the depot and every route starts at 0 and ends at $N + 1$. Let V' be the set of vertices with $V' = V \cup F'$. In order to indicate that a set contains the respective instance of the depot, the set is subscripted with 0 or $N + 1$. Hence, $F_0' = F' \cup \{0\}$, $V_0' = V' \cup \{0\}$, and $V_{N+1}' = V' \cup \{N + 1\}$. Now we can define the problem on a complete directed graph $G = (V_{0,N+1}', A)$ with the set of arcs $A = \{(i,j)| \ i,j \in V_{0,N+1}', \ i \neq j\}$. Each arc is associated with a distance d_{ij} and travel time t_{ij}. The battery charge is consumed at a rate of h and every traveled arc consumes $h \cdot d_{ij}$ of the remaining battery. Each customer $i \in V$ has positive demand q_i, service time s_i and time window $[e_i, l_i]$. All EVs have a load capacity of C and a battery capacity of Q. At a recharging station, the battery is charged at a recharging rate of g. The decision variables, τ_i, u_i and y_i keep track of the service starting time, remaining cargo level and battery SoC on arriving to vertex $i \in V_{0,N+1}'$, respectively. The binary decision variable x_{ij} takes value 1 if arc (i,j) is traversed and 0 otherwise. In Schneider et al. (2014) the battery is always recharged to full capacity, i.e. the recharge amount is $(Q - y_i)$. The EVRPTW-PR allows partial recharges by defining a new decision variable Y_i which represents the battery SoC on departure from station i.

$$\min \sum_{i \in V_0', j \in V_{n+1}'} d_{ij} x_{ij} \tag{2.1}$$

subject to

$$\sum_{j \in V_{n+1}'} x_{ij} = 1 \qquad \forall i \in V \tag{2.2}$$

$$\sum_{j \in V_{n+1}'} x_{ij} \leq 1 \qquad \forall i \in F' \tag{2.3}$$

$$\sum_{i \in V_0'} x_{ij} - \sum_{i \in V_{n+1}'} x_{ji} = 0 \qquad \forall j \in V' \tag{2.4}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0 (1 - x_{ij}) \leq \tau_j \qquad \forall i \in V_0, \forall j \in V_{n+1}' \tag{2.5}$$

$$\tau_i + t_{ij} x_{ij} + g(Y_i - y_i) - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j \qquad \forall i \in F', \forall j \in V_{n+1}' \tag{2.6}$$

$$e_j \leq \tau_j \leq l_j \qquad \forall j \in V_{0,n+1}' \tag{2.7}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}) \qquad \forall i \in V_0', \forall j \in V_{n+1}' \tag{2.8}$$

$$0 \leq u_0 \leq C \tag{2.9}$$

$$0 \le y_j \le y_i - (h \cdot d_{ij})x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in V, \forall j \in V'_{n+1} \qquad (2.10)$$

$$0 \le y_j \le Y_i - (h \cdot d_{ij})x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in F'_0, \forall j \in V'_{n+1} \qquad (2.11)$$

$$y_i \le Y_i \le Q \qquad \forall i \in F'_0 \qquad (2.12)$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V'_0, \forall j \in V'_{n+1} \qquad (2.13)$$

The objective function (2.1) minimizes the total distance traveled. Constraints (2.2) and (2.3) handle the connectivity of customers and visits to recharging stations, respectively. The flow conservation constraints (2.4) enforce that the number of outgoing arcs equals to the number of incoming arcs at each vertex. Constraints (2.5) and (2.6) ensure the time feasibility of arcs leaving the customers (and the depot), and the stations, respectively. Constraints (2.7) enforce the time windows of the customers and the depot. In addition, constraints (2.5)−(2.7) eliminate the sub-tours. Constraints (2.8) and (2.9) guarantee that demand of all customers are satisfied. Constraints (2.10) and (2.11) keep track of the battery SoC and make sure that it is never negative. Constraints (2.12) determine the battery SoC after the recharge at a station and make sure that it does not exceed its capacity. Finally, constraints (2.13) define the binary decision variables.

Proposition 2.1: If an optimal solution exists such that an EV leaves the depot with its battery partially charged, i.e., $Y_0^* < Q$, then the same EV departing from the depot fully charged is also optimal, i.e., $Y_0^* = Q$ is also optimal.

Proof: Let $Y_0^* < Q$ be optimal. Since fully recharging the battery at the depot does not delay the departure time of the EV, $Y_0^* = Q$ must also be optimal.

Corollary 2.1: If $Y_0^* < Q$ is optimal, then the problem has infinite multiple optima.

Proof: Let $\bar{Y}_0 < Q$ be optimal and $\bar{Y}_0 + \varepsilon \le Q$ not, where ε is a small positive scalar. Then following Proposition 2.1, multiple optima exist such that $\bar{Y}_0 \le Y_0^* \le Q$.

Proposition 2.2: If an optimal solution exists such that an EV has been recharged at least once and returns to the depot at the end of its route with positive battery state, i.e. $y_{n+1}^* > 0$, then its return to the depot with empty battery is also optimal, i.e. $y_{n+1}^* = 0$ is also optimal.

Proof: Let $y_{n+1}^* > 0$ be optimal. Since recharging the battery with less energy at the preceding station does not delay the departure time to cause any time window infeasibility, $y_{n+1}^* = 0$ must also be optimal.

Corollary 2.2: If $y^*_{n+1} > 0$ is optimal, then the problem has infinite multiple optima.

Proof: Let $\bar{y}_{n+1} > 0$ be optimal and $\bar{y}_{n+1} + \varepsilon$ not, where ε is a small positive scalar. Then following Proposition 2.2, multiple optima exist such that $0 \le y^*_{n+1} \le \bar{y}_{n+1}$.

Without loss of generality, we assume that an EV departs from the depot with a battery charged in full and returns to the depot with its battery fully consumed if it has been recharged at least once along its route.

2.4. Solution Methodology

We propose an ALNS method to solve the EVRPTW-PR. ALNS was introduced by Ropke and Pisinger (2006a) as an extension of the Large Neighborhood Search (LNS) framework put forward by Shaw (1998). Since local search methods can only make small changes to an existing solution their search space is narrow. Hence, they are unable to move from one promising area to another within the feasible region. To overcome this shortcoming, Ropke and Pisinger (2006a) considered large moves that rearrange up to 40% of the vertices instead of using small moves that relocate or exchange only a few arcs or vertices at each iteration. In a subsequent study, Ropke and Pisinger (2006b) developed a unified ALNS heuristic for a large class of VRP with Backhauls. Pisinger and Ropke (2007) improved this heuristic with additional algorithms and showed that the proposed framework gives competitive results in different VRP variants. Since then, ALNS has been successfully implemented to solve various VRPs, e.g. cumulative capacitated VRP (Ribeiro and Laporte, 2012), pollution-routing problem (Demir et al., 2012), two-echelon VRP (Hemmelmayr et al., 2012), pickup and delivery problems with transshipment (Qu and Bard, 2012) and with vehicle transfers (Masson et al., 2013), VRP with multiple routes (Azi et al., 2014), periodic inventory routing problem (Aksen et al., 2014), and production routing problem (Adulyasak et al., 2014).

2.4.1. Overview of the Proposed ALNS Approach

2.4.1.1. Initial Solution Construction

The initial solution is obtained by iteratively constructing feasible routes. The route construction begins with the nearest customer to the depot. Then, the insertion costs of all unassigned customers to all possible existing positions in the route are determined respecting the time window constraints, i.e. the insertion of customer i between nodes j and k is calculated as $d_{ji} + d_{ik} - d_{jk}$ if customers i, j, and k can be visited within their time windows. Next, the best insertion is performed. If no customer can be inserted because of low battery level, we use the Greedy Station Insertion algorithm described in section 2.4.2.2 and insert a customer along with a recharging station. In that case, the insertion cost becomes the difference between the total distance of the route after and before the insertions of the customer and the recharging station. When no customer can be inserted to the route due to capacity or time-window constraints, the route is finalized and the procedure is repeated by starting with a new route until all customers have been visited. The pseudocode of the initial solution construction procedure is given in Algorithm 2.1.

Algorithm 2.1: Initial solution construction

```
Start a new route with the customer closest to the depot
repeat
    Calculate insertion costs of all unserved customers to the current route
    if no customer can be added then
        Start a new route with the unserved customer closest to the depot
    else
        Select the customer which increases the distance least and make the insertion
    end if
    if a recharging station is needed then
        Perform Greedy Station Insertion
    end if
until all customers are served
```

2.4.1.2. ALNS Procedure

The proposed ALNS heuristic includes four classes of algorithms: Customer Removal (CR), Customer Insertion (CI), Station Removal (SR), and Station Insertion (SI). After the initial solution has been constructed, ALNS tries to improve it iteratively until a stopping condition is satisfied. We use an iteration number limit to terminate the heuristic. At each iteration, the existing feasible solution is destroyed by removing some nodes from their routes using a removal algorithm. These nodes consist of customers, recharging stations, or both. The resulting partial solution is then repaired using an insertion algorithm which heuristically inserts removed customers and/or recharging stations (removed or other) to the existing routes or new routes are created for these removed nodes in an attempt to obtain a better solution than the previous. Several removal and insertion algorithms are applied by selecting them dynamically and adaptively based on a probability calculated using their performances in the previous iterations. In order to calculate the selection probabilities, an adaptive weight w and a score π is assigned to each algorithm. High score corresponds to a successful mechanism and hence the mechanism should be selected with larger probability. Initially, all weights are equal and all scores are 0. In an iteration, if a new best solution has been found, then the scores corresponding to the removal and insertion algorithms which achieved that solution are increased by σ_1. If the algorithms have yielded a better solution than the previous then the scores are increased by σ_2. Finally, if the new solution is worse than the previous but accepted using the simulated annealing rule then the scores are increased by σ_3. The procedure is divided into segments which consist of N_C iterations for customer related mechanisms and N_S for station related mechanisms. At the end of each segment *s*, the weight of algorithm a is updated using the formula $w_a^{s+1} = w_a^s(1-\rho) + \rho\pi_a/\theta_a$, where ρ is the roulette wheel parameter, θ_a is the number of times it was used during segment s and π_a is the score associated with algorithm a. After updating the weights, the probabilities of the algorithms which will be used in the next segment (s+1) are calculated using the formula $P_a^{s+1} = w_a^s / \sum_{l=1}^{m} w_l^s$ and the scores are reset to zero.

The simulated annealing (SA) approach that accepts or rejects a solution is implemented as follows: if the number of vehicles in the new solution is smaller than that of the current solution or if they are equal but the total distance of the new solution is shorter then we accept the new solution. On the other hand, we reject the new solution if it requires more vehicles. When the numbers of vehicles are equal but the distance is longer, the new solution is accepted with probability $e^{-(f(X_{New})-f(X_{Current})/T}$, where , $f(X)$ denotes the total distance of solution X, X_{New} and $X_{Current}$ are the new and current solutions, respectively, and T is the current temperature. T is initially set to T_{init} and decreased at every iteration using the formula $T = T\varepsilon$, where $0 < \varepsilon < 1$ is the *cooling rate* parameter. T_{init} is determined using the *initial temperature control parameter* μ such that a solution which is μ% worse than the initial solution is accepted with probability 0.5.

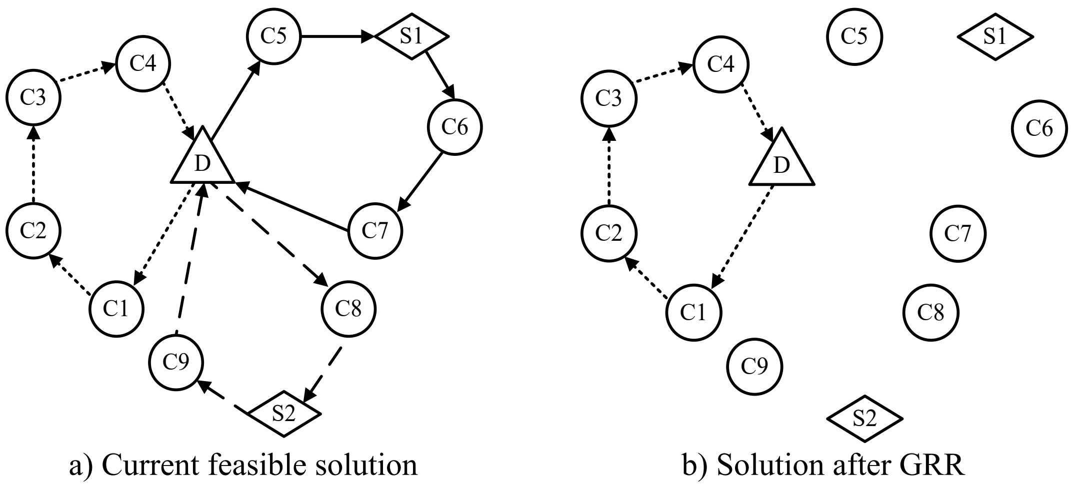
2.4.2. Removal Algorithms

2.4.2.1. Customer Removal

The current solution is destroyed by removing γ customers from the solution according to different rules and adding them in a removal list $\mathcal{L}$. γ depends on the total number of customers n_c and is determined randomly between $\underline{n_c}$ and $\overline{n_c}$ using a uniform distribution. The removal algorithms are selected in an adaptive manner from the set of algorithms CR.

We utilize *Random*, *Worst-Distance*, *Worst-Time*, *Shaw*, *Proximity-Based*, *Demand-Based*, *Time-Based*, *Zone Removal* algorithms presented in the literature and adopt the Route Removal algorithms presented in Emeç et al. (2016). Random Removal algorithm selects γ customers randomly and removes them from the solution. Worst-Distance Removal algorithm determines customers with high cost, where the cost is the sum of distances of the customer from the preceding and succeeding nodes in the route. Then, it removes the customer with $\lfloor \gamma . \lambda^{\kappa} \rfloor^{th}$ highest cost where $\lambda \in [0,1]$ is a random number and $\kappa \geq 1$ is a parameter which introduces randomness in the selection of customers to avoid the selection of the same customers repeatedly and is referred to as *worst removal determinism factor*. Worst-Time Removal algorithm is similar to Worst-Distance Removal algorithm where the cost of customer i is calculated as $|\tau_i - e_i|$.

Shaw Removal is designed to remove customers that are similar to each other with respect to several criteria and uses the following relatedness measure: $R_{ij} = \phi d_{ij} + \phi_2 |\tau_i - \tau_j| + \phi_3 l_{ij} + \phi_4 |q_i - q_j|$ where $\phi_1 - \phi_4$ are the Shaw parameters and $l_{ij} = -1$ if i and j are in the same route, and 1 otherwise. Small R_{ij} means high similarity. The algorithm first selects a customer i randomly. Then, it sorts the non-removed customers in the non-decreasing order of their relatedness value with a customer i and chooses the customer listed in position $\lfloor \gamma . \lambda^{\eta} \rfloor$ where η is a parameter called *Shaw removal determinism factor*. Proximity, Demand and Time-Based Removals are the special cases of *Shaw Removal* where ϕ_1, ϕ_2, ϕ_4 takes the value 1 and the other parameters are assumed to be 0. In the *Zone Removal,* the Cartesian coordinate system in which the nodes are located is divided into n_Z many smaller areas that are called as zones. A zone is randomly selected and all the customers in that zone are removed from the solution. More details of these algorithms can be found in Demir et al. (2012).



a) Current feasible solution b) Solution after GRR

Figure 2.2. Illustration of Greedy Route Removal

Random Route Removal (RRR) randomly chooses ω routes and removes all the customers visited in those routes. ω depends on the number of routes in the current solution and is determined randomly between 10% and m_r% of total number of routes. *Greedy Route Removal* (GRR) algorithm removes ω routes in a greedy way. ω is determined in the same way as in RRR. The routes are sorted in the non-decreasing order of the number of customers serviced and ω routes are removed starting from the first route in the order. The motivation is to distribute the customers in shorter routes into other existing routes in the solution in an attempt to reduce the number of vehicles. The procedure is illustrated in Figure 2.2 for ω=2.

Note that after a predetermined number of iterations N_{RR}, we explicitly perform RRR and GRR for n_{RR} iterations to extensively attempt to reduce the number of vehicles used. RRR and GRR remove the complete routes from the solution. On the other hand, since other removal algorithms remove customers from their routes the battery state, time, and remaining capacity of the EV at its arrival to a node should be updated. Furthermore, some recharges may no longer be necessary and those stations may be removed from the solution. In fact, an EV may visit a recharging station right before or after servicing a customer, and it might be beneficial to remove the customer from the solution with its preceding or succeeding station. So, we introduce the following two operators for the customer removal algorithms in addition to removing customers only (RCO) option:

Remove Customer with Preceding Station (RCwPS): We remove the customer in the removal list along with the preceding station, if any exists. The idea is to eliminate the visit to a station where recharging is not necessarily needed at that battery state if EV no longer visits the removed customer.

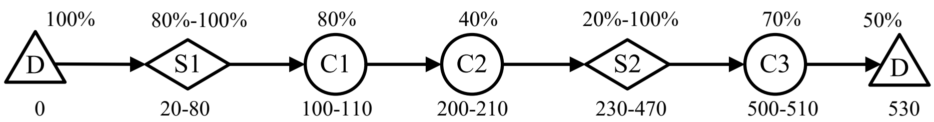
Remove Customer with Succeeding Station (RCwSS): We remove the customer in the removal list along with the succeeding station, if any exists. The idea is similar to RCwPS. The recharging may be needed after departing from a customer in order to be able to reach the next customer in the route. In that case, recharging is not necessarily needed at that battery state if the departure customer is removed from the solution and the station can be removed as well.

2.4.2.2. Station Removal

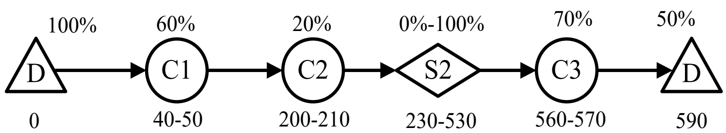
The recharging stations are the crucial components of the problem. Hence, removing them or changing their positions in the visit sequence of a route may also improve the solution. So, after a pre-determined number of iterations, an SR (followed by a Station Insertion) procedure is applied. The number of stations to be removed σ is determined in a similar fashion to γ based on the total number of stations in the current station. The *Random Station* and *Worst Distance Station Removal* mechanisms are similar to their customer removal counterparts. We also use the *Worst-Charge Usage Station Removal* which aims at removing the stations visited with high battery levels and *Full Charge Station Removal* which aims at promoting the PR option.

Worst-Charge Usage Station Removal: The motivation of this algorithm is to make the use of the battery as much as possible before a recharging is needed and increase the efficiency in using the stations. We promote the removal of the stations which an EV visits with relatively higher charge level. The stations are sorted in the non-increasing order of the battery level of the EVs that visit them for recharging and σ stations are removed starting from the first station in the order.

Full Charge Station Removal: The algorithm identifies the stations where EVs are fully charged and removes σ of them randomly.



a) Feasible route before SR



b) Time-window infeasible route after SR

Figure 2.3. Example of time-window infeasibility after SR

After the removal algorithms, the destroyed solution may become infeasible with respect to the time windows. Consider the route shown in Figure 2.3(a) as an example. The % numbers above the route indicate the SoC at the arrival to and departure from a node whereas the numbers under the route show the arrival and departure times. When S1 is removed from the route, the EV can still visit C1 and C2 in the given sequence. However, since its battery is empty, the recharging takes longer at S2, which delays its arrival to C3. Since the EV departs from C3 at a later time, it cannot return to D before the latest arrival time of 550 as shown in Figure 2.3(b).

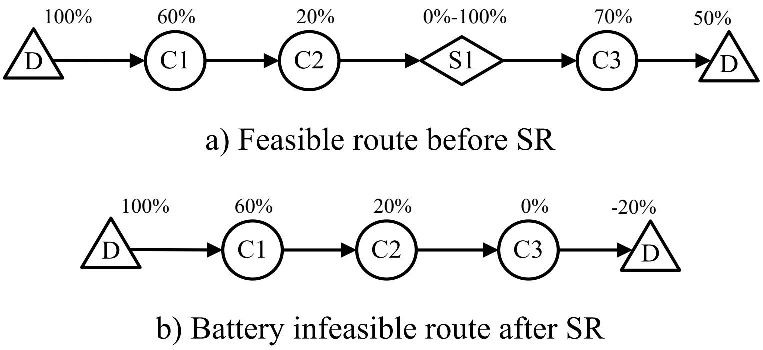


Figure 2.4. Example of battery infeasibility after SR

Figure 2.4 illustrates how the battery infeasibility may occur after a SR. Consider the feasible route in Figure 2.4(a). The EV is charged to full at S1. However, when S1 is removed, the battery level is not sufficient to return to the depot after visiting C3 as shown in Figure 2.4(b).

2.4.3. Insertion Algorithms

2.4.3.1. Customer Insertion

We use the *Greedy* and *Regret Insertion* algorithms from the literature. Greedy Insertion algorithm determines the best insertion position for customer i by calculating the cost of inserting it between all feasible pairs of nodes j and k and selecting the position with the minimum cost. The procedure is repeated for all customers and the customer who has the minimum insertion cost is inserted to its designated position. Regret-k Insertion prevents the myopic nature of Greedy Insertion by avoiding the customers which may yield higher costs in the subsequent iterations. It calculates the difference between the cost of the first and k^{th} best insertions of the customers and insert the one with the highest difference to its best position. In our ALNS we utilize *Regret*-2 and *Regret*-3 methods. In addition, we propose the *Time-Based Insertion* and adapt the *Zone Insertion* of Demir et al. (2012) as follows:

Time Based Insertion: In this algorithm, the insertion cost is calculated as the difference between the total route durations before and after the insertion of a customer. For each customer, the algorithm determines the best insertion position among all routes based on this insertion cost. The customer that increases the route duration the least is selected and inserted. The procedure is repeated for the remaining customers until all customers are inserted. The aim of this algorithm is to increase the number of customers visited by an EV by combining compatible customers with respect to their time windows or distances.

Zone Insertion: The algorithm uses the Time-Based Insertion criterion above when selecting a customer. However, instead of investigating all routes in the solution, it considers the routes within a randomly selected zone only. The zones are determined in the same way with *Zone Removal*.

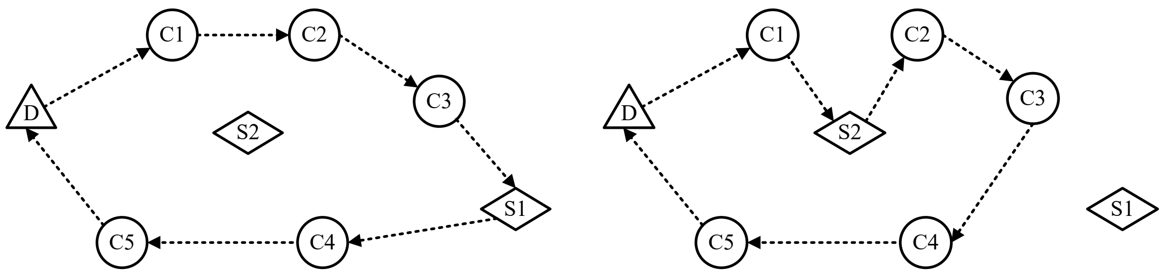
Note that a customer insertion may be feasible with respect to service time-window but infeasible with respect to the battery state (referred to as battery infeasibility). In that case, the Greedy Station Insertion (described in Section 2.4.2.2) is applied to make the destroyed solution charge feasible.

To determine the battery SoC and the recharge amount at a station visited in the implementation of CI algorithms we use the assumptions stated at the end of Section 2.3: an EV departs from the depot with a full battery and returns to the depot by completely consuming its battery if it has been recharged at least once along its route. So, in the case the EV is recharged only once during its route then: (i) if the customer is inserted between the depot and the station the insertion only affects the arrival SoC at the station; (ii) if the customer is inserted between the station and the depot the recharge amount is increased such that the EV returns to the depot with empty battery.

If multiple recharges exist along the route and the customer is inserted between the depot and the first station visited, we follow the procedure (i) described above. If the customer is inserted between two consecutive stations or between the last station visited and the depot the amount recharged at the last station visited is increased by the additional energy needed to visit that customer. If the recharge duration makes the insertion infeasible with respect to service time window of an existing customer, then we attempt to reduce the recharge duration by increasing the battery charge level at the arrival to that station. This is achieved by recharging the EV longer at the previous station making sure that the time-window feasibility of the customers visited between these two consecutive stations is maintained. In any case, if the insertion is feasible with respect to service time window but battery infeasible the Greedy Station Insertion (see Section 2.4.2.2) is applied to make the destroyed solution charge feasible.

2.4.3.2. Station Insertion

After removing some stations, the current feasible solution may become battery infeasible. In order to repair the solution, stations must be inserted to the infeasible routes. We make an infeasible route feasible by identifying the first customer at which the vehicle arrives with a negative battery level and inserting a station into the partial route prior to that customer. The difference from CI algorithms is that SI algorithms do not necessarily insert the stations which have been removed in SR. Since the stations are always available and it is assumed that as many stations as needed are available, any station can be inserted throughout the algorithm. The SR and SI procedures are illustrated on an example in Figure 2.5. A feasible route is depicted in Figure 2.5(a). Suppose, S1 is removed using an SR algorithm. Next, S2 is inserted between C1 and C2 by maintaining both time-window and battery feasibility. The resulting route in Figure 2.5(b) is shorter than the initial.



a) Feasible solution b) Improved solution after SR and SI

Figure 2.5. An improved route after SR and SI procedure

We use the following three SI algorithms:

Greedy Station Insertion (GSI): This algorithm determines the first customer in the route at which the vehicle arrives with negative battery level and inserts the "best" (which increases the distance least) station on the arc between that customer and the previous customer. If this insertion is not feasible, then the previous arcs are attempted in the same manner.

Greedy Station Insertion with Comparison: The algorithm determines the best station on the arc leading to the customer where the battery level is negative as in GSI and compares the outcome with the case of inserting the corresponding best station on the immediate predecessor arc. The insertion which increases the route distance the least is performed. If both insertions are infeasible, the GSI procedure is applied by considering the previous arcs.

Best Station Insertion: We determine the best station insertions between the customer that the EV arrives at with negative battery level and the depot or a previously visited station by considering all the arcs backwards in the route. We select the best feasible insertion and perform it.

The procedure is repeated for all customers where the EVs arrive with negative battery level. If a station insertion cannot be performed feasibly, we return to the previous feasible solution. The battery state of the EV and/or the recharge quantities in the implementation of the SI algorithms are determined in a similar fashion as in CI algorithms.

The pseudocode of the ALNS approach is provided in Algorithm 2.2.

Algorithm 2.2: ALNS algorithm

```
Generate an initial solution
j ← 1
repeat
    if j ≡ 0 (mod N_SR) then
        Select SR algorithm and remove stations
        Select SI algorithm and repair solution
    else if j ≡ 0 (mod N_RR) then
        for n_RR iterations do
            Select RRR or GRR algorithm and remove customers
            Select CI algorithm and repair solution
        end for
    else
        Select CR algorithm and remove customers
        if destroyed solution infeasible then
            Perform Greedy Station Insertion
        end if
        Select CI algorithm and repair solution
    end if
    Using SA criterion, accept/reject solution
    j ← j + 1
    if j ≡ 0 (mod N_C) then
        Update adaptive weights of CR and CI algorithms
    else if j ≡ 0 (mod N_S) then
        Update adaptive weights of SR and SI algorithms
    end if
until stop-criterion met
```

2.5. Computational Study

To validate the performance of the proposed ALNS approach we perform computational experiments using the EVRPTW data. This data set consists of 36 small and 56 large instances generated by Schneider et al. (2014) based on the well-known VRPTW instances of Solomon. The large instances include three main problem classes where 100 customers and 21 recharging stations are clustered (C), randomly distributed (R), and both clustered and randomly distributed (RC) over a 100×100 grid. Each set has also two subsets, type 1 and type 2, which differ by the length of the time windows and the vehicle load and battery capacities. The small instances include three subsets of 12 problems, each involving 5, 10 and 15 customers drawn randomly from the large instances.

We first tuned the parameters values using a subset of EVRPTW instances. Then, we solved the large EVRPTW problems and compare the results with the benchmarks reported in the literature. Finally, we report the solutions we achieved using the EVRPTW-PR setting and discuss the results. The algorithm was coded in the Java programming language.

2.5.1. Parameter Tuning

Our tuning methodology is in line with those adopted in the literature (Ropke and Pisinger, 2006a, 2006b; Pisinger and Ropke, 2007; Demir et al., 2012, Emeç et al., 2016). We selected six problems and performed 10 runs by considering up to 10 values for each parameter. We omitted C1 and C2 problem classes since they usually converged to the same solutions for different parameter values and did not provide much information about the contribution of the parameter value on the solution quality. Consequently, we selected the instances R107, RC101, RC104, RC105, R205 and RC205 for parameter tuning.

For each value we calculated the average percent deviation from the average of the best achieved solutions, determined the one that yielded the least average percent deviation, and fixed the parameter value. We repeated this procedure until all parameter values had been tuned. In the EVRPTW, we set the initial values according to the best values reported in Emeç et al. (2016). In the EVRPTW-PR, we initialized the values using the best values obtained for the EVRPTW and performed another parameter tuning using the same procedure. The parameters, their considered values and the corresponding deviations, and their final values selected are given in Appendix A.

We observed that if the score of the worse solution (σ_3) is greater than the score of the better solution (σ_2). It allows diversification by rewarding non-improved solutions as noted in Ropke and Pisinger (2006a) and Demir et al. (2012). So, our setting of the parameters σ_1, σ_2, and σ_3 also rewards a worse solution more than a better solution as follows: $\sigma_1 \geq \sigma_3 \geq \sigma_2$.

Ropke and Pisinger (2006a) set the number of iterations to 25000 and noted that additional runtime had a minor contribution to the solution quality. Our convergence analysis showed similar results. So, we also set the number of iterations to 25000. The lower and upper limits for the number of customers n_c to be removed are set as $\underline{n_c} = min\,\{0.1|N|, 30\}$ and $\overline{n_c} =$

$min\,\{0.4|N|, 60\}$, respectively. The removal algorithms are selected in an adaptive manner from the Customer Removal or Station Removal algorithms.

2.5.2. Numerical Results for the EVRPTW Instances

In this section, we investigate the effectiveness of the proposed algorithm in the case where partial recharges are not allowed, and aim at validating its performance by comparing it to the state-of-the-art methodologies designed for solving that particular case. Table 2.1 compares our average results and the average of the results reported in Goeke and Schneider (2015) and Hiermann et al. (2015) to the original results given in Schneider et al. (2014). Our results were obtained using the parameters tuned to the values given in Appendix A. The benchmark results from the literature were also obtained in the same way by fixing the parameters of the corresponding algorithm to the values tuned instead of running the algorithms using different values for the parameters. The first column in Table 2.1 denotes the instances. '$\#Veh$' and 'TD' refer to the number of vehicles and the total distance, respectively. '$\Delta\%$' is the percentage deviation from the distances reported in Schneider et al. (2014), if the number of vehicles is same, and is calculated as $(TD^{SSG} - TD^{M})/TD^{M}$, where SSG stands for the VNS/TS of Schneider et al. (2014) and M refers to the corresponding method, i.e. GS for Goeke and Schneider (2015), HPH for Hiermann et al. (2015), and KÇ for this study. A negative $\Delta\%$ value means improvement.

Table 2.1. Average results for EVRPTW obtained with fixed parameters

Instance	SSG		GS		HPH		KÇ	
Type	*#Veh*	*TD*	*#Veh*	Δ%	*#Veh*	Δ%	*#Veh*	Δ%
C1	10.67	1050.04	10.67	-0.31	10.67	0.16	10.89	0.78
C2	4.00	640.92	4.00	0.00	4.00	0.06	4.00	0.00
R1	12.83	1268.60	12.83	-0.80	13.00	0.11	13.25	0.69
R2	2.64	919.04	2.64	-0.47	2.64	0.53	2.82	-0.07
RC1	13.13	1415.84	13.13	-0.50	13.00	-0.48	13.38	0.13
RC2	3.13	1146.76	3.13	-0.15	3.13	0.95	3.25	0.08
Average				-0.41		0.25		0.25

SSG: Schneider et al. (2014), GS: Goeke and Schneider (2015), HPH: Hiermann et al. (2015), and KÇ: Our method

Our results show that our ALNS approach performs better in type-2 problems but it converges to solutions with one additional vehicle in several instances as compared to other methods. Even though the performance of Hiermann et al. (2015) is better in type-1 problems, the overall performance of our approach is similar. We also observe that the recent work of Goeke and Schneider (2015) has a superior performance, improving many of the best solutions to date. We note that Goeke and Schneider (2015) used the numbers of vehicles achieved in Schneider et al. (2014) as *a priori* information to construct their initial routes, which might have a positive effect both in run time and solution quality. Nevertheless, with fixed parameters our approach improved the best solutions of 11 instances.

Table 2.2. Comparison with the best-known solutions of the EVRPTW instances

	BKS			KÇ				BKS			KÇ		
Inst.	*#Veh*	*TD*	Ref.	*#Veh*	*TD*	Δ%	Inst.	*#Veh*	*TD*	Ref.	*#Veh*	*TD*	Δ%
c101	12	1053.83	SSG	12	1053.83	0.00	c201	4	645.16	SSG	4	645.16	0.00
c102	11	1051.38	GS	11	1056.12	0.45	c202	4	645.16	SSG	4	645.16	0.00
c103	10	1034.86	GS	11	1001.81	-	c203	4	644.98	SSG	4	644.98	0.00
c104	10	961.88	GS	10	**951.57**	**-1.08**	c204	4	636.43	SSG	4	636.43	0.00
c105	11	1075.37	SSG	11	1075.37	0.00	c205	4	641.13	SSG	4	641.13	0.00
c106	11	1057.65	HPH	11	1057.65	0.00	c206	4	638.17	SSG	4	638.17	0.00
c107	11	1031.56	SSG	11	1031.56	0.00	c207	4	638.17	SSG	4	638.17	0.00
c108	10	1095.66	GS	11	1015.68	-	c208	4	638.17	SSG	4	638.17	0.00
c109	10	1033.67	GS	10	1069.16	3.32							
r101	18	1663.04	HPH	18	1679.06	0.95	r201	3	1264.82	SSG	3	1265.67	0.07
r102	16	1487.41	GS	16	1519.80	2.13	r202	3	1052.32	SSG	3	1052.32	0.00
r103	13	1271.35	GS	13	1312.50	3.14	r203	3	895.54	GS	3	895.54	0.00
r104	11	1088.43	SSG	12	1071.89	-	r204	2	779.49	GS	2	780.98	0.19
r105	14	1442.35	GS	15	1383.29	-	r205	3	987.36	GS	3	987.36	0.00
r106	13	1324.10	GS	14	1276.15	-	r206	3	922.19	GS	3	922.70	0.06
r107	12	1150.95	GS	12	**1148.43**	**-0.22**	r207	2	845.26	GS	2	847.14	0.22
r108	11	1050.04	SSG	11	1051.59	0.15	r208	2	736.12	GS	2	736.12	0.00
r109	12	1261.31	GS	13	1214.72	-	r209	3	867.05	GS	3	871.22	0.48
r110	11	1119.50	GS	12	1097.89	-	r210	3	846.20	GS	3	**843.65**	**-0.30**
r111	12	1106.19	SSG	12	1109.14	0.27	r211	2	827.89	GS	3	761.56	-
r112	11	1016.63	GS	11	1038.74	2.13							
rc101	16	1726.91	HPH	16	1731.07	0.24	rc201	4	1444.94	SSG	4	1446.84	0.13
rc102	14	1552.08	HPH	15	1551.69	-	rc202	3	1410.74	GS	3	1450.34	2.73
rc103	13	1350.09	GS	13	1351.73	0.12	rc203	3	1055.19	GS	3	1069.27	1.32
rc104	11	1227.25	GS	11	1232.45	0.42	rc204	3	884.80	GS	3	887.45	0.30
rc105	14	1475.31	HPH	14	**1473.24**	**-0.14**	rc205	3	1273.55	GS	3	1277.60	0.32
rc106	13	1427.21	GS	14	1414.99	-	rc206	3	1188.63	GS	3	1207.64	1.57
rc107	12	1274.89	SSG	12	1283.05	0.64	rc207	3	985.03	GS	3	994.48	0.95
rc108	11	1197.83	GS	11	1209.11	0.93	rc208	3	836.29	GS	3	841.34	0.60
Avg	12.21			12.52		0.60		3.19			3.22		0.33

SSG: Schneider et al. (2014), GS: Goeke and Schneider (2015), HPH: Hiermann et al. (2015), and KÇ: Our method

In Table 2.2, we report the best solutions we observed throughout our entire experimental study and compare them against the best-known solutions (BKS) from the literature achieved in a similar way. Note that Schneider et al. (2014) and Goeke and Schneider (2015) reported the best results they obtained throughout all the experiments they performed as their BKS whereas Hiermann et al. (2015) performed their tests with fixed parameters only. The results in Table 2.2 also show that our ALNS approach performs better in type-2 problems. Although it finds solutions with one additional vehicle in some instances, it improved the best-known solutions of four instances, of which three are type-1 problems. Furthermore, it achieved the same best-known solution in 16 instances. In comparison, Schneider et al. (2014) found the best-known solutions in 18 instances whereas Hiermann et al. (2015) and Goeke and Schneider (2015) improved the solutions of 4 and 30 instances, respectively.

With respect to computational effort, it is not possible to make a fair comparison among the algorithms since they utilize different processors and data structures. We only note that Schneider et al. (2014), Hiermann et al. (2015), and Goeke and Schneider (2015) report average run times of 15.34 minutes[1], 15.92 minutes[2], and 2.78 minutes[3], respectively, whereas our average run time is comparable to the first two with 12.26 minutes [4]. Goeke and Schneider (2015) has an outstanding run time performance; yet it is not possible to assess the contribution of feeding the *a priori* number of vehicles information to this performance.

2.5.3. Experiments on EVRPTW Instances Using PR Scheme

We first analyze the performance of our ALNS on small EVRPTW instances. This allows us to make a comparison with the (near-)optimal solutions found by using CPLEX 12.6.1 both for the FR and PR cases. Next, we solve the large instances using two different ALNS implementations and discuss potential gains that can be achieved through different PR strategies.

2.5.3.1. Numerical Results for Small-Size Instances

Schneider et al. (2014) provided the optimal solution for 25 small-size problems and upper bound for the remaining 11 for the FR case using CPLEX 12.2 with a time limit of 7200 seconds. We solved all these problems with CPLEX 12.6.1 and confirmed the optimality of these upper bounds. Note that ALNS was also able to find the optimal solution for all these instances.

For the PR case, we report the solutions obtained by CPLEX and ALNS in Table 2.3 in comparison with the optimal solutions achieved with FR. 'FR Optimal' refers to the optimal solution of the FR scheme and '$t(sec)$' is the run time in seconds. The time limit for CPLEX is set to 7200 seconds. 'Δ^{CPLEX} %' denotes the percentage deviation of the total distance of the ALNS solution from the total distance of the solution found by CPLEX. 'Δ^{FR} %' gives the

1 Intel Core i5 processor with 2.67 GHz speed and 4 GB RAM, operating on Windows 7 Professional
2 Intel Core2 Quad CPU Q6600 processor with 2.40 GHz speed and 4 GB RAM, operating on 64-bit Linux
3 Intel Core i7 processor with 2.8 GHz speed and 8 GB RAM, operating on Windows 7 Enterprise
4 Intel Xeon E5 processor with 3.30 GHz speed and 32 GB RAM, operating on 64-bit Windows 7

percentage deviation from the optimal distance of the solution with the FR scheme and shows the benefit of adopting the PR strategy.

Table 2.3. Comparison of results obtained with CPLEX and ALNS on the small-size instances

	FR Optimal		PR CPLEX			PR ALNS				
Instance	*#Veh*	*TD*	*#Veh*	*TD*	*t(sec)*	*#Veh*	*TD*	Δ^{CPLEX}%	Δ^{FR}%	*t(sec)*
C101-5	2	257.75	2	257.75	0.31	2	257.75	0.00	0.00	0.03
C103-5	1	176.05	1	175.37	2.73	1	175.37	0.00	-0.39	0.05
C206-5	1	242.55	1	242.56	5.38	1	242.56	0.00	0.00	0.07
C208-5	1	158.48	1	158.48	1.37	1	158.48	0.00	0.00	0.06
R104-5	2	136.69	2	136.69	0.47	2	136.69	0.00	0.00	0.04
R105-5	2	156.08	2	156.08	3.39	2	156.08	0.00	0.00	0.04
R202-5	1	128.78	1	128.78	0.95	1	128.78	0.00	0.00	0.08
R203-5	1	179.06	1	179.06	1.12	1	179.06	0.00	0.00	0.10
RC105-5	2	241.30	2	233.77	3.06	2	233.77	0.00	-3.22	0.03
RC108-5	2	253.93	2	253.93	3.76	2	253.93	0.00	0.00	0.04
RC204-5	1	176.39	1	176.39	2.17	1	176.39	0.00	0.00	0.08
RC208-5	1	167.98	1	167.98	1.05	1	167.98	0.00	0.00	0.07
C101-10	3	393.76	3	388.25	50.26	3	388.25	0.00	-1.42	0.10
C104-10	2	273.93	2	273.93	5.15	2	273.93	0.00	0.00	0.17
C202-10	1	304.06	1	304.06	7.52	1	304.06	0.00	0.00	0.20
C205-10	2	228.28	2	228.28	2.01	2	228.28	0.00	0.00	0.16
R102-10	3	249.19	3	249.19	1.83	3	249.19	0.00	0.00	0.11
R103-10	2	207.05	2	206.12	6.76	2	206.12	0.00	-0.45	0.17
R201-10	1	241.51	1	241.51	11.40	1	241.51	0.00	0.00	0.21
R203-10	1	218.21	1	218.21	1.62	1	218.21	0.00	0.00	0.62
RC102-10	4	423.51	4	423.51	3.07	4	423.51	0.00	0.00	0.09
RC108-10	3	345.93	3	345.93	2.90	3	345.93	0.00	0.00	0.09
RC201-10	1	412.86	1	412.86	7200.00	1	412.86	0.00	0.00	0.17
RC205-10	2	325.98	2	325.98	3.26	2	325.98	0.00	0.00	0.19
C103-15	3	384.29	3	348.46	1008.00	3	348.46	0.00	-10.28	0.23
C106-15	3	275.13	3	275.13	0.47	3	275.13	0.00	0.00	0.15
C202-15	2	383.62	2	383.62	24.07	2	383.62	0.00	0.00	0.29
C208-15	2	300.55	2	300.55	0.92	2	300.55	0.00	0.00	0.26
R102-15	5	413.93	5	412.78	7200.00	5	412.78	0.00	-0.28	0.12
R105-15	4	336.15	4	336.15	1.39	4	336.15	0.00	0.00	0.09
R202-15	2	358.00	2	358.00	462.89	2	358.00	0.00	0.00	0.51
R209-15	1	313.24	1	313.24	610.64	1	313.24	0.00	0.00	0.92
RC103-15	4	397.67	4	397.67	20.27	4	397.67	0.00	0.00	0.12
RC108-15	3	370.25	3	370.25	101.45	3	370.25	0.00	0.00	0.15
RC202-15	2	394.39	2	394.39	113.43	2	394.39	0.00	0.00	0.31
RC204-15	1	407.45	1	403.38	7200.00	1	382.22	-5.54	-1.01	1.35
Average					668.47			-0.15	-0.47	0.21

The results show that our ALNS is able to solve all small instances very efficiently. CPLEX found the optimal solution for 33 instances within the time limit and ALNS was able to obtain the optimal solution in all these instances. In RC201-10 and R102-15, we found the same upper bound as CPLEX and in RC204-15 our solution is 5.54% better than the upper bound provided by CPLEX. The average run time of ALNS is only 0.21 seconds whereas CPLEX spent 668 seconds on the average. It should be noted that the run time of CPLEX is not the time it takes until it finds the optimal solution or the upper bound but the time it stops after the optimality conditions have been satisfied or the time limit has been reached.

When we compare the solutions that we found with the PR scheme to the optimal solutions of the FR scheme, we see that the number of EVs utilized remained same and the routes were improved only in seven instances. We believe that the potential advantage of the PR strategy is not evident in these instances because their sizes are too restrictive to allow alternate routes. Nevertheless, the savings in total distance may exceed 10% as we observed in C103-15.

2.5.3.2. Numerical Results for Large-Size Instances

In our experimental study with the large instances we also implement a strategy where the partial recharge is allowed at a pre-determined constant amount and test different amounts for comparison purposes. In that case, the implementation of SI algorithms is same, but the PR is performed at a constant level. If the battery is not sufficient to complete the route, we fully recharge the battery at the last station visited before the infeasibility occurs, if the time windows of the customers permit.

The results are illustrated in Table 2.4. The results obtained by the FR scheme are given in the 'FR' column for comparison. The next three main columns report the best solutions found by ALNS using different PR strategies. '*q free*' refers to the case where the PR is performed at any level, i.e. the recharge amount is a decision variable, whereas in 'q=0.3', 'q=0.4', and 'q=0.5' the PR is allowed at the constant levels of 30%, 40%, and 50% of the battery capacity, respectively. Note that in some instances, the results obtained using the PR scheme are worse than the best results found with the FR scheme. For consistency, we preferred to give them as reported by ALNS even though the solutions to EVRPTW are also feasible to EVRPTW-PR.

These results clearly show the advantage of using the PR strategy over the FR restriction: when PR quantity is not fixed (q free) the average total distance reduces by 1.64% in the instances for which the numbers of vehicles used with FR and PR schemes are the same. On the other hand, the reductions are 0.80%, 0.91%, and 0.73% if PR is performed at the level of q=0.3, q=0.4, and q=0.5, respectively. In addition, we observe that solutions with one less vehicle are found in nine instances when q is free while the same situation occurs in six instances for q=0.3 and in three instances for q=0.4 and q=0.5. Moreover, we see PR has more potential for saving from the number of vehicles in type-1 problems, which is an expected outcome since those problems are more restrictive due to narrow time-windows and shorter route durations. Due to

Table 2.4. EVRPTW-PR results for different recharge strategies

	FR		PR (*q* free)			PR (*q*=0.3)			PR (*q*=0.4)			PR (*q*=0.5)		
Inst.	*#Veh*	*TD*	*#Veh*	*TD*	Δ%	*#Veh*	*TD*	Δ%	*#Veh*	*TD*	Δ%	*#Veh*	*TD*	Δ%
c101	12	1053.83	12	1051.23	-0.25	12	1043.38	-1.00	12	1045.98	-0.75	12	1053.82	0.00
c102	11	1056.12	11	1034.24	-2.12	11	1034.48	-2.09	11	1032.49	-2.29	11	1040.34	-1.52
c103	11	1001.81	**10**	973.39	-	**10**	981.16	-	**10**	1028.32	-	**10**	1030.42	-
c104	10	951.57	10	886.72	-7.31	10	891.05	-6.79	10	892.84	-6.58	10	894.03	-6.44
c105	11	1075.37	11	1037.78	-3.62	11	1051.98	-2.22	11	1051.98	-2.22	11	1054.36	-1.99
c106	11	1057.65	11	1024.18	-3.27	11	1030.41	-2.64	11	1036.88	-2.00	11	1046.57	-1.06
c107	11	1031.56	**10**	1058.11	-	11	1014.45	-1.69	11	1013.63	-1.77	11	1013.63	-1.77
c108	11	1015.68	**10**	1033.50	-	11	999.55	-1.61	11	1002.54	-1.31	11	1000.62	-1.51
c109	10	1069.16	10	960.03	-11.37	10	990.53	-7.94	10	947.20	-12.88	10	946.84	-12.92
c201	4	645.16	4	629.95	-2.41	4	642.84	-0.36	4	642.84	-0.36	4	641.13	-0.63
c202	4	645.16	4	629.95	-2.41	4	641.07	-0.64	4	641.07	-0.64	4	641.07	-0.64
c203	4	644.98	4	629.95	-2.39	4	641.07	-0.61	4	641.07	-0.61	4	641.07	-0.61
c204	4	636.43	4	629.95	-1.03	4	635.80	-0.10	4	636.43	0.00	4	636.43	0.00
c205	4	641.13	4	629.95	-1.77	4	638.17	-0.46	4	631.26	-1.56	4	638.17	-0.46
c206	4	638.17	4	629.95	-1.30	4	641.13	0.46	4	631.26	-1.09	4	638.17	0.00
c207	4	638.17	4	629.95	-1.30	4	641.13	0.46	4	638.17	0.00	4	638.17	0.00
c208	4	638.17	4	629.95	-1.30	4	634.19	-0.63	4	631.26	-1.09	4	638.17	0.00
r101	18	1679.06	18	1661.33	-1.07	18	1636.69	-2.59	18	1651.80	-1.65	18	1679.27	0.01
r102	16	1519.80	16	1461.48	-3.99	16	1466.58	-3.63	16	1461.38	-4.00	16	1467.92	-3.53
r103	13	1312.50	13	1262.75	-3.94	14	1254.22	-	13	1265.90	-3.68	13	1276.80	-2.80
r104	12	1071.89	**11**	1078.99	-	**11**	1131.39	-	12	1065.14	-0.63	12	1065.17	-0.63
r105	15	1383.29	15	1373.94	-0.68	15	1380.62	-0.19	15	1380.44	-0.21	15	1396.79	0.97
r106	14	1276.15	**13**	1310.46	-	14	1288.90	0.99	14	1295.60	1.50	14	1284.19	0.63
r107	12	1148.43	12	1118.91	-2.64	12	1132.35	-1.42	12	1131.01	-1.54	12	1126.42	-1.95
r108	11	1051.59	11	1031.14	-1.98	11	1045.97	-0.54	11	1044.82	-0.65	11	1048.49	-0.30
r109	13	1214.72	13	1201.04	-1.14	13	1193.76	-1.76	13	1209.30	-0.45	13	1194.80	-1.67
r110	12	1097.89	**11**	1112.80	-	**11**	1090.92	-	12	1095.98	-0.17	12	1093.73	-0.38
r111	12	1109.14	12	1084.13	-2.31	12	1102.07	-0.64	12	1096.22	-1.18	12	1101.27	-0.71
r112	11	1038.74	11	1017.31	-2.11	11	1035.16	-0.35	11	1017.52	-2.09	11	1037.90	-0.08
r201	3	1265.67	3	1266.06*	0.03	3	1266.54	0.07	3	1274.00	0.65	3	1262.10	-0.28
r202	3	1052.32	3	1052.32	0.00	3	1054.70	0.23	3	1053.57	0.12	3	1055.48	0.30
r203	3	895.54	3	895.54	0.00	3	896.71	0.13	3	895.83	0.03	3	895.70	0.02
r204	2	780.98	2	780.14	-0.11	3	720.15	-	3	723.08	-	2	801.29	2.53
r205	3	987.36	3	987.36	0.00	3	989.03	0.17	3	995.03	0.77	3	991.52	0.42
r206	3	922.70	3	922.70	0.00	3	925.40	0.29	3	934.88	1.30	3	925.67	0.32
r207	2	847.14	2	846.59	-0.06	2	853.12	0.70	3	811.63	-	2	849.49	0.28
r208	2	736.12	2	736.12	0.00	2	736.75	0.09	2	736.75	0.09	2	737.40	0.17
r209	3	871.22	3	868.95	-0.26	3	871.31	0.01	3	872.03	0.09	3	871.55	0.04
r210	3	843.65	3	843.36	-0.03	3	850.84	0.84	3	849.15	0.65	3	850.16	0.77
r211	3	761.56	**2**	862.56	-	3	761.56	0.00	3	766.56	0.65	3	766.56	0.65
rc101	16	1731.07	16	1684.84	-2.74	**15**	1743.90	-	16	1689.40	-2.47	16	1704.19	-1.58
rc102	15	1551.69	**14**	1155.50	-	**14**	1566.40	-	**14**	1555.90	-	**14**	1558.51	-
rc103	13	1351.73	13	1329.58	-1.67	13	1351.51	-0.02	13	1342.49	-0.69	13	1345.04	-0.50
rc104	11	1232.45	11	1202.93	-2.45	11	1226.33	-0.50	11	1229.38	-0.25	11	1223.79	-0.71
rc105	14	1473.24	14	1458.49	-1.01	14	1458.28	-1.03	14	1467.44	-0.40	14	1470.40	-0.19
rc106	14	1414.99	**13**	1422.96	-	**13**	1440.45	-	**13**	1455.21	-	**13**	1417.40	-
rc107	12	1283.05	12	1261.03	-1.75	12	1272.09	-0.86	12	1265.16	-1.41	12	1261.44	-1.71
rc108	11	1209.11	11	1185.68	-1.98	11	1184.06	-2.12	11	1235.85	2.16	11	1209.64	0.04
rc201	4	1446.84	4	1446.84	0.00	4	1454.11	0.50	4	1463.70	1.15	4	1456.02	0.63
rc202	3	1450.34	3	1416.96	-2.36	4	1244.76	-	4	1245.10	-	4	1242.59	-
rc203	3	1069.27	3	1069.27	0.00	3	1089.74	1.88	3	1092.22	2.10	3	1084.06	1.36
rc204	3	887.45	3	887.76	0.04	3	887.29	-0.02	3	892.25	0.54	3	886.23	-0.14
rc205	3	1277.60	3	1262.22	-1.22	4	1162.16	-	4	1145.34	-	4	1151.93	-
rc206	3	1207.64	3	1213.89	0.51	3	1206.09	-0.13	3	1208.36	0.06	3	1209.42	0.15
rc207	3	994.48	3	993.49	-0.10	3	996.35	0.19	3	992.14	-0.24	3	993.26	-0.12
rc208	3	841.34	3	839.71	-0.19	3	847.82	0.76	3	843.98	0.31	3	844.76	0.40
Avg					-1.64			-0.80			-0.91			-0.73

* 1258.39 was also observed during our tests

the same fact, we also observe that the average distances improve significantly in type-1 instances, c1 problem set in particular. When q is free the average improvement in total distance is 2.89% in type-1 problems whereas the average improvement in type-2 problems is only 0.66%.

Not surprisingly ALNS using the variable PR amount has a better performance compared to ALNS with constant PR. On the other hand, its average computation time is 9.7% longer compared to the case when q=0.3 (16.77 minutes vs. 15.29 minutes). The difference in run time is particularly significant in r2 and rc2 problems where ALNS with variable PR takes 25.1% and 10.3% more time, respectively. Since the time-windows are wider and routes are longer in these problem types, ALNS needs more time to evaluate the insertions and determine the recharge quantity. Nevertheless, the additional time brings substantial improvements both in the number of vehicles and total distance traveled. Overall, these results suggest the PR scheme is effective, particularly in cases where the time windows are more restrictive.

2.6. Conclusions and Future Research

In this chapter, we investigated the partial recharge strategies for the EVRPTW, namely EVRPTW-PR, and proposed an ALNS algorithm to solve it. Some of the existing ALNS mechanisms were adopted from the literature whereas new removal and insertion mechanisms specific to the EVRPTW were developed to handle the visits to recharging stations and to incorporate the PR decisions.

We used the instances generated by Schneider et al. (2014) to validate the performance of the proposed ALNS. We first solved the EVRPTW instances and compared our results to those reported in Schneider et al. (2014), Goeke and Schneider (2015), and Hiermann et al. (2015). We also reported four new best-known solutions. For the proposed EVRPTW-PR we solved the same instances. The results revealed that the routes can be significantly improved when PR is allowed, even if at a pre-determined constant level.

In this study, we only allowed PR using the same power level. The problem can be extended to a multiple recharge power options at different speeds and costs as discussed in Felipe et al. (2014). Further research on this topic may also address the heterogeneous fleet case. The heterogeneity within this context does not only arise from the vehicle capacities but from their batteries as well since the cruising range of EVs and discharge/recharge durations differ depending on their battery condition and age. Furthermore, the travel times may vary due to traffic conditions, accidents, construction, etc., which may have a significant impact on the routing decisions due to limited driving range of the EVs. In addition, we assume that recharging stations are always available, which may not be true in real life and there may be queues in the stations. So, variability in both travel and recharging times arises as an interesting and challenging topic to be investigated within the stochastic context.

Chapter 3

A Matheuristic Method for the Electric Vehicle Routing Problem with Time Windows and Fast Chargers

3.1. Introduction

Transportation systems have a major impact on global energy consumption and CO_2 emissions with a share of around 20-25%. In the US, 26% of the total GHG emissions in 2014 was generated by transportation systems that utilize fossil fuels (www.epa.gov). Furthermore, 74% of the domestic freight in 2012 was moved by trucks and the freight volume is expected to grow by 39% in 2040 (Bureau of Transportation Statistics, 2014). Similarly, the EU reported that transportation was a main contributor with 23.2% of total GHG emissions in 2014 and freight transport activity is predicted to grow by around 80% in 2050 compared to 1990 (ec.europa.eu).

Transportation will remain a major and growing source of GHGs in the future. Hence, governments initiated new environmental measures and targets for reducing emissions and cutting the dependency on fossil fuels. For instance, US government targets reducing GHG emissions 20% below 2008 levels by 2020 (www.state.gov). The EU aims a reduction of 80-95% by 2050 with respect to 1990 (White Paper on Transport, 2011). In December 2016, a commitment was signed by 194 countries in New York to set a global action to stop the global temperature rise (unfccc.int). Since transportation plays a major part in GHG emissions and road transport contributes with a 75% share, the new regulations bring limitations to the use of ICEVs. In the EU, the use of ICEVs will be reduced by 50% in urban transport by 2030 and phased out by 2050. City logistics in major European urban centers will be CO_2-free by 2030 (White Paper on Transport, 2011). The parliaments of Netherlands and Norway recently passed new motions that will end sales of new cars powered by fossil fuels after 2025 (Edelstein, 2016).

Similarly, German Federal Council accepted a resolution that bans the sales of fossil fuel cars by 2030 (Khan, 2017).

The targets set by governments and the new regulations encourage the usage of AFVs such as solar, electric, biodiesel, LNG, CNG vehicles. Many municipalities, government agencies, non-profit organizations, and private companies are converting their fleets to include AFVs, either to reduce their environmental impact voluntarily or to meet new environmental regulations (Erdoğan and Miller-Hooks, 2012). Consequently, the advancements in the EV technology have gained momentum in parallel with the growing environmental concerns in societies.

EVRP is an extension to the Capacitated Vehicle Routing Problem (CVRP) where a fleet of EVs is used instead of ICEVs. The energy stored in the battery is consumed along the journey proportional to the distance travelled and the EV may need recharging to complete its tour. Recharging may be performed at any battery SoC. The stations are scarce and recharging may require a significant amount of time, compared to short refueling times at petrol stations. In this chapter, we address the EVRPTW which was firstly introduced by Schneider et al. (2014). EVRPTW assumes that recharging time is a linear function of the energy transferred and the battery is fully charged. Bruglieri et al. (2015) relaxed the full charge restriction and allowed partial recharging with any quantity up to the battery capacity, which is the current practice in real-world applications.

In this study, we extend the EVRPTW-PR by introducing fast charging option and refer to this problem as EVRPTW and Fast Charging (EVRPTW-FC). Basically, we assume that the stations are equipped with multiple charger types. They vary in power supply, power voltage, and maximum current options, which affect the recharge duration. We formulate this problem as a 0-1 mixed integer linear program and propose a matheuristic approach to solve it efficiently. Our approach combines the ALNS with an exact method. At each iteration of the ALNS, the feasible solution is destroyed by removing certain customers and stations from their routes and then repaired by inserting the removed customers back to the solution along with stations when recharging is necessary. When a station is inserted, the charger type and recharge quantity are also determined. The solution found by ALNS is then improved periodically by solving a mixed linear integer program which optimizes the decisions associated with recharging stations, charger types, and recharge quantities given the sequence of the customers visited.

The main contributions of this chapter can be summarized as follows:

- We extend EVRPTW-PR to allow fast charging using multiple charging equipment types, and present two mathematical programming formulations of EVRPTW-FC.
- As a solution methodology, we develop a matheuristic approach which combines ALNS with an exact method.
- The proposed ALNS involves new destroy and repair mechanisms specific to the nature of the problem.
- For a given sequence of customers, we propose a novel formulation of the charging sub-problem that can be solved to optimality fast.
- We devise an extensive experimental design to validate the performance of the proposed methodology and to show the benefits of fast charging.

The remainder of the chapter is organized as follows. Section 3.2 reviews the relevant literature. Section 3.3 describes the problem and presents the mathematical models. The proposed solution approach is described in Section 3.4. Then, Section 3.5 presents the computational study and provides the numerical results. Finally, the chapter closes with concluding remarks in Section 3.6.

3.2. Related Literature

VRPs with AFVs context have been studied by several researchers in recent years. Schneider et al. (2014) introduce EVRPTW as an extension to GVRP. Bruglieri et al. (2015) and Chapter 2 relaxes the full recharge restriction and allow batteries to be recharged up to any level. The former minimizes the number of vehicles, travel time, waiting time, and recharging time, develops a Variable Neighborhood Search Branching method, and uses it to solve small size instances. The latter extends the model of Schneider et al. (2014) to formulate EVRPTW with Partial Recharges and proposes an ALNS approach that improves some of the best-known results in the literature. Bruglieri et al. (2016) formulate a more effective mathematical model for GVRP by reducing the number of variables and eliminating dominated stations for each pair of customers.

Montoya et al. (2017) is the first study that extended EVRP to consider nonlinear charging functions. The objective function minimizes the total time which includes travel and charging time. The authors propose a hybrid metaheuristic to solve the problem and introduce new benchmark instances. New formulations of this problem are proposed in Froger et al. (2017).

Yang and Sun (2015) model location and routing decisions simultaneously for a capacitated EV fleet. They consider battery swap stations (BSSs) instead of recharging stations where the EVs always depart from stations with full battery. Hof et al. (2017) also investigate EVRP with BSSs and develop an Adaptive VNS (AVNS) approach to solve it. Recently, Schiffer and Walther (2017) propose a location-routing model for EVRPTW allowing partial recharging at customer locations. They analyze the effects of different objective functions. Paz et al. (2017) also model the same problem considering multi depots and battery swapping option in some locations.

Pelletier et al. (2017) integrate battery degradation into the model and optimize charging schedules at the depot. They also provide managerial insights considering degradation, grid restrictions, charging costs, and charging schedules of the fleet. A detailed survey of the goods distribution with EVs can be found in Pelletier et al. (2016) and Pelletier et al. (2017).

3.3. Problem Description and Model Formulation

3.3.1. Problem Description

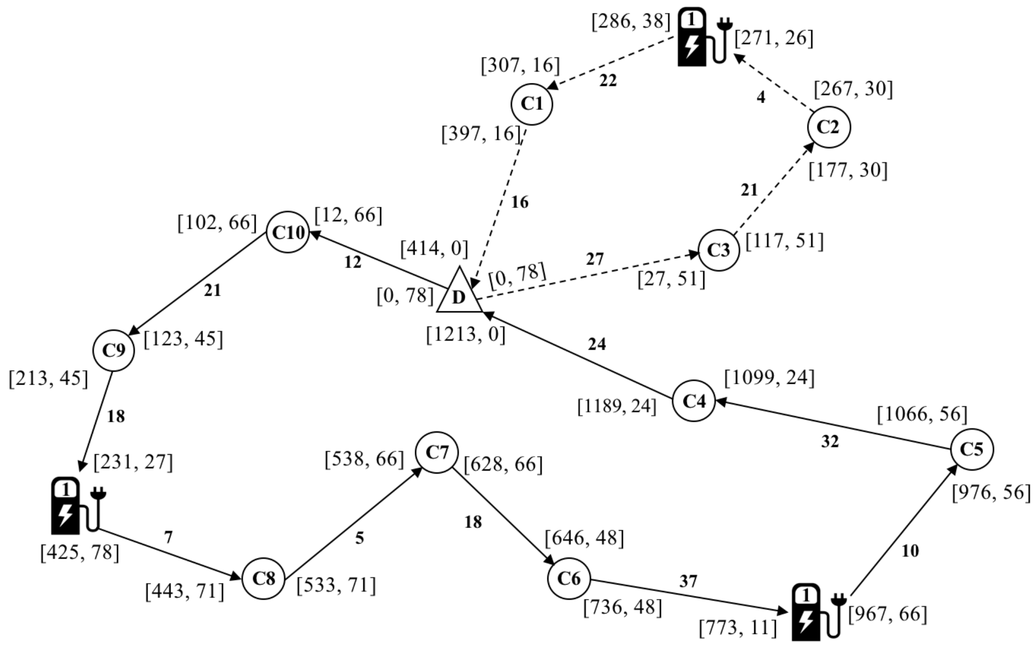
Given a homogeneous fleet of EVs, EVRPTW-FC aims to determine a set of routes involving customers with known demands, delivery time windows, service durations, and recharging stations with different types of chargers. The charging levels can be classified into three categories: Level 1 (1.4 kW to 1.9 kW), Level 2 (4 kW to 19.2 kW), and Level 3 (50 kW to 100 kW) (Yilmaz and Krein, 2013). In line with the current technology, we assume that every station is equipped with three types of chargers, which may be referred to as normal, fast, and super-fast charger, respectively. While recharging takes less time in fast and super-fast charging options, the unit cost of energy is higher since the installation of the chargers requires substantial electrical infrastructure and the equipment is more expensive. The charge durations are linear with respect to time at the first phase of charging which corresponds to almost full battery while the second phase is non-linear and can take hours to obtain a fully charged battery (Montoya et al., 2017). On the other hand, it is a common industrial practice to operate within the first phase because recharging the battery up to full capacity can adversely affect its lifespan (Sweda et al., 2017). So, without loss of generality we assume linear recharging times in this study. In addition, we allow only one recharge between two consecutive customers which is the realistic situation within the context of urban logistics. Our objective function is hierarchical where minimizing the number of vehicles is the primary objective while minimizing the total cost of energy consumed is the secondary.

To describe the general setting and highlight the advantage of using fast chargers we employ the instance c104c10-s3 of Schneider et al. (2014) which involves ten customers and three stations. The problem is solved using CPLEX and the optimal solution is illustrated in Figure 3.1. The recharging stations are represented with charger icons. The numbers on the icons refer to level 1 (normal), level 2 (fast), and level 3 (super-fast) chargers. The cargo and battery capacities of the vehicles are 200 and 77.75 units, respectively. The vehicles travel one unit distance in one unit of time consuming one unit of energy. Normal, fast, and super-fast chargers transfer one unit of energy in 3.47, 0.62, and 0.28 time units, respectively, at the cost of 1, 1.1, and 1.2 units, respectively. The demands and time windows are provided in Table 3.1. The values given in brackets placed at the beginning (end) of an arc represent the time and battery states of charge, respectively, at departure (arrival) from (at) the corresponding customer. The arc distances are shown in bold. For ease of understanding, we rounded all the values in the figure to the nearest integer whereas the objective function values are exact.

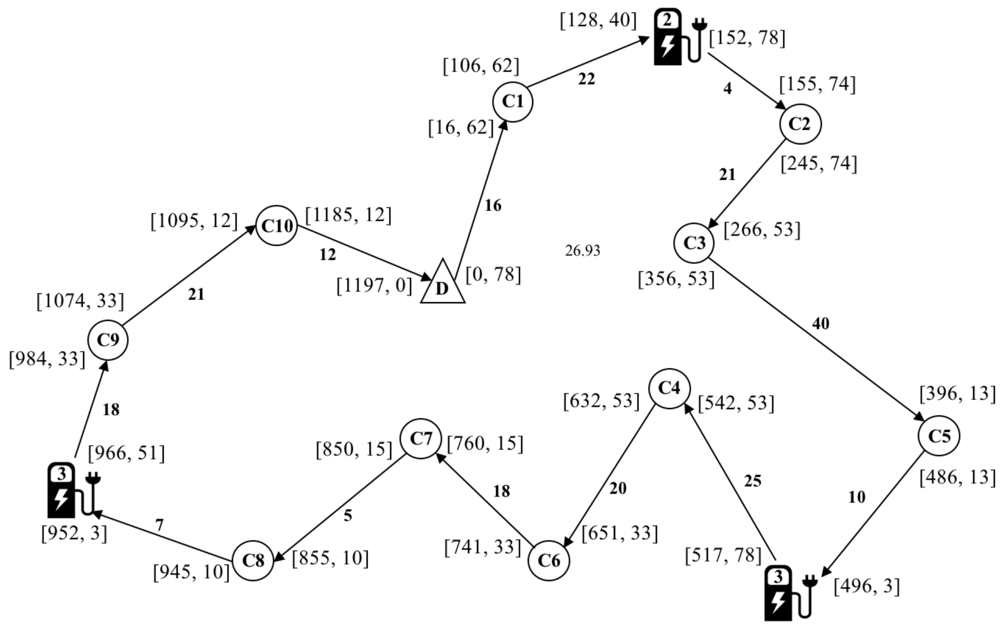
Table 3.1. Demand and time-window data for the example illustrated in Figure 3.1

	Depot	Customers									
	D	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Demand	-	10	10	30	10	10	40	20	10	20	20
Early service time	0	0	177	0	0	0	0	0	0	0	0
Late service time	1236	1129	243	1119	1122	1094	1111	1126	1122	1113	1133

In Figure 3.1.a, we see the optimal solution of the problem when only normal chargers are available at the stations. Two vehicles travel a total distance of 273.93 units at a total energy cost of 273.93. The two values coincide because the problem involves only normal chargers and one unit of energy is consumed to travel one unit of distance. On the other hand, Figure 3.1.b shows that all customers can be served by only one vehicle in a single tour when fast and super-fast chargers are available. This is due to the reduced recharging times at stations which enable the EV to catch the time windows of all customers. In this case, the total distance travelled is 239.13 and total cost of energy is 267.60. Although fast and super-fast recharges are more expensive, the total cost of energy decreases because the EV makes a shorter trip consuming less energy.



(a) Optimal solution using single charger (Total cost=273.93)



(b) Optimal solution using multiple chargers (Total cost=267.60)

Figure 3.1. Route plans when each recharging station is equipped with (a) only normal chargers, (b) normal, fast and super-fast chargers

3.3.2. Problem Formulation

We follow the notation used in Chapter 2 for ease of understanding. Let V and F denote the set of customers and the set of recharging stations, respectively. Since recharging stations may be visited multiple times by the same vehicle or different vehicles, we create sufficient number of copies and allow at most one visit to each in the mathematical model. So, we define F' as the set of all recharging stations along with their copies and $V' = V \cup F'$. We assume that each station is equipped with all types of chargers but only one is used at each visit to the station. This assumption can be easily relaxed but may not be practical in the real business environment. We also assume that all EVs are recharged with normal (cheapest) charger type over night at the depot and depart in the morning with full battery. To keep track of each EV's energy consumption along its tour we create copies of the depot by defining sets DD and AD as the departure depot and arrival depot vertices, respectively. We define $V_{DD} = V \cup DD$ and $V_{AD} = V \cup AD$. Let $V'_{DD} = V' \cup DD$, $V'_{AD} = V' \cup AD$, and $F'_{DD} = F' \cup DD$. Then, the problem can be represented by a complete graph $G = (V'_{DD,AD}, A)$ where $A = \{(i,j) | i \in V'_{DD}, j \in V_{AD}, i \neq j\}$ and $V'_{DD,AD} = V' \cup AD \cup DD$.

Each arc is associated with distance d_{ij} and travel time t_{ij}. The energy is consumed at a rate of h and the battery is discharged by $h.d_{ij}$ when the vehicle traverses arc (i,j). Each customer $i \in V$ is associated with demand q_i, service time s_i, and time window $[e_i, l_i]$. The fleet is homogeneous and consists of vehicles with cargo capacity C and battery capacity Q. The continuous decision variables τ_i, u_i, and y_i keep track of the service start time, remaining cargo level, and remaining battery level upon arrival to each vertex, respectively. Y_i keeps track of the battery SoC at the departure from either the depot or a station. Finally, the binary variable x_{ij} takes the value 1 if arc (i,j) is traversed and 0, otherwise. The mathematical notation is given in Table 3.2.

Table 3.2. Mathematical notation

Sets:	
V	Set of customers
F	Set of recharging stations
F'	Set of recharging stations with their copies
V'	Set of customers and stations with their copies ($V \cup F'$)
DD	Set of departure depots
AD	Set of arrival depots
V_{DD}	Set of customers and departure depots ($V \cup DD$)

Sets:	
V_{AD}	Set of customers and arrival depots ($V \cup AD$)
V'_{DD}	Set of customers, departure depots, and stations with their copies ($V' \cup DD$)
V'_{AD}	Set of customers, arrival depots, and stations with copies ($V' \cup AD$)
F'_{DD}	Set of departure depots and stations with their copies ($F' \cup DD$)
$V'_{DD,AD}$	Set of customers, arrival/departure depots, and stations with their copies ($V' \cup AD \cup DD$)
Parameters:	
d_{ij}	Distance from node *i* to node *j*
t_{ij}	Travel time from node *i* to node *j*
q_i	Demand of customer *i*
s_i	Time required to serve customer *i*
$[e_i, l_i]$	Service time window of customer *i*
C	Cargo capacity of the vehicles
Q	Battery capacity of the vehicles
Decision variables:	
τ_i	Service starting time at node *i*
u_i	Remaining cargo level at node *i*
y_i	Battery SoC at the arrival at node *i*
Y_i	Battery SoC at the departure from node *i*
x_{ij}	1 if the vehicle traverses arc (*i, j*); 0 otherwise
a_i	1 if the vehicle is recharged with normal charger at station *i*; 0 otherwise
b_i	1 if the vehicle is recharged with fast charger at station *i*; 0 otherwise
θ_i^m	Amount of energy recharged at station *i* using charger type *m*

In what follows, we present two alternative 0-1 mixed integer linear programming formulations of the problem.

3.3.2.1. Model 1

In this model, we define binary variables a_i and b_i to determine which charging equipment is used to recharge the vehicle at station $i \in F'$: $a_i = 1$ if charger type 1 (normal) is used, $b_i = 1$ if type 2 (fast) is used, and $a_i = b_i = 0$ if the charger type is 3 (super-fast). The battery recharging rate and unit energy cost depend on the charger type $m \in M$ and are referred to as g^m and c^m, respectively. Since we consider three charger types, $M = \{1,2,3\}$ and $m = 1$ corresponds to the normal (slowest) charger whereas $m = 3$ represents the super-fast (fastest) charger. Then, the problem is formulated as follows:

$$\min \quad \sum_{i \epsilon F'} \sum_{m \epsilon M} c^m \theta_i^m + c^1 \left(Q \sum_{i \epsilon DD} \sum_{j \epsilon V'} x_{ij} - \sum_{i \epsilon AD} y_i \right) \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j \epsilon V_{AD}} x_{ij} \leq 1 \qquad \forall i \in F' \tag{3.2}$$

$$\sum_{j \epsilon V'_{AD}} x_{ij} = 1 \qquad \forall i \in V \tag{3.3}$$

$$\sum_{i \epsilon V'_{DD}} x_{ij} = \sum_{i \epsilon V'_{AD}} x_{ji} \qquad \forall j \in V' \tag{3.4}$$

$$\sum_{j \epsilon V'} x_{ij} \leq 1 \qquad \forall i \in DD \tag{3.5}$$

$$\sum_{i \epsilon V'} x_{ij} \leq 1 \qquad \forall j \in AD \tag{3.6}$$

$$\sum_{i \epsilon DD} \sum_{j \epsilon V'} x_{ij} = \sum_{i \epsilon AD} \sum_{j \epsilon V'} x_{ji} \tag{3.7}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0 (1 - x_{ij}) \leq \tau_j \qquad \forall i \in V_{DD}, \forall j \in V'_{AD} \tag{3.8}$$

$$\tau_i + t_{ij} x_{ij} + \sum_{m \in M} g^m \theta_i^m - (l_0 + g^1 Q)(1 - x_{ij}) \leq \tau_j \qquad \forall i \in F', \forall j \in V_{AD} \tag{3.9}$$

$$e_j \leq \tau_j \leq l_j \qquad \forall j \in V'_{DD,AD} \tag{3.10}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}) \qquad \forall i \in V'_{DD}, \forall j \in V'_{AD} \tag{3.11}$$

$$0 \leq u_i \leq C \qquad \forall i \in DD \tag{3.12}$$

$$0 \leq y_j \leq y_i - (h.d_{ij}) x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in V, \forall j \in V'_{AD} \tag{3.13}$$

$$0 \leq y_j \leq Y_i - (h.d_{ij}) x_{ij} + Q(1 - x_{ij}) \qquad \forall i \in F'_{DD}, \forall j \in V_{AD} \tag{3.14}$$

$$0 \leq y_i \leq Y_i \leq Q \qquad \forall i \in F'_{DD} \tag{3.15}$$

$$Y_i = Q \qquad \forall i \in DD \tag{3.16}$$

$$Y_i - y_i = \sum_{m \in M} \theta_i^m \qquad \forall i \in F' \tag{3.17}$$

$$0 \leq \theta_i^1 \leq Q a_i \qquad \forall i \in F' \tag{3.18}$$

$$0 \leq \theta_i^2 \leq Q b_i \qquad \forall i \in F' \tag{3.19}$$

$$0 \leq \theta_i^3 \leq Q(1 - a_i - b_i) \qquad \forall i \in F' \tag{3.20}$$

$$a_i, b_i \in \{0,1\} \qquad \forall i \in F' \tag{3.21}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.22}$$

The objective function (3.1) minimizes the total energy cost which consists of two terms. The first term corresponds to the total cost of energy recharged along the route. The second is the total cost of energy. All vehicles are recharged fully using the cheapest (slowest) charger type at the depot overnight. Note that the second term includes the battery SoC at the end of the trip. More specifically, the value of the remaining energy is deducted from the total cost since that amount of energy has not been consumed en-route. Constraints (3.2)−(3.4) are the connectivity constraints which ensure that each customer is visited exactly once, and each recharging station may be visited at most once. Constraints (3.5) and (3.6) keep track of departures from and arrivals at the depots. Constraints (3.7) guarantee that all EVs departed from the depot arrive at the depot at the end of their tour. The start times to service are controlled by constraints (3.8)−(3.10). Constraints (3.11) and (3.12) observe the load on the vehicle and make sure that total load does not exceed the cargo capacity. Constraints (3.13) and (3.14) keep track of battery SoC when departing from customers and stations, respectively. Constraints (3.15) define the bounds for variables y_i and Y_i while constraints (3.16) ensure that EVs depart from the depot with full battery. Constraints (3.17) determine the amount of energy transferred while constraints (3.18)−(3.20) control which charger type is utilized for recharging. Note that a_i and b_i cannot be 1 simultaneously because of nonnegativity of θ_i^3 variables. Finally, constraints (3.21) and (3.22) define the binary decision variables.

3.3.2.2. Model 2

The second model is a simple modification of EVRPTW-PR formulation and was first presented in Çatay and Keskin (2017). Here, instead of defining a_i and b_i variables to determine the charger type utilized at the station, we use three copies of each station where each copy represents a different charger type. In other words, each recharging station is equipped with only one charger type, but we have three stations at the same location. Thus, the total number of stations increases three-fold. So, F includes all these stations and F' is the set of stations and their copies to allow multiple visits to each station. Let g_i and c_i be the recharging rate and unit energy cost for station $i \in F'$, respectively. g_0 is the recharging rate of the slowest charger and c_0 is the associated unit cost. Then, the mathematical model is formulated as follows:

$$minimize \sum_{i\epsilon F'} c_i(Y_i - y_i) + c_0 \left[Q \sum_{i\epsilon DD} \sum_{j\epsilon V'} x_{ij} - \sum_{i\epsilon AD} y_i \right] \tag{3.23}$$

s.t. (3.2)−(3.16) and (3.22)

The objective function (3.23) represents the same energy cost as (3.1) but with different terms. All constraints in Model 1 remain in Model 2 except constraints (3.17)−(3.21) which are associated with charger types.

3.3.2.3. Evaluation of Model 1 and Model 2

We compare the efficiency of models 1 and 2 by solving small instances of EVRPTW data set of Schneider et al. (2014). The data set includes three subsets of 12 problems, each involving 5, 10, and 15 customers. We used CPLEX 12.6.2 solver running on a single thread and the time limit is set to 7200 seconds. The experiments were carried on a workstation with Intel Xeon E5 3.30 GHz processor and 64 GB RAM.

Table 3.3. Comparison of the two models

	Model 2			Model 1			
#Cust	*#Opt*	*#NFS*	*AvgTime*	*#Opt*	*#NFS*	*AvgTime*	*#Better*
5	12	0	265	12	0	<1	0
10	7	0	3618	12	0	179	0
15	0	1	7200	3	0	5582	6

The results are summarized in Table 3.3. The column "*#Cust*" gives the number of customers in the problem set. "*#Opt*" refers to the number of optimal solutions found within the time limit whereas "*#NFS*" indicates the number of instances for which no feasible solution could be found. "*AvgTime*" is the average of the run times in seconds for each subset and "*#Better*" in the last column reports the number of instances in which Model 1 gives better solutions than Model 2. Since 5-customer instances are very small, CPLEX found the optimal solutions with both models; however, the solution times were significantly smaller with Model 1. For the 10-customer instances, CPLEX failed to prove the optimality of five problems within the time limit using Model 2 while all problems were solved to optimality with Model 1, again in substantially less time. Finally, for the 15-customer instances, three problems were solved optimally with Model 1 and none with Model 2. Furthermore, Model 2 could not yield a feasible solution in one problem. Overall, Model 1 provided the optimal solutions faster and in many instances, it provided better upper bounds when the time limit is reached. So, we decided to use the results obtained with Model 1 to benchmark our solution methodology that we will describe in the next section.

3.4. Description of the Matheuristic

For solving the EVRPTW-FC, we propose a two-phase matheuristic approach where in the first phase, we attempt to find good heuristic solutions using ALNS and then improve them using CPLEX in the second phase. In this phase, we resort to CPLEX solver but any open-source or commercial solver can be utilized instead. Matheuristics use mathematical models in a heuristic framework and they have been applied to various routing problems. We refer the interested reader to Archetti and Speranza (2014) for the details of the approach and an overview of implementations.

In our matheuristic approach, while ALNS explores the neighborhoods to find promising routes, after every Ω iterations we further enhance the current best solution by optimizing the charging decisions along the tour of an EV by fixing the sequence of customers visited. A similar problem was also solved by Montoya et al. (2017). They name this problem as Fixed-Route Vehicle-Charging Problem (FRVCP) after the Fixed-Route Vehicle-Refueling Problem (FRVRP) introduced by Suzuki (2014). Montoya et al. (2017) solve the EVRP by using a sequence-first split-second approach where they first construct a TSP tour and then split it to extract vehicle routes by ignoring the EV range limit. If any of the resulting routes is energy-infeasible, then they try to repair it by solving FRVCP. This approach cannot be implemented for solving EVRPTW because building vehicle routes without considering the recharge needs/durations of EVs and then trying to insert stations may cause many time-window violations. So, at each iteration of ALNS our repair procedure yields a feasible solution which is further improved by solving the mathematical programming with CPLEX.

For most of the ALNS destroy and repair mechanisms we resort to the neighborhoods utilized in Chapter 2. In addition, we propose new removal and insertion methods specific to the fast charging nature of the problem.

3.4.1. Removal Heuristics

Since the problem has two types of vertices, namely customers and recharging stations, their removal will have different impact on the solution. So, we employ separate customer removal (CR) and recharging station removal (SR) operators for destroying the solution.

3.4.1.1. Customer Removal

In addition to the well-known CR heuristics widely used in the literature such as Random, Worst-Distance, Worst-Time, Shaw, Proximity-based, Demand-based, Time-based, Zone, Random Route, and Greedy Route removals, we utilize Remove Customer with Preceding Station and Remove Customer with Succeeding Station operators introduced in Chapter 2 where customers are removed along with the station visited immediately before or after serving that customer. At each iteration, one of these CR operators is selected randomly to remove γ customers from the solution and put them in a removal list. The value of γ depends on the total number of customers and is determined randomly between $\underline{n}_c$ and $\bar{n}_c$ using a uniform distribution.

3.4.1.2. Station Removal

We use Random and Worst-Distance Station removals proposed in Chapter 2. In addition, we propose the following two new SR operators:

Least Used Station Removal: The motivation behind this heuristic is to reduce the cost of visiting recharging stations often. So, we attempt to eliminate unnecessary recharges and satisfy the energy needs of EVs by visiting less number of stations. This can be achieved by utilizing the visited stations to recharge the battery as much as possible instead of recharging small quantities with frequent visits. The operator lists the stations (chargers) in the non-decreasing order of the quantity of energy they charge and removes a pre-determined number of stations from the top of the list.

Expensive Station Removal: Our aim in this removal heuristic is to save from energy cost by eliminating unnecessary recharges using more expensive charging options. The operator lists the stations (chargers) in the non-increasing order of the cost they incur and removes a pre-determined number of stations from the top of the list.

In all the SR operators, σ recharging stations are removed from the solution after every N_{SR} iterations. σ is determined in a similar way as γ based on the number of stations visited in the current solution.

3.4.2. Insertion Heuristics

As in removal heuristics, different insertion mechanisms are designed for customers and recharging stations. A customer insertion (CI) mechanism is used after every CR operation whereas the station insertion (SI) follows only an SR operation.

3.4.2.1. Customer Insertion

We use the Greedy, Regret-2, Time-based, and Zone insertions as proposed in Chapter 2. In addition, we employ these mechanisms only with the fastest recharging option when a station insertion is needed to feasibly add a removed customer into a tour. We refer to these new operators as Fast Recharge (FR) Greedy, FR Regret-2, FR Time-based, and FR Zone insertions. Our aim is to shorten the charge durations which may allow serving more customers along the tour and thus reduce the number of vehicles. Eventually, the charge-related decisions will later be optimized in the second phase using a solver as described in Section 3.4.4.

3.4.2.2. Station Insertion

We adapt the Greedy and Best Station insertions introduced in Chapter 2 using the cost criterion as follows: when a recharging station is inserted in a route, first, we try the normal charger since it is the cheapest option. If the normal charge is infeasible due to its longer duration, we try fast and super-fast chargers consecutively. This procedure is repeated for all feasible stations and candidate stations are determined along with the charger type. Then, the insertion is performed according to the criteria used in the corresponding SI operator.

3.4.3. Constructing the Initial Solution

We use three initialization approaches for comparison. The first uses the best-known solutions reported Chapter 2. Basically, these solutions were obtained by only allowing normal recharge at the stations and can be considered as an upper bound for the fast recharge case. In the second approach, we implement the ALNS of Chapter 2 by allowing the super-fast recharge only and feed its solution to initiate the matheuristic. The last approach randomly puts all customers into the removal list and applies the FR Greedy CI heuristic. Henceforth, we will refer to these initialization approaches as IA 1, IA 2, and IA 3, respectively.

3.4.4. Route Enhancement

To improve the solution quality, we employ a post-optimization procedure systematically throughout the ALNS process. This procedure uses CPLEX to optimize the charge-related decisions along each EV route by fixing the sequence of the customers. These decisions include the locations of the stations, selection of the charger type, and the amount of energy transferred.

An easy way to solve this problem is to use Model 1 by reducing the customer set to include only those visited along the route of the vehicle considered and fixing their sequence. However, this formulation will be weak and solving it may require significant computation time particularly when the EV makes frequent stops. To overcome this drawback and speed-up the algorithm, we propose a tighter formulation which also eliminates the need for using copies of the stations. Our approach is similar to the ideas presented in Bruglieri et al. (2016). Let $\bar{V} = \{1, \dots, K\}$ be the set of customers served by the vehicle. 0 and $K+1$ represent the depot. We define $\bar{V}_0 = \bar{V} \cup \{0\}$ and $\bar{V}_{K+1} = \bar{V} \cup \{K+1\}$. Let $\theta_{i,i+1}^{m}$ denote the amount of energy recharged using charger type m if the EV visits a station on its way from customer i to customer $i+1$. As in Model 1, binary variables $a_{i,i+1}$ and $b_{i,i+1}$ are used to determine the charger type if the EV is recharged at a station between customers i and $i+1$. Note that these variables are defined for only consecutive customers and the number of $a_{i,i+1}$ and $b_{i,i+1}$ variables is the same as the number of arcs on the route. The mathematical model is formulated as follows:

$$\textit{minimize.} \quad \sum_{i\in F}\sum_{m\in M} c^m\theta_{i,i+1}^m + c^1(Q - y_{K+1}) \tag{3.24}$$

s.t.

$$\tau_i + s_i + \sum_{j\in F}(t_{ij} + t_{j,i+1})x_{ij} + \sum_{m\in M} g^m\theta_{i,i+1}^m \le \tau_{i+1} \qquad \forall i \in \bar{V}_0 \tag{3.25}$$

$$e_i \le \tau_i \le l_i \qquad \forall i \in \bar{V}_{K+1} \tag{3.26}$$

$$y_i - h\left[d_{i,i+1}\left(1 - \sum_{j\in F} x_{ij}\right) + \sum_{j\in F}\left(d_{ij} + d_{j,i+1}\right)x_{ij}\right] + \sum_{m\in M}\theta_{i,i+1}^m = y_{i+1} \qquad \forall i \in \bar{V}_0 \tag{3.27}$$

$$y_i - h\sum_{j\in F} d_{ij}x_{ij} \ge 0 \qquad \forall i \in \bar{V}_0 \tag{3.28}$$

$$\sum_{m\in M}\theta_{i,i+1}^m \le Q\sum_{j\in F} x_{ij} \qquad \forall i \in \bar{V}_0 \tag{3.29}$$

$$\sum_{m\in M}\theta_{i,i+1}^m \le Q - \left(y_i - h\sum_{j\in F} d_{ij}x_{ij}\right) \qquad \forall i \in \bar{V}_0 \tag{3.30}$$

$$0 \le \theta_{i,i+1}^1 \le Qa_{i,i+1} \qquad \forall i \in \bar{V}_0 \tag{3.31}$$

$$0 \le \theta_{i,i+1}^2 \le Qb_{i,i+1} \qquad \forall i \in \bar{V}_0 \tag{3.32}$$

$$0 \le \theta_{i,i+1}^3 \le Q(1 - a_{i,i+1} - b_{i,i+1}) \qquad \forall i \in \bar{V}_0 \tag{3.33}$$

$$y_0 = Q \tag{3.34}$$

$$a_{i,i+1}, b_{i,i+1} \in \{0,1\} \qquad \forall i \in \bar{V}_0 \tag{3.35}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in \bar{V}_0, j \in F \tag{3.36}$$

The objective function (3.24) minimizes total energy cost of the route. The first term represents the cost of recharges en-route while the second is the cost of energy, calculated based on the difference of SoCs between departure from and arrival at the depot. Constraints (3.25)−(3.26) satisfy time-window feasibility of the customers and the depot. Constraints (3.27) keep track of the battery SoC: if the EV does not visit a recharging station after leaving customer i, then SoC at arrival at customer $i+1$ is calculated by subtracting the energy consumed along the arc $(i, i+1)$ from SoC at customer i. If it visits a station, then the energy recharged at station j is added to the amount described in the previous case and the energy consumed along the arcs (i, j) and $(j, i+1)$ is subtracted. Constraints (3.28) make sure that if the EV visits a recharging station after customer i, SoC at the arrival at station j is nonnegative. Constraints (3.29) ensure that the recharging variables $\theta_{i,i+1}^m$ take a positive value only if the vehicle visits a station between customers i and $i+1$. Constraints (3.30) guarantee that SoC after the recharge does not exceed the battery capacity Q. Constraints (3.31)−(3.33) determine the charger type if the EV is recharged between i and $i+1$. Constraint (3.34) makes sure that the vehicle departs from the depot with a full battery while constraints (3.35)−(3.36) define the binary decision variables.

We also introduce a pre-processing procedure to reduce the set of the recharging stations as follows: Let i and $i+1$ be two consecutive customers in the route and $F_{i,i+1}$ be the set of recharging stations that the EV may visit when traveling from customer i to $i+1$. Initially, $F_{i,i+1} = F$. Then, we make a pairwise comparison of the stations with respect to their distance to customers i and $i+1$. For instance, consider two stations $j, j' \in F_{i,i+1}$. If $d_{ij'} > d_{ij}$ and $d_{j',i+1} > d_{j,i+1}$ then j is said to dominate j' and j' cannot be visited in the optimal solution since j is closer to both i and $i+1$. Hence, j' is removed from $F_{i,i+1}$. We repeat this procedure for all station pairs in $F_{i,i+1}$ to reduce its size. The same procedure is applied to all customer pairs $(i, i+1)$ in the route.

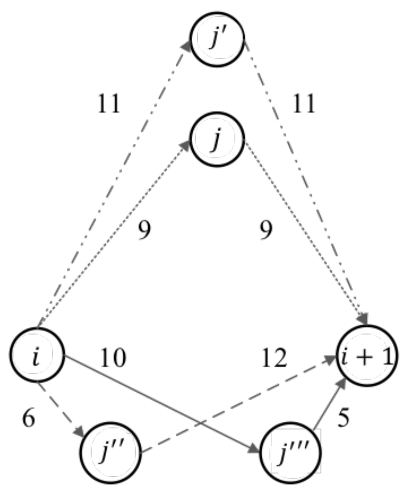


Figure 3.2. Set of recharging stations between customers i and $i+1$

Figure 3.2 illustrates the case of four recharging stations $j,\ j', j''$, and j''', which can be visited between customers i and $i+1$. It can be easily identified that j' is dominated by j as explained above. However, neither one of j, j'', and j''' dominates the other since both conditions are not satisfied. So, $F_{i,i+1}$ includes j, j'', and j'''.

In Figure 3.3, the structure of fixed-route problem is illustrated using a segment of the route consisting of three customer vertices. Since the pre-processing may eliminate some of stations that can be visited between a pair of customers, each arc is associated with a different recharging station set. So, we do not need to create any copies of the stations to allow multiple visits in the mathematical model. This decreases the number of decision variables significantly.

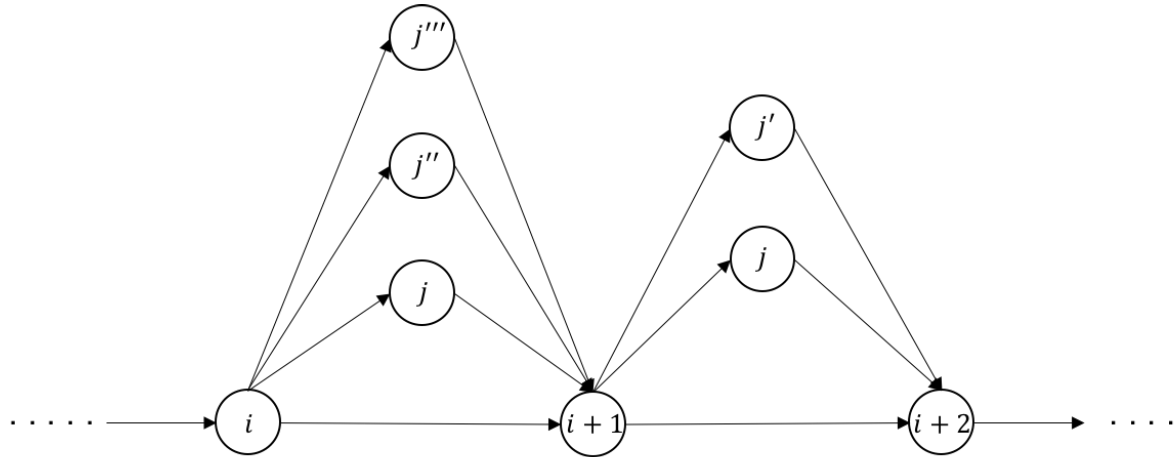


Figure 3.3. Station insertion between two nodes

3.4.5. Reducing the Number of Vehicles

Since the primary objective is to minimize the number of vehicles, we devote some iterations throughout the ALNS to this purpose. After every N_{RR} iterations we dedicate n_{RR} consecutive iterations to remove customers using the Random Route Removal or Greedy Route Removal operators and insert them using FR CI heuristics described in Section 4.2.1. Using only super-fast chargers when recharging is needed decreases the duration of the recharge and allow serving more customers along the route which may not be possible otherwise due to time-windows restrictions. This yields longer routes with more frequent visits to customers and thus offers an opportunity to reduce the total number of vehicles in the fleet.

The general structure of the proposed matheuristic is provided in Appendix B.

3.5. Experimental Design and Numerical Results

We test the performance of the proposed matheuristic on the EVRPTW data sets that Schneider et al. (2014). The data set of Felipe et al. (2014) is referred to as FORT instances and consists of two different configurations involving five and nine stations. Each configuration includes three sets of ten instances with 100, 200 and 400 customers distributed randomly. In total, the data set includes 60 instances.

In ALNS, we used the same parameter values as reported in Chapter 2. For the recharging speed and cost of different chargers, we used the values given in Felipe et al. (2014). In the optimization phase, we employed CPLEX 12.6.2 with its default setting using single thread. The matheuristic was implemented in Java programming language and the experiments are conducted on the same workstation described in Section 3.2.3.

3.5.1. Results for Large Instances

We first investigate how different initialization approaches and optimization frequencies affect the solution quality in order to determine the best configuration. Next, we examine the benefits of utilizing fast chargers in terms of fleet size and energy costs.

3.5.1.1. Analysis of Different Configurations

We optimize the recharging decisions of the best solution of that round every Ω iterations, which we refer to as CPLEX call frequency. On the one hand, choosing this number too small may increase the run time. On the other hand, choosing it too large may deteriorate the solution quality. Our preliminary experiments revealed that calling CPLEX after 200 and 500 iterations shows a good compromise. So, we decided to consider these values for further investigation.

Our stopping criterion is a limit on the number of iterations. For different initialization algorithms, we set different limits. We perform 25,000 iterations of ALNS when we utilize IA 1 and IA 2 to generate the initial solution. Then, we apply the matheuristic for 10,000 iterations. When we utilize IA 3 for initialization, the solution is constructed very fast by the greedy algorithm whereas the matheuristic is performed for 25,000 iterations. In other words, we allow a more intensive search during the initial solution generation in the former case whereas in the latter case matheuristic is the only actor and it benefits from the mathematical programming more rigorously.

Table 3.4. Comparison of results obtained with different configurations

	IA 1				IA 2				IA 3			
	$\Omega = 200$		$\Omega = 500$		$\Omega = 200$		$\Omega = 500$		$\Omega = 200$		$\Omega = 500$	
Instance	*#Veh*	*TC*	*#Veh*	*TC*	*#Veh*	*TC*	*#Veh*	*TC*	*#Veh*	*TC*	*#Veh*	*TC*
c101	**12**	**1043.38**	**12.00**	**1043.38**	**12.00**	**1043.38**	**12.00**	**1043.38**	12.00	1044.51	12.00	1044.51
c102	10	1078.12	10.00	1096.73	**10.00**	**1009.86**	**10.00**	**1009.86**	10.00	1080.95	10.00	1055.41
c103	10	962.78	10.00	962.78	**10.00**	**961.78**	**10.00**	**961.78**	10.00	992.25	10.00	997.27
c104	9	1113.85	10.00	878.81	**9.00**	**1006.12**	**9.00**	**1006.12**	10.00	893.66	10.00	900.29
c105	10	1102.47	10.00	1124.58	**10.00**	**1031.58**	**10.00**	**1031.58**	10.00	1141.21	10.00	1166.01
c106	10	1141.19	10.00	1082.12	**10.00**	**1044.96**	**10.00**	**1044.96**	10.00	1083.49	10.00	1166.84
c107	10	1017.80	10.00	1022.24	**10.00**	**1015.81**	**10.00**	**1015.81**	10.00	1146.72	10.00	1048.87
c108	10	1025.15	10.00	1025.15	**10.00**	**1022.36**	**10.00**	**1022.36**	10.00	1191.61	10.00	1074.91
c109	**10**	**940.38**	**10.00**	**940.38**	10.00	959.66	10.00	959.66	10.00	1126.70	10.00	1057.56
c201	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**
c202	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	4	641.13
c203	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	4	638.17	4	638.17
c204	3	746.75	3	787.77	**3**	**719.89**	3	720	3	741.99	3	846.81
c205	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**
c206	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**
c207	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	638.17	**4**	**629.95**
c208	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	**4**	**629.95**	4	630.06
r101	17	1772.39	17	1829.99	**16**	**1815.77**	17	1642.9	17	1845.09	17	1817.59
r102	15	1583.3	15	1561.71	**15**	**1436.95**	**15**	**1436.95**	15	1572.01	15	1585.49
r103	13	1256.95	12	1395	**12**	**1233.72**	**12**	**1233.72**	12	1423.86	12	1472.09
r104	11	1076.25	11	1076.25	**10**	**1048.69**	**10**	**1048.69**	11	1210.75	11	1261.95
r105	13	1515.67	13	1547.43	**13**	**1364.98**	13	1366.89	14	1549.83	14	1572.14
r106	12	1446.52	13	1300.36	**12**	**1280.16**	**12**	**1280.16**	12	1386.04	12	1435.11
r107	11	1159.07	11	1184.88	**10**	**1130.86**	**10**	**1130.86**	11	1261.46	11	1234.65
r108	10	1213.32	11	1030.48	**10**	**1010.68**	**10**	**1010.68**	10	1223.7	10	1168.6
r109	12	1359.11	12	1310.09	**11**	**1198.55**	**11**	**1198.55**	12	1433.76	12	1389.78
r110	11	1089.9	11	1089.9	**10**	**1127.7**	**10**	**1127.7**	11	1206.9	11	1336.29
r111	11	1180.33	11	1192.5	**11**	**1086.1**	**11**	**1086.1**	11	1325.5	11	1276.08
r112	11	1016.3	11	1016.3	**10**	**1015.49**	**10**	**1015.49**	11	1181.79	11	1168.92
r201	3	1262.06	3	1262.06	3	1295.95	**3**	**1257.50**	3	1576.58	3	1378.97
r202	**3**	**1051.46**	**3**	**1051.46**	3	1060.18	3	1060.18	3	1068.4	3	1097.62
r203	**3**	**895.54**	**3**	**895.54**	3	898.96	3	898.96	3	932.9	3	923.22
r204	**2**	**779.71**	2	780.13	2	785.7	2	785.7	2	816.47	3	728.46
r205	**3**	**987.36**	**3**	**987.36**	3	1001.85	3	1006.86	3	1020.8	3	1047.23
r206	**3**	**922.7**	**3**	**922.7**	3	928.56	3	928.56	3	956.58	3	974.17
r207	**2**	**846.43**	**2**	**846.43**	2	857.07	2	859.3	3	825.55	3	829.16
r208	**2**	**736.13**	**2**	**736.13**	2	737.43	2	739.64	2	739.15	2	747.66
r209	**3**	**866.67**	**3**	**866.67**	3	900.77	3	900.77	3	931.26	3	892.07
r210	**3**	**843.21**	**3**	**843.21**	3	856.76	3	859.13	3	879.16	3	871.9
r211	2	862.56	2	862.56	**2**	**840.61**	2	857.74	3	783.63	3	797.63
rc101	15	1744.85	15	1744.85	**14**	**1800.73**	15	1640.57	15	1827.93	15	1810.49
rc102	14	1526.31	14	1526.27	**13**	**1557.39**	**13**	**1557.39**	14	1657.06	13	1645.16
rc103	12	1389.5	12	1444.88	**12**	**1355.71**	**12**	**1355.71**	12	1497.5	12	1552.26
rc104	11	1201.04	11	1200.24	**10**	**1192.6**	10	1193.86	11	1257.48	11	1389.82
rc105	14	1449.53	13	1587.57	**13**	**1425.92**	**13**	**1425.92**	13	1488.58	13	1590.97
rc106	13	1402.95	13	1398.85	**12**	**1388.85**	**12**	**1388.85**	13	1541.1	13	1490.22
rc107	11	1294.2	11	1300.44	11	1249.03	**11**	**1247.87**	11	1366.75	11	1440.09
rc108	11	1182.84	11	1182.84	**10**	**1199.24**	**10**	**1199.24**	11	1302.43	11	1262.57
rc201	**4**	**1446.84**	**4**	**1446.84**	4	1485.23	4	1485.23	4	1482.53	4	1501.32
rc202	**3**	**1416.96**	**3**	**1416.96**	3	1426.88	3	1424.86	4	1287.24	4	1313.06
rc203	**3**	**1064.33**	**3**	**1064.33**	3	1081.57	3	1081.57	3	1144.73	3	1114.98
rc204	3	886.23	**3**	**886.19**	3	895.18	3	895.18	3	898.97	3	894.21
rc205	3	1257.92	3	1257.92	**3**	**1256.30**	**3**	**1256.30**	3	1341.91	3	1451.59
rc206	**3**	**1206.06**	**3**	**1206.06**	3	1229.67	3	1229.67	3	1223.51	3	1296.31
rc207	3	992.14	3	992.14	**3**	**991.65**	**3**	**991.65**	3	1073.96	3	1090.87
rc208	**3**	**839.71**	**3**	**839.71**	3	884.76	3	885.76	3	886.36	3	944.14
#Best		23		23		38		33		5		4

For each configuration, we performed 30 runs for each instance and reported the best results in Table 3.4. "*#Veh*" and "*TC*" represent the number of vehicles needed and total cost of energy, respectively. The best solutions among six different configurations are indicated in bold. The row *"#Best*" shows the total number of instances for which the corresponding configuration yielded the best solution.

We observe that the initialization approaches have a significant effect on the performance of the matheuristic. While IA 2 yields better solutions in type-1 instances where customers have narrow time windows, IA 1 performs better in type-2 instances which involve customers with wide time windows. In other words, determining the initial solution through ALNS by considering only normal chargers works better in type-2 problems whereas using the same initialization approach with super-fast chargers has a better performance in type-1 problems. Moreover, we also see that the superiority of IA 2 in type-1 problems is usually in terms of the number of vehicles while in type-2 problems IA 1 performs slightly better than IA 2 in total cost. The former is an expected outcome as the utilization of super-fast chargers may significantly cut down the recharge time at stations and allow the EV serve more customers along its route which will translate into a reduction in fleet size. However, the latter can be considered as a surprising result and we will further elaborate on this issue in the next section.

Table 3.5. Average run times of different configurations (in minutes)

	IA 1		IA 2		IA 3	
Data set	$\Omega = 200$	$\Omega = 500$	$\Omega = 200$	$\Omega = 500$	$\Omega = 200$	$\Omega = 500$
c1	4.89	4.85	2.20	2.18	5.49	4.60
c2	38.31	38.70	22.27	22.63	46.58	45.13
r1	3.73	3.67	1.26	1.23	2.89	2.83
r2	74.33	72.39	31.19	33.15	64.10	68.60
rc1	3.27	3.21	1.07	1.08	2.51	2.75
rc2	33.56	33.69	12.62	12.65	37.03	37.22
Average	26.35	26.08	11.77	12.15	26.43	26.85

When we examine the role of CPLEX call frequency on the solution quality we do not observe any substantial difference, yet $\Omega = 200$ performs slightly better than $\Omega = 500$, which is not surprising and supports the role of optimization in achieving higher quality solutions. Independent of the value of Ω, the results obtained by using IA 3 are inferior than those given by the other two initialization approaches. In other words, searching for a good initial solution pays back the effort spent.

In Table 3.5, we report the average run time of each problem subset to evaluate the computational effort required by different configurations. The numbers are in minutes. The results indicate that type-2 problems require more time than type-1 problems. This is well expected and in parallel with many studies that utilized Solomon (1987) data because wide time windows expand the search space bringing more feasible insertions to be evaluated and long planning horizon allows an EV make longer trips visiting more customers (more than 50 in certain instances) requiring more recharges. On the other hand, we observe that the run time of the implementation with IA 2 is significantly smaller than those with IA 1 and IA 3. Although IA 1 and IA 2 perform the same number of iterations using the same ALNS mechanisms, the station insertion procedure requires less computational effort in IA 2. This is due to the fact that the search for a feasible station for insertion takes more time when only normal chargers are available because of their longer recharging durations.

Regarding CPLEX call frequency, we do not see any major difference between the run times with $\Omega = 200$ and $\Omega = 500$ in any setting. This indicates that CPLEX does not require extensive computational effort to find the optimal solution for the fixed-route problem and validates the effectiveness of the proposed mathematical formulation. To further investigate the computational burden of route enhancement with CPLEX we illustrate the percentage of total computational time spent by ALNS and CPLEX in Figure 3.4. This figure reveals that CPLEX does not require more than 1% of the total run time in type-2 problems whereas in type-1 problems the optimization can take up to 8% of total time. Although route enhancement seems to require relatively more effort in type-1 problems, the total computation time for type-2 problems is substantially higher (see Table 3.4) and the time devoted to route enhancement corresponds to a small proportion within this large amount of time.

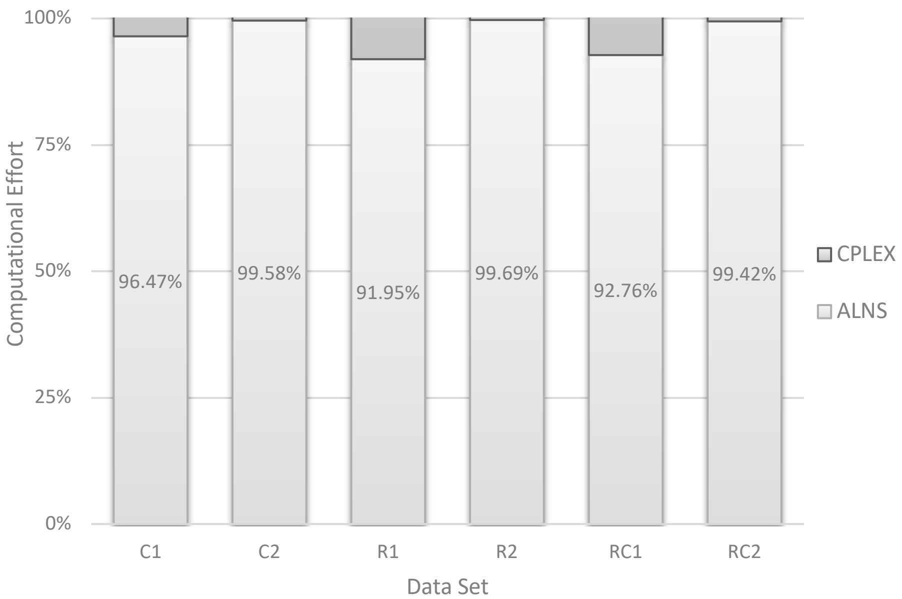


Figure 3.4. Percentage of computational effort required by ALNS vs. CPLEX

In summary, we can conclude that our matheuristic using IA 2 for initialization and performing route enhancement every 200 iterations (i.e. $\Omega = 200$) exhibits the best performance in terms of both solution quality and run time.

3.5.1.2. Effect of Multiple Chargers

In this section, we compare the best results obtained by the proposed matheuristic with the best results that ALNS of Chapter 2 yields for the single charge case where the stations are equipped with only Level 1 (normal) chargers. The results are presented in Table 3.6. In this table, "*Q*", "*TC*" and "*TD*" stand for the battery capacity, the total cost and total distance, respectively. Total cost is equivalent to total distance travelled in the normal charge case. Columns "#*L*1", "#*L*2", and "#*L*3" report the total number of recharges performed by using Level 1, Level 2, and Level 3 chargers, respectively, and "*Qty*" shows the corresponding quantity of energy transferred. The improvements over the single charger results are highlighted in bold.

When we compare the 3-charger results with 1-charger results we see that fast charging is more beneficial when the customers have narrow time windows (i.e. type-1 problems). This is expected because the time spent at stations for recharging the EV can be reduced significantly with fast chargers and the vehicle may be able to serve additional customers along its route including customers that cannot be served otherwise due to strict time-window restrictions. If the EV can serve more customers along its route, then more efficient solutions may be constructed which require fewer number of vehicles and/or consume less energy due to shortened travel distance. The cost of total energy may also go down depending on the charger types utilized and the quantity of energy transferred. In type-1 data set, out of 29 instances we have achieved better solutions in 28 whereas the solution for one instance (c101) has not changed. In addition, the number of vehicles is decreased by two in 8 instances and by one in 20. The fleet size is reduced in all r- and rc-type problems. Furthermore, both the fleet size and energy cost are improved in 13 type-1 instances. We can conclude that the improvements in type-1 problems were accomplished by utilizing fast and super-fast chargers effectively based on the number of recharges and the energy quantities given in the last four columns.

When we analyze the results for type-2 instances we observe that fast charging is able to reduce the fleet size in only one instance (c204) out of 27 and by only one vehicle. Since the time windows can be easily satisfied in these problems the vehicles can serve more customers on their routes and the number of EVs is already few (between 2 and 4). Hence, a reduction in the fleet size is usually impossible. We also observe that fast charging does not help cutting down

Table 3.6. Comparison of results obtained by multiple fast charging vs. single normal charging

	1-Charger					3-Chargers								
Instance	*Q*	*#Veh*	*TC/TD*	*#L1*	*Qty*	*#Veh*	*TD*	*TC*	*#L1*	*Qty*	*#L2*	*Qty*	*#L3*	*Qty*
c101	79.69	12	1043.38	8	165.96	12	1043.38	1043.38	8	165.96	0	0.00	0	0.00
c102	79.69	11	1032.49	10	202.56	**10**	998.48	1009.86	6	143.66	3	70.67	1	21.57
c103	79.69	10	973.39	9	229.52	10	946.79	**961.78**	5	98.76	2	87.40	2	31.25
c104	79.69	10	886.72	7	140.48	**9**	984.82	1006.12	7	93.07	5	136.11	1	38.43
c105	79.69	11	1037.78	9	206.52	**10**	1024.01	**1031.58**	7	172.50	3	75.72	0	0.00
c106	79.69	11	1024.18	9	212.81	**10**	1028.89	1044.96	6	125.67	3	104.42	1	28.11
c107	79.69	10	1058.11	11	273.32	10	1005.84	**1015.81**	7	138.76	3	99.71	0	0.00
c108	79.69	10	1033.50	10	252.33	10	1014.21	**1022.36**	8	174.26	3	81.54	0	0.00
c109	79.69	10	946.84	9	170.88	10	940.38	**940.38**	9	164.42	0	0.00	0	0.00
c201	118.31	4	629.95	3	168.26	4	629.95	629.95	3	168.26	0	0.00	0	0.00
c202	118.31	4	629.95	3	168.26	4	629.95	629.95	3	168.26	0	0.00	0	0.00
c203	118.23	4	629.95	3	168.50	4	629.95	629.95	3	168.50	0	0.00	0	0.00
c204	118.12	4	629.95	3	168.83	**3**	697.22	719.89	3	157.52	3	143.98	1	41.36
c205	117.78	4	629.95	3	169.85	4	629.95	629.95	3	169.85	0	0.00	0	0.00
c206	117.70	4	629.95	3	170.09	4	629.95	629.95	3	170.09	0	0.00	0	0.00
c207	117.66	4	629.95	3	170.21	4	629.95	629.95	3	170.21	0	0.00	0	0.00
c208	117.66	4	629.95	3	170.21	4	629.95	629.95	3	170.21	0	0.00	0	0.00
r101	62.14	18	1641.42	22	533.93	**16**	1788.25	1815.77	23	518.81	12	275.20	0	0.00
r102	62.14	16	1461.38	24	469.22	**15**	1422.69	**1436.95**	19	353.90	6	142.64	0	0.00
r103	62.14	13	1262.75	17	457.31	**12**	1195.76	**1233.72**	8	136.87	10	254.93	2	62.32
r104	67.15	11	1078.99	12	343.29	**10**	1015.81	**1048.69**	3	102.62	7	154.61	3	87.09
r105	62.14	15	1373.94	18	454.85	**13**	1338.52	**1364.98**	15	278.83	9	264.56	0	0.00
r106	62.60	13	1310.46	18	496.66	**12**	1246.47	**1280.16**	10	186.17	14	336.85	0	0.00
r107	66.28	12	1118.91	14	326.82	**10**	1096.09	1130.86	6	90.64	12	337.60	1	5.06
r108	64.06	11	1031.14	13	331.25	**10**	987.04	**1010.68**	7	135.59	5	192.14	2	22.16
r109	65.17	13	1197.57	15	360.49	**11**	1156.36	1198.55	5	82.61	9	306.66	2	57.64
r110	67.12	11	1090.92	17	352.61	**10**	1078.66	1127.70	4	77.84	7	168.81	4	160.81
r111	65.80	12	1084.13	13	304.56	**11**	1064.34	1086.10	8	153.74	5	155.99	1	30.81
r112	65.48	11	1017.31	14	297.03	**10**	986.79	**1015.49**	6	82.95	8	211.04	1	37.99
r201	187.86	3	1262.10	6	698.52	3	1257.50	**1257.50**	7	693.92	0	0.00	0	0.00
r202	238.34	3	1052.32	3	337.30	3	1051.46	**1051.46**	3	336.44	0	0.00	0	0.00
r203	187.90	3	895.54	4	331.84	3	895.54	895.54	4	331.84	0	0.00	0	0.00
r204	247.66	2	780.14	2	284.82	2	779.71	**779.71**	2	284.39	0	0.00	0	0.00
r205	198.88	3	987.36	3	390.72	3	987.36	987.36	3	390.72	0	0.00	0	0.00
r206	181.23	3	922.70	3	379.01	3	922.70	922.70	3	379.01	0	0.00	0	0.00
r207	267.18	2	846.59	2	312.23	2	846.43	**846.43**	2	312.07	0	0.00	0	0.00
r208	218.03	2	736.12	2	300.06	2	736.12	736.12	2	300.06	0	0.00	0	0.00
r209	181.83	3	868.95	4	323.46	3	866.67	**866.67**	4	321.18	0	0.00	0	0.00
r210	187.87	3	843.36	3	299.98	3	843.21	**843.21**	3	299.84	0	0.00	0	0.00
r211	265.71	2	862.56	2	331.14	2	840.61	**840.61**	4	309.19	0	0.00	0	0.00
rc101	79.69	15	1754.75	22	573.65	**14**	1769.82	1800.73	17	403.60	7	191.95	2	58.61
rc102	79.69	14	1526.31	17	455.00	**13**	1531.90	1557.39	12	269.33	8	213.47	1	20.68
rc103	79.69	13	1329.58	13	366.84	**12**	1332.38	1355.71	12	222.11	4	129.22	1	52.05
rc104	79.69	11	1202.93	14	326.34	**10**	1165.39	**1192.60**	7	121.53	8	221.88	1	25.08
rc105	79.69	14	1449.53	17	356.63	**13**	1403.53	**1425.92**	10	156.40	7	223.88	0	0.00
rc106	79.69	13	1402.95	16	372.68	**12**	1369.51	**1388.85**	10	236.05	8	193.48	0	0.00
rc107	79.69	12	1261.03	14	314.68	**11**	1221.72	**1247.87**	8	161.81	6	149.77	1	55.85
rc108	79.69	11	1164.32	13	289.47	**10**	1171.71	1199.24	7	156.53	5	161.34	1	56.95
rc201	211.04	4	1446.84	4	602.68	4	1446.84	1446.84	4	602.68	0	0.00	0	0.00
rc202	273.13	3	1416.96	4	597.57	3	1416.96	1416.96	4	597.57	0	0.00	0	0.00
rc203	209.92	3	1069.27	5	439.51	3	1064.33	**1064.33**	7	434.57	0	0.00	0	0.00
rc204	159.68	3	886.23	6	407.19	3	886.19	**886.19**	5	407.15	0	0.00	0	0.00
rc205	194.58	3	1262.22	8	678.48	3	1255.15	**1256.30**	8	659.89	1	11.51	0	0.00
rc206	229.26	3	1206.09	6	518.31	3	1206.06	**1206.06**	5	518.28	0	0.00	0	0.00
rc207	212.23	3	992.14	4	355.45	3	991.65	**991.65**	5	354.96	0	0.00	0	0.00
rc208	165.63	3	839.71	4	342.82	3	839.71	839.71	4	342.82	0	0.00	0	0.00

the costs either: total energy cost is reduced in 12 instances and the average improvement is only 0.17%. The main contributor to this average value is problem r211 where a 2.54% reduction in energy cost is achieved. These results are in parallel with Desaulniers et al. (2016) who highlighted the minor influence of wide time-window constraints on recharging decisions. Further investigation on these results reveals that $\#L2$ and $\#L3$ values are 0 in most of the instances. So, these slight improvements were achieved by an extended search of the solution space of single charger case rather than by using fast or super-fast chargers. Noting again that IA 1 constructs the initial solution by using only normal chargers, these results now explain why the matheuristic using IA 1 showed a better performance in type-2 problems in Section 5.1.1.

3.5.2. Results for Small Instances

In this section, we solve the small instances to compare the solutions of the matheuristic with the optimal solutions or best bounds given by CPLEX. With CPLEX, we solve Model 1 using a single thread and set the time limit to 7200 seconds. Since the problems are smaller, we perform IA 2 for 10,000 iterations and run the matheuristic 10,000 iterations with $\Omega = 200$. The results are presented in Table 3.7. The computation times reported in columns "*Time*" are in seconds. The values given following "c" and "s" in the instance names represent the number of customers and stations, respectively. Note that CPLEX results reported with a run time of 7200 seconds show the best upper bounds found within the given time limit and are not necessarily the optimal solutions. CPLEX solves all 5- and 10-customer instances to optimality. Our matheuristic also solves them optimally; however, it requires more computational time in most of the instances. On the other hand, the matheuristic outperforms CPLEX in 15-customer instances both in terms of solution quality and run time. The improved results are highlighted in bold in the table. We see that our matheuristic achieved better cost figures in two instances (c103c15-s5 and rc204c15-s7) and provided a solution with one less vehicle in another instance (rc202c15-s5). The latter case corresponds to a major improvement as the fleet size is reduced to a single vehicle from two. We believe that these results show the effectiveness of the proposed matheuristic approach.

Table 3.7. Comparison of results on small size instances

Instance	CPLEX			Matheuristic		
	#Veh	*TC*	*Time*	*#Veh*	*TC*	*Time*
c101c5-s3	2	250.69	< 1	2	250.69	2.50
c103c5-s2	1	175.37	< 1	1	175.37	2.76
c206c5-s4	1	242.56	< 1	1	242.56	3.02
c208c5-s3	1	164.34	< 1	1	164.34	2.62
r104c5-s3	2	136.69	< 1	2	136.69	0.98
r105c5-s3	2	156.08	< 1	2	156.08	2.09
r202c5-s3	1	128.88	< 1	1	128.88	3.08
r203c5-s4	1	179.06	< 1	1	179.06	3.43
rc105c5-s4	2	233.77	< 1	2	233.77	1.67
rc108c5-s4	2	253.93	< 1	2	253.93	2.44
rc204c5-s4	1	185.16	< 1	1	185.16	3.17
rc208c5-s3	1	167.98	< 1	1	167.98	3.23
c101c10-s5	3	382.93	1	3	382.93	4.34
c104c10-s4	1	267.60	27	1	267.60	8.48
c202c10-s5	1	304.06	1	1	304.06	7.99
c205c10-s3	1	283.29	3	1	283.29	5.91
r102c10-s4	3	249.19	1	3	249.19	3.78
r103c10-s3	2	206.30	57	2	206.30	5.22
r201c10-s4	1	241.25	1159	1	241.25	7.76
r203c10-s5	1	222.64	14	1	222.64	26.90
rc102c10-s4	4	415.99	12	4	415.99	3.13
rc108c10-s4	3	347.90	2	3	347.90	4.27
rc201c10-s4	1	412.86	865	1	412.86	6.50
rc205c10-s4	2	325.98	< 1	2	325.98	6.86
c103c15-s5	2	368.91	7200	2	**368.80**	12.47
c106c15-s3	2	310.79	1375	2	310.79	10.04
c202c15-s5	2	381.23	453	2	381.23	19.29
c208c15-s4	1	339.21	7200	1	339.21	21.09
r102c15-s8	5	411.03	7200	5	411.03	5.73
r105c15-s6	3	340.62	353	3	340.62	5.63
r202c15-s6	1	449.81	7200	1	449.81	30.08
r209c15-s5	1	313.24	7200	1	313.24	44.49
rc103c15-s5	4	397.67	7200	4	397.67	5.91
rc108c15-s5	3	370.25	7200	3	370.25	6.87
rc202c15-s5	2	394.39	7200	**1**	648.05	15.95
rc204c15-s7	1	392.76	7200	1	**340.25**	62.45

3.5.3. Results for FORT instances of Felipe et al. (2014)

To the best of our knowledge, Felipe et al. (2014) is the only study that addressed EVRP with multiple charger types and partial recharges but without considering time windows and using a different objective function which minimizes total cost of energy and battery degradation. To further evaluate the performance of our method, we adapt it to solve their case and perform five runs for each instance of FORT data. We implement IA2 with Ω=200 and run the algorithm with two different configurations: (A) 25,000 iterations of ALNS for initialization and 10,000 iterations of matheuristic; (B) 2500 iterations of ALNS and 1000 iterations of matheuristic. Our aim in performing test (B) is to investigate the trade-off between the solution quality and computation time, and to make a fair comparison with the heuristic of Felipe et al. (2014) in terms of run time. Note that they coded their algorithm in Fortran 95 and executed on an Intel Core i5 2.8 GHz processor and 8 GB RAM.

The average results are presented in Table 3.8 and detailed results are given in Appendix C. In this table, "*N*" and "*S*" refer to the number of customers and number of stations in the data, respectively. "*Avg TC*" and "*Avg Time*" report the average total cost and the average computation time (in seconds) of the corresponding problem set. "*% Imp*" shows the percentage improvement achieved by our matheuristic for each configuration and calculated as $(\text{FORT} - \text{Matheuristic})/\text{FORT}$.

Table 3.8. Comparison of average results with Felipe et al. (2014) on FORT instances[*]

		FORT		Matheuristic (A)			Matheuristic (B)		
N	*S*	*Avg TC*	*Avg Time*	*Avg TC*	*Avg Time*	*% Imp*	*Avg TC*	*Avg Time*	*% Imp*
100	9	71.19	274	64.66	181	9.17	65.01	20	8.73
	5	71.59	268	65.12	180	9.02	65.24	31	8.85
200	9	110.38	533	98.75	798	11.19	101.81	80	8.62
	5	114.36	522	101.27	770	11.44	104.02	122	9.03
400	9	195.75	1181	176.61	2936	9.84	182.43	329	6.84
	5	203.18	1101	181.61	3247	10.58	188.18	533	7.33
Average			647		1352	10.21		186	8.23

The average results are presented in Table 3.8 and detailed results are given in Appendix C. In this table, "*N*" and "*S*" refer to the number of customers and number of stations in the data, respectively. "*Avg TC*" and "*Avg Time*" report the average total cost and the average computation time (in seconds) of the corresponding problem set. "*% Imp*" shows the percentage improvement achieved by our matheuristic for each configuration and calculated as $(\text{FORT} - \text{Matheuristic})/\text{FORT}$.

The results reveal that our matheuristic provides significantly better solutions than FORT using both configurations (A) and (B). Reducing the number of iterations in configuration (B) deteriorates the solution quality by 2% on the average while the average run time is approximately $1/8$ of that of configuration (A). Felipe et al. (2014) reported an average run time of 647 seconds whereas our matheuristic (A) and (B) spent on the average 1352 and 186

[*] The cost figures are kindly provided by Gregorio Tirado. The average computation times correspond to the average running times of SA approach reported in Felipe et al. (2014).

seconds, respectively. Even though the problem addressed is slightly different and the two methods were executed on different processors, we believe that these results show the superiority of our matheuristic and confirm its effectiveness.

3.6. Conclusions

In this chapter, we tackled the Electric Vehicle Routing Problem with Time Windows and Fast Chargers. In EVRPTW-FC, the stations are equipped with multiple chargers which vary in power supply, power voltage, and maximum current options. We considered three charger types, namely normal, fast, and super-fast. We formulated two different mathematical models of this problem and compared them in terms of solution quality and computational time. Since the medium and large size problems are intractable, we developed a matheuristic approach to solve the problem efficiently. Our approach combines ALNS with an exact method. In ALNS, while we employed destruction and repair algorithms from the literature, we also introduced new mechanisms specific to the nature of the problem. In the exact method, we fixed the sequence of the customers visited by each vehicle in the solution provided by ALNS and utilized CPLEX solver to optimize the charging related decisions. We also developed an efficient mathematical formulation for this fixed-route single-vehicle problem to be able to find the optimal solution in reasonable run time.

We tested the performance of our algorithm on both small and large benchmark instances from the literature. Our numerical results in small-size instances showed that our matheuristic outperformed CPLEX both in solution quality and run time. In large-size instances, the results revealed the advantage of using fast charging in terms of fleet size and energy consumption. Specifically, we were able to obtain route plans requiring less EVs or reducing energy cost or both in all instances where the time windows are narrow. On the other hand, the influence of the availability of fast chargers was minor when the time windows are wide.

In this study, we assumed that all stations were already located and equipped with all types of chargers. However, this may not be the case considering the high installation costs and lack of infrastructure. So, the problem can be extended to a location routing problem where the recharging stations are sited, their charger equipment and capacities are determined, and the EVs are routed simultaneously. Further research on this topic may also address the heterogeneous fleet case where the vehicles vary by their cargo capacities, battery condition and age which affect their cruising range and discharge/recharge durations. Furthermore, we assumed that recharging stations and chargers were always available. In real life, there may be queues in the stations and the EVs may need to wait for service. Alternatively, it may drive to another station. So, variability in recharging times can be investigated within the stochastic context. The authors are currently working on this extension.

Chapter 4

Electric Vehicle Routing Problem with Soft Time Windows and Time-Dependent Waiting Times at Recharging Stations

4.1. Introduction

The existing literature of the EVRPTW assumes that recharging at a station starts as soon as the EV arrives. However, in practice the number of chargers in a station is limited and a charger may not be available at the time of the vehicle's arrival. It may therefore have to queue before recharging, and this waiting time needs to be taken into account in the routing decisions. The waiting time may vary depending on the location of the station and on the time of the visit. Some variations in the waiting time are difficult to predict. For instance, if a traffic accident or a special event occurs near a station, there may be long queues. On the other hand, some delays are easier to foresee such as those observed during rush hour congestion. In this study, we assume that the expected queue length at all stations and at any time of the day is known in advance. Furthermore, the charging time is a non-linear concave function of the charge amount (Pelletier et al., 2017, Montoya et al., 2017).

In this chapter, we extend the EVRP by considering time-dependent waiting times at the recharging stations using an M/G/1 queueing system. We use a non-linear charging function and soft time windows at the customer locations. We approximate the random waiting times by their expected values. The planning horizon is split into a predetermined number of time intervals and an average queue length is assigned to each station for different time intervals. The routing decisions are then made according to these time-dependent waiting times at the recharging stations. We formulate the problem as a mixed integer linear program and propose a matheuristic to solve it efficiently. Our algorithm is a combination of the well-known ALNS metaheuristic and a mixed integer linear programming.

This chapter makes two main scientific contributions. It first introduces a new problem called the Electric Vehicle Routing Problem with Soft Time Windows and Time Dependent Waiting Times at Recharging Stations, TD-EVRPSTW, using a comprehensive objective function that minimizes the total cost of fuel, drivers, vehicles, overtime, and lateness penalties at the customer locations. It formulates the problem in two ways: a first comprehensive formulation is used to compute an optimal solution from scratch, while the second formulation optimizes a single route in which the customer sequence is known in advance. Secondly, it proposes a matheuristic that combines ALNS with an exact solver along with problem-specific mechanisms.

The remainder of the chapter is organized as follows. Section 4.2 provides a summary of the literature. Section 4.3 describes the problem and presents the mathematical model, while Section 4.4 describes the proposed solution approach with the single-route optimization formulation. Section 4.5 presents the computational study and provides the numerical results, followed by concluding remarks in Section 4.6.

4.2. Related Literature

The EVRP is first studied by Conrad and Figliozzi (2011) assuming a constant recharging time. The model allows charging fully or up to the 80% of the battery capacity at only customer locations. Erdoğan and Miller-Hooks (2012) then study the routing of AFVs which are refueled up to the tank capacity at alternative fuel stations within a constant amount of time. Schneider et al. (2014) introduce time windows and a linear charging function assuming that the battery is fully recharged. Since then, several extensions have been proposed. The full charge assumption is first relaxed and models are developed to allow partial charging (Bruglieri et al., 2015; Chapter 2). Some models take into account the fact that the stations have different chargers (Felipe et al., 2014; Sassi et al., 2014; Li-ying and Yuan-bin, 2015; Chapter 3). Several studies have analyzed the heterogeneous fleet case. Some of them consider a mixed fleet composed of EVs with different characteristics (Desaulniers et al., 2016; Ji et al., 2018) whereas in other studies, the fleet includes different ICEVs as well (Sassi et al., 2014; Goeke and Schneider, 2015; Hiermann et al., 2018; Kopfer and Vornhusen, 2018). Some recent papers have dealt with both the location of the stations and the routing (Li-ying and Yuan-bin, 2015; Yang and Sun, 2015; Paz et al., 2018; Schiffer and Walther, 2017). Although recharging is performed at charging stations in most studies, some papers consider battery swap stations (BSS) where the discharged battery is replaced with a fully recharged one (Paz et al., 2018; Wang et al., 2018; Jie et al., 2018), and wireless charging systems (WCS) where the battery is recharged by an inductive charging system placed along the roads while the EV is traveling (Li et al., 2018). Some recent studies have considered EVs within the context of the two-echelon VRP (Jie et al., 2018; Breunig et al., 2018), technician routing (Villegas et al., 2018) and reverse logistics (Zhang et al., 2018a). Finally, the linear charging function is relaxed by Montoya et al. (2017) and Froger et al. (2017a,b) who use a concave piecewise linear function. Afroditi et al. (2014) and Pelletier et al. (2016) review EVs in goods distribution. Regarding the unknown service times at the stations, in Sweda et al. (2017) the stations are unavailable with a certain probability. If a vehicle visits an occupied station, it should wait for some time which is random. Froger et al. (2017b) solve a related problem in which the stations have limited number of chargers (one, two or three) and an EV may need to wait for service if the chargers are busy with recharging other EVs in the fleet. In this problem, the use of the chargers depends on the routing and charging decisions, whereas in our setting the queue lengths are independent of these features.

Table 4.2 classifies the EVRP studies in terms of the main features of the problem. The column headings stand for recharging function (RC fcn), electricity consumption function (EC fcn), fleet composition (fleet comp), charger types in the stations, objective function (obj fcn), charge amount, solution methodology, and existence of time windows (TW) and station location decisions (Lctn) in the problem; the letters L and NL represent linear and non-linear functions; Ho and Hc correspond to homogeneous and heterogeneous fleets; (A)VNS, TS, ALNS, ILS, HC, AC, DP, and CG stand for (adaptive) variable neighborhood search, tabu search, adaptive large neighborhood search, iterated local search, heuristic concentration, ant colony, dynamic programming, and column generation, respectively; MIP is used for the papers that only formulate the mathematical model and report results obtained by a solver. Explanations of the numbers associated with the objective function are provided in Table 4.1.

Table 4.1. Meanings of the objective functions in Table 4.2

1:	Vehicle cost	8:	Total time cost
2:	Total travel cost	9:	Fuel cost
3:	Recharging cost	10:	Battery cost
4:	Waiting cost	11:	Charging cost
5:	Station installation cost	12:	Profit of visits
6:	Stopping cost at a station	13:	Operational costs
7:	Overcharging cost	14:	Battery swapping cost

The time-dependent vehicle routing literature has mainly focused on time-dependent travel times. The papers can be classified as static or dynamic if the times are fixed for a given time interval, or change dynamically, respectively. Static models include deterministic (Malandraki and Daskin, 1992; Jung and Haghani, 2001; Hashimoto et al., 2008; Jabali et al., 2012; Franceschetti et al., 2017) or stochastic features (Van Woensel et al., 2008; Nahum and Hadas, 2009; Taş et al., 2013; Huang et al., 2017; Çimen and Soysal, 2017). Fleischmann et al., (2004); Haghani and Jung, (2005); Potvin et al., (2006) and Schilde et al., (2014) are some papers based on dynamic models. On the other hand, only Taş et al. (2016) consider the time-dependent service times in the context of the traveling salesman problem. A comprehensive review of the time-dependent VRP is provided by Gendreau et al. (2015).

4.3. Problem Description and Formulation

Given a homogeneous fleet of EVs, the problem is to determine a set routes that cover all customers, which have demands and soft time windows, while preserving time and energy feasibility. The EVs may visit recharging stations to charge their batteries and continue their routes. The stations are equipped with a single charging technology. For this reason, an arriving EV may face a queue and have to wait to be serviced. This waiting time varies depending on the time of the day. The energy transferred is assumed to be a concave function of the charging time (Montoya et al., 2017) and the cost of charging is proportional to the amount of energy transferred. The EVs have to pay for the energy they receive. If an EV arrives at a customer before the early service time, it has to wait until that time, but if it arrives later than the late service time, a penalty proportional to the lateness is incurred. Furthermore, the vehicles have an acquisition cost and the drivers are paid an hourly wage. If an EV arrives at the depot after the regular shift hours, its driver is paid an overtime wage. The objective is to determine minimum cost routes satisfying all constraints.

4.3.1. Time-Dependent Waiting Time Functions

We assume that each recharging station is equipped with a single charger and the EVs arrive at the stations according to a Poisson distribution with mean λ. Hence, each station is a system involving a charger as the server and the EVs as the entities. When an EV arrives at a station, if the server is idle, it receives service immediately. On the other hand, if there is another EV recharging at the time of the arrival, then the newly arriving EV has to wait until the recharging finishes. Furthermore, there may be other EVs which have already queued. In this case, the newly arriving EV should join the queue and wait until all EVs, which arrived before, are serviced. Since the stations are public, we do not have information about other EVs and their recharge amounts. Hence, it is difficult to assign a distribution function for the service time. For this reason, the service time is drawn from a general distribution whose mean and standard deviation are known. In fact, it can be any distribution. Since this is a case with a single server,

Table 4.2. EVRP literature review

Papers	RC fcn.		EC fcn.		Fleet comp.		Charger type		Obj. fcn.	Charge amount		Solution method	TW	Lctn.
	L	NL	L	NL	Ho	Hc	Single	Multi.		Full	Partial			
Conrad and Figliozzi (2011)	-		✓		EVs		✓		1, 2, 3, 8	✓	✓	Iterative construction, improvement heuristics	-	-
Erdoğan and Miller-Hooks (2012)	-		✓		AFVs		✓		2	✓		Several heuristics	-	-
Schneider et al. (2014)	✓		✓		EVs		✓		1, 2	✓		Hybrid VNS and TS	✓	-
Felipe et al. (2014)	✓		✓		EVs			✓	2, 3		✓	Several heuristics and SA	✓	-
Sassi et al. (2014)	✓		✓			EVs, ICEVs		✓	1, 2, 3		✓	Several heuristics	✓	-
Li-ying and Yuan-bin (2015)	***		✓		EVs			✓	3, 5, 8	✓		Hybrid AVNS and TS	✓	✓
Goeke and Schneider (2015)	✓		*			EVs, ICEVs	✓		2, 3, 8,9,10		✓	ALNS	✓	-
Bruglieri et al. (2015)	✓		✓		EVs		✓		1, 2, 3, 4		✓	VNS branching	✓	-
Ding et al. (2015)	✓		✓		EVs		✓		2		✓	Hybrid VNS and TS	✓	-
Yang and Sun (2015)	-		✓		EVs		BSS		2, 5	-		Several heuristics	-	✓
Keskin and Çatay (2016)	✓		✓		EVs		✓		1, 2		✓	ALNS	✓	-
Desaulniers et al. (2016)	✓		✓		EVs		✓		2	✓	✓	Branch-price-and-cut	✓	-
Hiermann et al. (2016)	✓		✓			EVs	✓		1, 2	✓		Branch-price, and ALNS	✓	-
Wen et al. (2016)	✓		✓		EVs		✓		1, 2		✓	ALNS	✓	-
Lin et al. (2016)	✓		✓		EVs		✓		1, 2, 3	✓		MIP	-	-
Sweda et al. (2017)	**		✓		EVs			✓	2, 3, 4		✓	Several optimal and heuristic procedures	-	-
Montoya et al. (2017)		✓	✓		EVs		✓		8		✓	Hybrid ILS and HC	-	-
Schiffer and Walther (2017)	✓		✓		EVs		✓		1, 2, 5	✓	✓	MIP	✓	✓
Leggieri and Haouari (2017)	✓		✓		AFVs		✓		2		✓	MIP	-	-
Froger et al. (2017a)		✓	✓		EVs		✓		8		✓	MIP	-	-
Froger et al. (2017b)		✓	✓		EVs		✓		8		✓	Two-stage matheuristic	-	-
Paz et al. (2018)	****		✓		EVs		BSS		1, 2, 5	✓	✓	MIP	✓	✓
Li et al. (2018)	✓		✓		EVs		single chargers and WCS		3, 8		✓	MIP	-	-
Kopfer and Vornhusen (2018)	✓			✓		EVs, ICEVs	✓		11		✓	MIP	✓	-
Wang et al. (2018)	-		✓		EVs		BSS		12	✓		Heuristic procedures	✓	-
Zhang et al. (2018a)	-			✓	EVs		✓		3	✓		AC and ALNS	-	-
Schiffer and Walther (2018)	✓		✓		EVs		✓		1, 5, 8		✓	ALNS, DP	✓	✓
Zhang et al. (2018b)	✓		✓		EVs		✓		1, 2, 3, 4, 8		✓	TS	✓	-
Breunig et al. (2018)	✓		✓		EVs		✓		1, 2	✓		LNS, Exact algorithm	-	-
Jie et al. (2018)	-		✓			EVs	BSS		2, 13, 14		✓	Hybrid CG and ALNS	-	-
Macrina et al. (2018)	✓		✓		EVs, ICEVs		✓		2, 3		✓	ILS	✓	-
Keskin and Çatay (2018)	✓		✓		EVs			✓	1, 3		✓	ALNS based matheuristic	✓	-
Villegas et al. (2018)		✓	✓		EVs, ICEVs			✓	1,2, 3, 6		✓	GRASP based matheuristic	✓	-
Hiermann et al. (2018)	✓		✓			EVs, ICEVs, PHEVs	✓		1,3,9		✓	GA, LNS with IP solver	✓	-

* depends on vehicle mass, speed and gradient of the terrain
** linear but has different cost components for different charge levels
*** linear but has different cost and rates for different chargers
**** linear, constant for battery swapping and parallel with service time

Poisson arrivals and generally distributed service times, we may consider the M/G/1 queueing system. In M/G/1 queue, customers arrive one by one according to a Poisson distribution. The letter M in the queue representation stands for “memoryless property” of the exponential distribution. Since the arrivals are Poisson arrivals, interarrival times between two consecutive customers follow exponential distribution with parameter λ. The service times are independent and identically distributed with distribution function F_S and density f_S. Since this is a general distribution, i.e., the service times does not belong to a specific distribution, G stands for the “general distribution”. Finally, the number 1 shows that there is only one server in the system. Let the mean and standard deviation of the service time be $E[S]$ and $Var[S]$, respectively. In order to satisfy stability of the system, the utilization rate of the charger, $\rho = \lambda E[S]$ should be less than one. The EVs receive service according to a first-in-first-out (FIFO) discipline. Upon arrival of a new EV, if the charger is idle, the EV receives service immediately. If the charger is busy with recharging one EV but there are no other EVs waiting in the queue, a newly arriving EV waits for the remaining recharging time of the EV which is in service. If there is one EV waiting in the queue as well as an EV in service, then the waiting time for a newly arriving EV is the remaining recharging time of the EV being serviced and all recharging time of other EV waiting in the queue. The average waiting time is derived considering all such possible states of the system.

We handle time-dependent waiting times by discretizing the time and dividing the planning horizon into a set of time intervals $[t^m - t^{m+1}]$, $m \in M$. This approach is similar to the time-dependent travel times case where the time is split into intervals and each interval is assigned an average speed. The starting and ending times of interval m are denoted by t^m and t^{m+1}, respectively. Since the traffic density changes with time, the arrival rate of the vehicles is also a function of time. Different stations may have different arrival rates since they are positioned at different locations. In Figure 4.1, we divide the day into five time intervals and illustrate a sample pattern of the arrival rate function $\lambda_i(t)$ for station i. The noon interval is the most crowded time of the day. Late afternoon and morning are the second and third most crowded times, while the evening and night have the least traffic densities. In real life, $\lambda_i(t)$ is a smooth function as depicted in Figure 4.1. However, for modeling purposes, we will discretize the time and approximate $\lambda_i(t)$ by a piecewise linear function as illustrated in Figure 4.2. In this figure, the arrival rates for station i during the morning, noon, late afternoon, and evening time intervals are assumed constant and equal to λ_i^1, λ_i^2, λ_i^3 and λ_i^4, respectively. In order to preserve the FIFO property, transient periods should be introduced between the intervals. It will satisfy that, an EV which arrives at a station later than another EV cannot leave the station earlier.

Then these transient zones will have variable rates, contrary to other zones. These assumptions lead to an M/G/1 queueing system with Poisson arrivals having varying rates and a service time coming from a general distribution function.

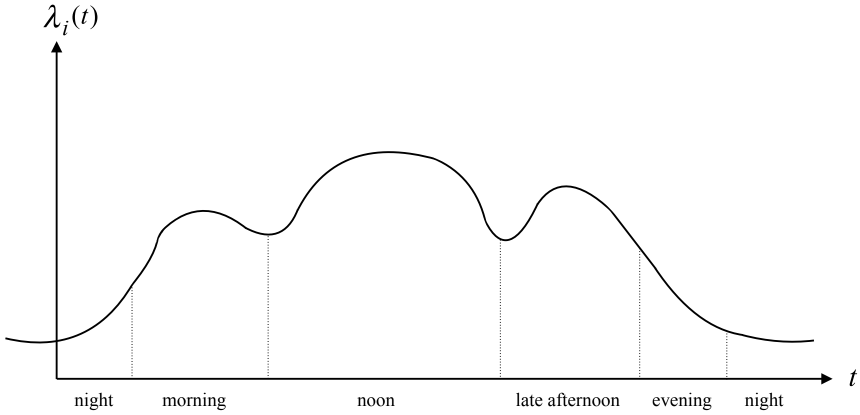


Figure 4.1. Arrival rate of EVs at station i as a function of time

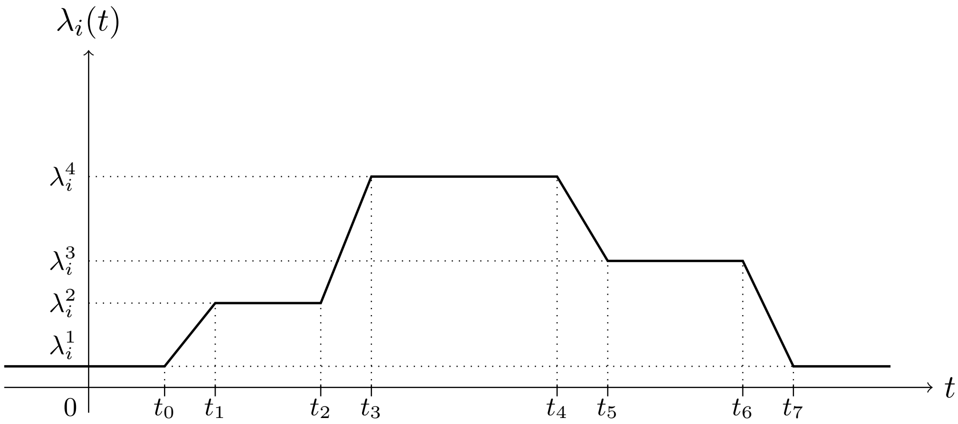


Figure 4.2. Piecewise linear approximation of the arrival rate as a function of time

For an EV arriving at station i at time t, the expected waiting time in the queue, $E[W_i(t)]$ can then be calculated using the steady-state equations of the M/G/1 queueing system as follows:

$$E[W_i(t)] = \frac{\rho}{1-\rho}\frac{C_S^2+1}{2}E[S] , \qquad \text{where} \qquad C_S^2 = \frac{Var(S)}{E^2[S]} \quad . \tag{4.1}$$

Clearly, $E[W_i(t)]$ is non-linear in the transient zones. Hence, we approximate it by a linear function as shown in Figure 4.3

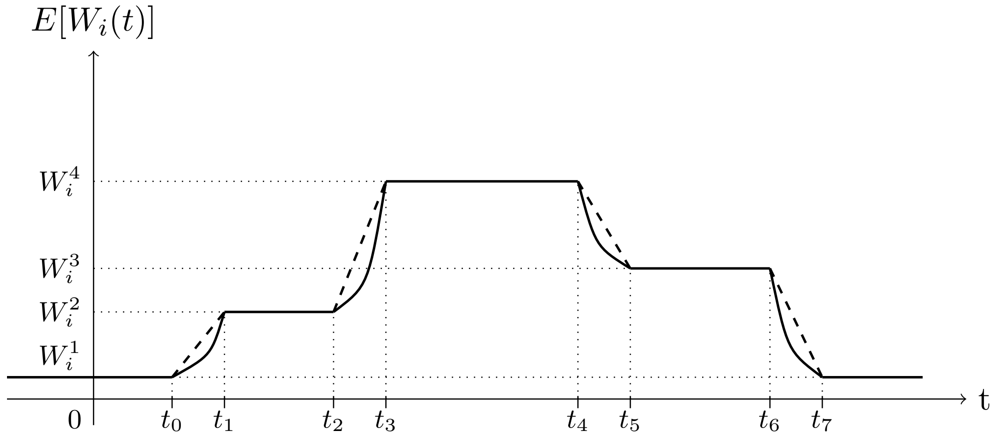


Figure 4.3. Piecewise linear approximation of waiting time in the queue as a function of time

Each segment of the function $E[W_i(t)]$ is characterized by an intercept W_i^m and a slope s_i^m. Hence, if a vehicle arrives at station i in the m^{th} time interval, i.e. at time t, it will wait for a time $W_i^m + s_i^m(t - t^m)$ in the queue before being serviced.

The service time for an EV is the recharging time at a station. In this study, we assume that the EVs operate between 10% and 90% of the battery SoC to minimize the degradation (Pelletier et al., 2017). Hence, all EVs enter and leave a station with an SoC within this range. To calculate the mean and variance of the service time, we assume that the amount of energy transferred follows a uniform distribution between 0 and $0.8q$, where q is the battery capacity. Hence, we can calculate the expected recharge amount, the expected time required to recharge the battery, and the variance of the recharging time using the equations of moments of the uniform distribution. Let s be the recharging time. Then the expected service time at a recharging station will be $E[s] = (0.8q \times r)/2 = 0.4q \times r$, where r is the recharging rate. Similarly, the variance of the service time can be computed as $Var[s] = (0.8q \times r)^2/12$.

4.3.2. Mathematical Formulation

Let $V = \{1, \dots, n\}$ be the set of vertices representing customers to be served, and let S be the set of recharging stations. Since a station may be visited multiple times, we create copies of the stations. We denote by F' the set of recharging stations along with their copies. Let $V' = V \cup F'$ be the set of customers and stations. Each vehicle may have different battery SoC when departing from the depot and arriving at the depot. For this reason, copies of the depot vertex should also be defined. Let V^d be the set of vertices from which the vehicles depart, and let V^a

be the set of vertices at which the vehicles arrive. Defining the sets $V_0 = V \cup V^d$, $V_{N+1} = V \cup V^a$, $V_{0,N+1} = V^d \cup V_{N+1}$, $V'_0 = V' \cup V^d$, $V'_{N+1} = V' \cup V^a$, $V'_{0,N+1} = V^d \cup V'_{N+1}$, $F'_0 = F' \cup V^d$, the problem can be represented by a complete graph $G = (V'_{0,N+1}, A)$, where $A = \{(i,j) | i,j \in V'_{0,N+1}, i \neq j\}$.

With each arc are associated a distance d_{ij} and a travel time t_{ij}. Each vehicle has a load capacity C and a battery capacity Q. The energy is consumed at a rate of h per distance unit. Hence, traversing arc (i,j) requires $h.d_{ij}$ units of energy. Each customer $i \in V$ is associated with demand d_i, service time s_i and soft time window $[e_i, l_i]$. Arrivals later than l_i are allowed but are penalized by a late arrival cost c_p multiplied by the lateness. We also assume that the depot has soft time windows whose violation is penalized by an overtime wage c_o. Each vehicle has an operating cost c_f which may represent maintenance and acquisition costs, and the driver is paid at a wage of c_d per time unit.

The charging function is known to be concave (Pelletier et al. 2017). Hence, we adopt a non-linear charging function and approximate it by a piecewise linear function as in Montoya et al. (2017). The unit cost of energy is c_e. Figure 4.4 illustrates the piecewise linear approximation of the charging function, where c_l and a_l represent the time and charge level, respectively, for the breakpoints $l \in B$, and $B = \{0, \dots, b\}$ is the set of breakpoints of the piecewise linear approximation.

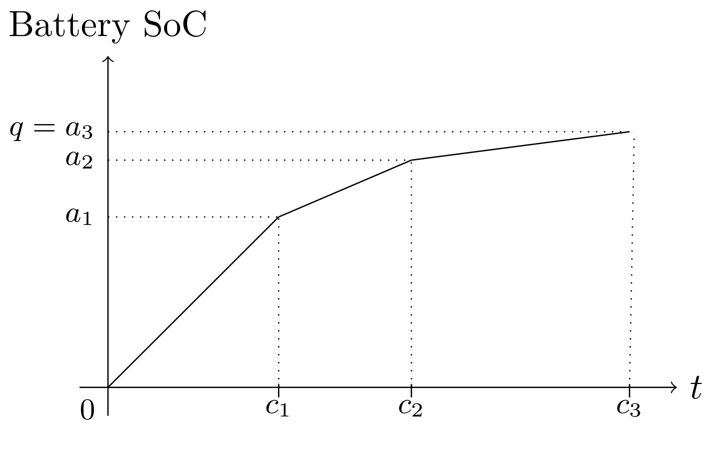


Figure 4.4. Piecewise linear approximation for the charging function (Montoya et al., 2017)

Table 4.3. Mathematical notation

Sets:	
V	Set of customers
F	Set of recharging stations
F'	Set of recharging stations with their copies
V'	Set of customers and stations with their copies ($V \cup S'$)
V^d	Set of departure depots
V^a	Set of arrival depots
F_0'	Set of departure depots and stations with their copies ($V^d \cup F'$)
F_{N+1}'	Set of arrival depots and stations with their copies ($F' \cup V^a$)
V_0	Set of departure depots and customers ($V \cup V^d$)
V_{N+1}	Set of arrival depots and customers ($V \cup V^a$)
$V_{0,N+1}$	Set of departure and arrival depots with customers ($V^d \cup V_{N+1}$)
V_0'	Set of departure depots, customers, and stations with their copies ($V^d \cup V'$)
V_{N+1}'	Set of arrival depots, customers, and stations with their copies ($V' \cup V^a$)
$V_{0,N+1}'$	Set of all vertices ($V^d \cup V_{N+1}'$)
$\tilde{V}$	Set of i, j pairs such that $i \in F', j \in V_{N+1}$, and $i \in V_0, j \in V_{N+1}'$
Parameters:	
d_{ij}	Distance from node *i* to node *j*
t_{ij}	Travel time from node *i* to node *j*
d_i	Demand of customer *i*
s_i	Service time of customer *i*
e_i	Early service time of customer *i*
l_i	Late service time of customer *i*
C	Cargo capacity of the vehicles
Q	Battery capacity of the vehicles
c_l	Time of breakpoint l in the approximated recharging time function
a_l	SoC of breakpoint l in the approximated recharging time function
t^m	Beginning of m^{th} time interval
W_j^m	Average waiting time at station j at the beginning of m^{th} time interval
s_j^m	Slope of the $E[W_j(t)]$ function of station j in m^{th} time interval
h	Fuel consumption rate
c_e	Unit energy cost
c_p	Unit late service cost of customers
c_d	Driver wage per unit time
c_o	Overtime wage per unit time
c_f	Fixed vehicle cost
Decision variables:	
x_{ij}^m	1 if EV departs from vertex i and arrives at vertex j in time interval m, 0 otherwise
z_{il}	1 if battery SoC at the arrival at station i is between a_{l-1} and a_l, $l \in B \setminus \{0\}$, 0 otherwise
y_{il}	1 if battery SoC at the departure from station i is between a_{l-1} and a_l, $l \in B \setminus \{0\}$, 0 otherwise
τ_i	Service starting time upon arrival at vertex i
v_i	Lateness at customer i due to the late arrival of the vehicle
u_i	Remaining cargo capacity upon arrival at vertex i
y_i^a	Battery SoC at vertex i

Decision variables:

y_i^d	Battery SoC when departing from station i
w_i	Queue waiting time of the EV arriving at station i
r_{il}^a	Coefficient of breakpoint l in the piecewise approximation when EV arrives at station i
r_{il}^d	Coefficient of breakpoint l in the piecewise approximation when EV departs from station i

Unlike what is done in Montoya et al. (2017), we assume that all stations are equipped with the same charger type; hence the vehicles are always subject to the same charging function. We denote by W_j^m the expected waiting time at station j in time interval m, while s_j^m is the slope of the piecewise linear waiting time function in time interval m. The sets, parameters and decision variables of the problem are presented in Table 4.3.

Given the above definitions, the mathematical model for the problem with time-dependent waiting times at the stations is formulated as follows:

$$\text{minimize} \quad c_e \sum_{i \in V_0'} \sum_{j \in V_{N+1}'} \sum_{m \in M} d_{ij} x_{ij}^m + c_p \sum_{i \in V} v_i + (c_o - c_d) \sum_{i \in V^a} v_i \tag{4.2}$$

$$+ c_f \sum_{i \in V_0'} \sum_{j \in V^a} \sum_{m \in M} x_{ij}^m + c_d \sum_{i \in V^a} \tau_i$$

subject to

$$\sum_{m \in M} \sum_{j \in V_{N+1}'} x_{ij}^m = 1 \qquad i \in V \tag{4.3}$$

$$\sum_{m \in M} \sum_{j \in V_{N+1}'} x_{ij}^m \leq 1 \qquad i \in F' \tag{4.4}$$

$$\sum_{m \in M} \sum_{i \in V_0'} x_{ij}^m = \sum_{m \in M} \sum_{i \in V_{N+1}'} x_{ji}^m \qquad j \in V' \tag{4.5}$$

$$\sum_{m \in M} \sum_{j \in V'} x_{ij}^m \leq 1 \qquad i \in V^d \tag{4.6}$$

$$\sum_{m \in M} \sum_{i \in V'} x_{ij}^m \leq 1 \qquad j \in V^a \tag{4.7}$$

$$\sum_{m \in M} \sum_{i \in V^d} \sum_{j \in V'} x_{ij}^m = \sum_{m \in M} \sum_{i \in V^a} \sum_{j \in V'} x_{ji}^m \tag{4.8}$$

$$\tau_i + (t_{ij} + s_i) \sum_{m \in M} x_{ij}^m - l_0 \left(1 - \sum_{m \in M} x_{ij}^m\right) \leq \tau_j \qquad i \in V_0, j \in V_{N+1}' \tag{4.9}$$

$$\tau_i + \left(\sum_{l \in B} r_{il}^d c_l - \sum_{l \in B} r_{il}^a c_l\right) + w_i(\tau_i) + t_{ij} - l_0 \left(1 - \sum_{m \in M} x_{ij}^m\right) \leq \tau_j \qquad i \in F', j \in V_{N+1} \tag{4.10}$$

$$t^m \sum_{i \in V_0} x_{ij}^m \leq \tau_j \leq t^{m+1} + l_0 \left(1 - \sum_{i \in V_0} x_{ij}^m\right) \qquad j \in V_{N+1}', m \in M \tag{4.11}$$

$$t^m \sum_{i \in F'} x_{ij}^m \leq \tau_j \leq t^{m+1} + l_0 \left(1 - \sum_{i \in F'} x_{ij}^m\right) \qquad j \in V_{N+1}, m \in M \tag{4.12}$$

$$e_i \leq \tau_i \qquad i \in V_{N+1}' \tag{4.13}$$

$$v_i \geq (\tau_i - l_i) \qquad i \in V_{N+1} \tag{4.14}$$

$$\tau_i \le l_0 \qquad i \in V^a \qquad (4.15)$$

$$w_i(\tau_j) \ge W_j^m + s_j^m(\tau_j - t^m) - M(1 - x_{ij}^m) \qquad i \in V_0, j \in F', m \in M \qquad (4.16)$$

$$y_i^a = \sum_{l \in B} r_{il}^a a_l \qquad i \in F' \qquad (4.17)$$

$$y_i^d = \sum_{l \in B} r_{il}^d a_l \qquad i \in F' \qquad (4.18)$$

$$y_j^a \le y_i^d - \sum_{m \in M} x_{ij}^m(hd_{ij}) + 0.9Q\left(1 - \sum_{m \epsilon M} x_{ij}^m\right) \qquad i,j \in \tilde{V} \qquad (4.19)$$

$$y_j^a \ge y_i^d - \sum_{m \in M} x_{ij}^m(hd_{ij}) - 0.9Q\left(1 - \sum_{m \epsilon M} x_{ij}^m\right) \qquad i,j \in \tilde{V} \qquad (4.20)$$

$$y_i^a \ge 0.1Q \sum_{m \epsilon M} \sum_{j \epsilon V_0'} x_{ji}^m \qquad i \in V_{N+1}' \qquad (4.21)$$

$$y_i^a \le y_i^d \le 0.9Q \qquad i \in V_0' \qquad (4.22)$$

$$u_i \le C \qquad i \in V^d \qquad (4.23)$$

$$u_j \le u_i - d_i \sum_{m \epsilon M} x_{ij}^m + C\left(1 - \sum_{m \epsilon M} x_{ij}^m\right) \qquad i \in V_0', j \in V_{N+1}' \qquad (4.24)$$

$$\sum_{l \epsilon B} r_{il}^a = \sum_{l \epsilon B \setminus \{0\}} z_{il} = \sum_{m \epsilon M} \sum_{j \epsilon V_{N+1}'} x_{ij}^m \qquad i \in F' \qquad (4.25)$$

$$r_{i0}^a \le z_{i1} \qquad i \in F' \qquad (4.26)$$

$$r_{il}^a \le z_{il} + z_{i,l+1} \qquad i \in F', l \in B \setminus \{0, b\} \qquad (4.27)$$

$$r_{ib}^a \le z_{ib} \qquad i \in F' \qquad (4.28)$$

$$\sum_{l \epsilon B} r_{il}^d = \sum_{l \epsilon B \setminus \{0\}} y_{il} = \sum_{m \epsilon M} \sum_{j \epsilon V_{N+1}'} x_{ij}^m \qquad i \in F' \qquad (4.29)$$

$$r_{i0}^d \le y_{i1} \qquad i \in F' \qquad (4.30)$$

$$r_{il}^d \le y_{il} + y_{i,l+1} \qquad i \in F', l \in B \setminus \{0, b\} \qquad (4.31)$$

$$r_{il}^a \le y_{ib} \qquad i \in F' \qquad (4.32)$$

$$\sum_{m \epsilon M} \sum_{j \epsilon F'} x_{ij}^m = 0 \qquad i \in F' \qquad (4.33)$$

$$x_{ij}^m \in \{0,1\} \qquad i \in V_0', j \in V_{N+1}', m \qquad (4.34)$$

$$z_{ij}^m \ge 0 \qquad i \in F', j \in V_{N+1}, m \qquad (4.35)$$

$$w_{ib}, w_{ib} \in \{0,1\} \qquad i \in F', b \in B \qquad (4.36)$$

$$y_i^a, y_i^d, w_i \ge 0 \qquad i \in F' \qquad (4.37)$$

$$v_i \ge 0 \qquad i \in V_{N+1} \qquad (4.38)$$

$$u_i, \tau_i \ge 0 \qquad i \in V_{0,N+1}' \qquad (4.39)$$

$$r_{il}^a, r_{il}^d \ge 0 \qquad i \in F', l \in B \qquad (4.40)$$

The objective function (4.2) minimizes total cost equal to the sum of five terms. The first term corresponds to the energy cost which is proportional to the distance traveled. The second and third terms are the penalties associated with customer and depot time window violations, respectively. The fourth term is total cost of vehicles, while the last term computes the driver cost. Constraints (4.3) ensure that each customer is visited exactly once whereas constraints (4.4) indicate that each station is visited at most once. The connectivity of the nodes is ensured by constraints (4.5). Constraints (4.6) and (4.7) keep track of departures from the depots and arrival at the depots. Constraint (4.8) ensures that the number of departure depots used should be equal to the number of arrival depots. Constraints (4.9) and (4.10) keep track of arrival times after departing from a customer or a recharging station, respectively. While traveling from vertex i to vertex j, if $x_{ij}^{\bar{m}}$ is 1, for instance, the arrival time at j should be in the $\bar{m}^{th}$ time interval, which is between $[t^{\bar{m}} - t^{\bar{m}+1}]$. This in ensured by constraints (4.11) for departures from the depot or a customer, and by constraints (4.12) for departures from a station. Constraints (4.13) mean that the vehicle starts its service after the early service time of that customer, while constraints (4.14) guarantee that lateness will have a positive value when vehicle arrives at a customer after its late service time. Constraints (4.15) state that the vehicles should end their trips before the maximum time limit l_0. Constraints (4.16) link the flow variables x_{ij}^{m} and the waiting time $w_j(\tau_j)$ at a station j. Constraints (4.17) and (4.18) determine the SoC of the vehicle at arrival at and departure from station i in terms of the breakpoints of the piecewise linear approximation. Constraints (4.19) and (4.20) track the SoC at departure from a recharging station and a customer/depot, respectively. The battery operates between 10% and 90% of its capacity in order to minimize the degradation. Constraints (4.21) and (4.22) set the battery utilization between 10% and 90% of its capacity. Constraints (4.23) and (4.24) keep track of load of the vehicle and ensure that it is initially smaller than the load capacity and it never goes below 0. Constraints (4.25) and (4.29) establish the relationship between the binary variables z_{il} and y_{il} and the coefficients of the breakpoints in the piecewise approximation when an EV enters and leaves a station, respectively. Constraints (4.26)−(4.28) and (4.30)−(4.32) ensure that the coefficients related to the piecewise linear approximation of the charging function take correct values. Constraint (4.33) prevents the consecutive visits to multiple stations. Finally, (4.34)−(4.40) define the domains of the decision variables.

4.4. Solution Methodology

Since the problem generalizes the classical VRP, which is NP-hard, it is intractable for large-size instances. Hence, we propose a matheuristic to solve it within reasonable time. Matheuristics have been used in vehicle routing problems successfully (Archetti and Speranza, 2014). In our approach, ALNS is used to search the feasible space and determine feasible routes, while the charging decisions are optimized by solving an integer linear program exactly by CPLEX.

4.4.1. Adaptive Large Neighborhood Search

The ALNS metaheuristic, proposed by Ropke and Pisinger (2006a,b), is a search framework based on iterative destroy and repair phases, that has been successfully employed for solving various VRP variants. In this study, we use two types of operators which are designed for customers and stations. Candidate solutions are accepted according to a simulated annealing criterion.

4.4.1.1. Initial Solution

At the beginning of the algorithm, all customers are inserted in a removal list in a random order. A greedy customer insertion procedure (see Section 4.4.1.3) is then applied to this list to construct the initial solution x_0.

4.4.1.2. Customer Removal

The current feasible solution $x_{current}$ is destroyed by the removal operators, and insertion operators are then applied to generate a new solution. Customer removal (CR) operators remove γ customers and add them to a removal list. We use the random, worst-distance, worst-time, Shaw, proximity-based, demand-based, time-based, zone, random route and greedy route removals which have been used in related studies (Demir et al., 2012, Emeç et al., 2016). In addition, we introduce the following operator.

Expensive customer removal: This operator tends to remove customers whose visiting cause a high increase in the objective function. It first identifies the customers whose predecessor or successor nodes are stations, and sorts them in non-increasing order of their costs which are calculated in the following way. Let station i be the neighboring node of customer j. Then the total cost of traveling, recharging and waiting will be $c_e\left(d_{ji} + y_i^d - y_i^a\right) + c_d w_i$. The operator removes from the solution the customers having the γ largest costs.

After the removal of the selected customers, some stations may become unnecessary to visit. A route refinement procedure is applied to eliminate these unnecessary stations. It evaluates the decrease in the objective function resulting from the elimination of each station if this elimination is feasible. The station whose removal reduces the objective function the most is removed from the solution. This operation is repeated until no feasible station removal exists.

4.4.1.3. Customer Insertion

This operator inserts the customers from the removal list back into the partial solution. We employ greedy, regret-2 and time-based insertion operators, as proposed in Chapter 2. However, some customers may need the insertion of a station to maintain feasibility. In such cases, greedy station insertion operator is applied to insert a station along with the customer.

4.4.1.4. Station Insertion

We use the greedy station insertion presented in Chapter 2. Here we propose a preprocessing procedure that eliminates the dominated stations and speeds up the algorithm. Since the station insertion is performed to make an infeasible customer sequence feasible, the possible insertion locations are known in advance. We can therefore eliminate some stations according to their queueing time and their distances to the customers between which they are inserted. To insert a non-dominated station between customers i and k we proceed as follows: Initially, the set of stations is the same as the original set of stations, i.e. $F_{ik} = F$. Let $j \in F$ and $j' \in F$ be two candidate stations. If $d_{ij} > d_{ij'}$ and $d_{jk} > d_{j'k}$, then j' is preferred to j. However, j may have a shorter queueing time at the time of arrival. Hence, we calculate w_j and $w_{j'}$, the queueing times at stations j and j' if these are visited after customer i. If $w_j > w_{j'}$ then j' is preferred also in terms of the total time spent to visit a station. Then, j is dominated and we eliminate it from F_{ik}. By comparing all the stations pairwise in this fashion, we obtain the reduced set F_{ik}. This procedure is illustrated in Figure 4.5.

Figure 4.5 depicts customers i and k and the stations j, j', j'' and j''' that can be inserted between them. It also shows the EVs waiting at each station at the time of arrival. Comparing j and j', one can see that $d_{ij'} > d_{ij}$ and $d_{j'k} > d_{jk}$. In addition, the waiting time at j' is more than at j since j' has two more EVs. Hence, station j' is dominated by j. For j and j'', clearly $d_{ij} > d_{ij''}$ and $d_{jk} > d_{j''k}$, but j has fewer EVs than j'' in its system. So, we cannot eliminate either of them. Finally, j''' cannot be eliminated since its distance to customer k is shorter than that of j and j''. In this case, the station insertion operator will only evaluate the stations j, j'' and j''' between customers i and k.

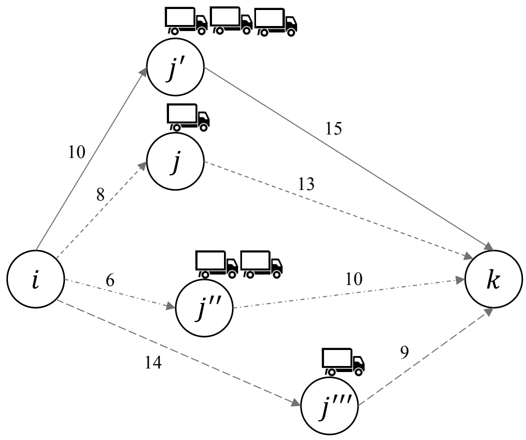


Figure 4.5. Elimination of dominated stations

4.4.2. Fixed Sequence Route Optimization

After every Δ iterations, the best-known solution x_Δ is improved by solving each of its routes to optimality without changing the customer sequence, as in Bruglieri et al., (2016), Montoya et al., (2017), and Koç et al., (2018). The solver optimizes the following decisions: when to visit a station, which station to visit, and how much to recharge.

4.4.2.1. Mathematical Model

Let V be the set of customers to be served along the route. Since the customer sequence is predetermined, we can form the set of consecutive node pairs $\bar{V}$. The set definitions of S, V', V_0, V_{N+1}, $V_{0,N+1}$, V_0', V_{N+1}' and $V_{0,N+1}'$ are the same with the new set V. In this case, 0 and $N+1$ involve single departure and arrival depots. Then, the problem can be represented by a complete graph $G = (V_{0,N+1}', A)$, where $A = \{(i,j) | i,j \in V_{0,N+1}', i \neq j\}$. Some decision variables are the same as in the main formulation presented in Section 4.4, while the new decision variables and sets are shown in Table 4.4.

Table 4.4. Mathematical notation

Sets:	
$\bar{V}$	Set of consecutive customer pairs in the route
V_0	Set of customers in the route and the departure depot
V_{N+1}	Set of customers in the route and the arrival depot
$V_{0,N+1}$	Set of customers in the route and the depots
Decision variables:	
x_{ij}^m	1, if the vehicle departing from customer i, goes to station j and arrives there in time interval m, 0 otherwise
z_{ikl}	1, if SoC at arrival at the station between consecutive customers i and k is between a_{l-1} and a_l, 0 otherwise

Decision variables:

w_{ikl}	1, if SoC at departure from the station between consecutive customers i and k is between a_{l-1} and a_l, 0 otherwise
y_{ik}^a	Battery SoC at a station visited between customers i and k
y_{ik}^d	Battery SoC when departing from a station visited between customers i and k
w_{ik}	Average queueing time of the EV arriving at a station visited between customers i and k
r_{ikl}^a	Coefficient of breakpoint l in the piecewise approximation when EV arrives at a station visited between customers i and k
r_{ikl}^d	Coefficient of breakpoint l in the piecewise approximation when EV departs from a station visited between customers i and k

Given the above definitions, the mathematical model for the problem with time dependent waiting times in stations is formulated as follows:

$$\text{minimize} \quad c_e(0.9q - y_{N+1}) + c_e \sum_{(i,k)\in\bar{V}} \left(y_{ik}^d - y_{ik}^a\right) + c_p \sum_{i\epsilon V} v_i + c_d\tau_{N+1} + (c_o - c_d)v_{N+1} + c_f \tag{4.41}$$

subject to

$$\tau_i + s_i + \sum_{j\in S}\sum_{m\epsilon M} x_{ij}^m\,(t_{ij} + t_{jk}) + w_{ik}(\tau_{ik}) + t_{ik}\left(1 - \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\right) + \left(\sum_{l\epsilon B} r_{ikl}^d c_l - \sum_{l\epsilon B} r_{ikl}^a c_l\right) \le \tau_k \qquad (i,k)\in\bar{V} \tag{4.42}$$

$$t^m \sum_{j\epsilon S} x_{ij}^m \le \tau_i + s_i + \sum_{j\epsilon S} x_{ij}^m t_{ij} \le t^{m+1} + l_0\left(1 - \sum_{j\epsilon S} x_{ij}^m\right) \qquad i\in V_0, m\in M \tag{4.43}$$

$$w_{ik}(\tau_{ik}) \ge W_j^m + s_j^m\left(\tau_i + s_i + t_{ij} - t^m\right) - M(1 - x_{ij}^m) \qquad (i,k)\in\bar{V}, j\in \mathrm{F}, m\in M \tag{4.44}$$

$$w_{ik}(\tau_{ik}) \le M\left(1 - \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\right) \qquad (i,k)\in\bar{V} \tag{4.45}$$

$$y_{ik}^a = \sum_{l\in B} r_{ikl}^a a_l \qquad (i,k)\in\bar{V} \tag{4.46}$$

$$y_{ik}^d = \sum_{l\in B} r_{ikl}^d a_l \qquad (i,k)\in\bar{V} \tag{4.47}$$

$$y_i - h\left(d_{ik}\left(1 - \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\right) + \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\,(d_{ij} + d_{jk})\right) + \left(y_{ik}^d - y_{ik}^d\right) = y_k \qquad (i,k)\in\bar{V} \tag{4.48}$$

$$y_i - h\sum_{j\epsilon S}\sum_{m\epsilon M} d_{ij}x_{ij}^m \ge y_{ik}^a - 0.9q\left(1 - \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\right) \qquad (i,k)\in\bar{V} \tag{4.49}$$

$$y_i - h\sum_{j\epsilon S}\sum_{m\epsilon M} d_{ij}x_{ij}^m \le y_{ik}^a + 0.9q\left(1 - \sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m\right) \qquad (i,k)\in\bar{V} \tag{4.50}$$

$$0.1q\sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m \le y_{ik}^a \le y_{ik}^d \le 0.9q\sum_{j\epsilon S}\sum_{m\epsilon M} x_{ij}^m \qquad (i,k)\in\bar{V} \tag{4.51}$$

$$\sum_{l\epsilon B} r_{ikl}^a = \sum_{l\epsilon B\setminus\{0\}} z_{ikl} = \sum_{m\epsilon M}\sum_{j\epsilon S} x_{ij}^m \qquad (i,k)\in\bar{V} \qquad (4.52)$$

$$r_{ik0}^a \le z_{ik1} \qquad (i,k)\in\bar{V} \qquad (4.53)$$

$$r_{ikl}^a \le z_{ikl} + z_{ik,l+1} \qquad (i,k)\in\bar{V}, l\in B\setminus\{0,b\} \qquad (4.54)$$

$$r_{ikb}^a \le z_{ikb} \qquad (i,k)\in\bar{V} \qquad (4.55)$$

$$\sum_{l\epsilon B} r_{ikl}^d = \sum_{l\epsilon B\setminus\{0\}} w_{ikl} = \sum_{m\epsilon M}\sum_{j\epsilon S} x_{ij}^m \qquad (i,k)\in\bar{V} \qquad (4.56)$$

$$r_{ik0}^d \le w_{ik1} \qquad (i,k)\in\bar{V} \qquad (4.57)$$

$$r_{ikl}^d \le w_{ikl} + w_{ik,l+1} \qquad (i,k)\in\bar{V}, l\in B\setminus\{0,b\} \qquad (4.58)$$

$$r_{ikb}^d \le w_{ikb} \qquad (i,k)\in\bar{V} \qquad (4.59)$$

$$y_i \ge 0.1q \qquad i\in V_{N+1} \qquad (4.60)$$

$$y_0 = 0.9q \qquad (4.61)$$

$$y_i, \tau_i \ge 0 \qquad i\in V_{0,N+1} \qquad (4.62)$$

$$w_{ik}, y_{ik}^a, y_{ik}^d \ge 0 \qquad (i,k)\in\bar{V} \qquad (4.63)$$

$$0 \le r_{ikl}^a, r_{ikl}^d \le 1 \qquad (i,k)\in\bar{V}, l\in B \qquad (4.64)$$

$$z_{ikl}, w_{ikl} \in \{0,1\} \qquad (i,k)\in\bar{V}, l\in B\setminus\{0\} \qquad (4.65)$$

$$13-15$$

The objective function has the same components as in (4.2). Constraints (4.42) keep track of service beginning times for consecutive customers i and k. Constraints (4.43) ensure that if any station is visited after customer i in time period m, then the service start time should lie within time interval $[t^m - t^{m+1}]$. Constraints (4.44) link the flow variables x_{ij}^m and the waiting time $w_{ik}(\tau_{ik})$ at a station between consecutive customers i and k. Constraints (4.46)−(4.47) determine the arrival and departure SoC values at the station visited between consecutive customers i and k in terms of the breakpoints in the piecewise linear approximation. The battery SoC upon arrival at consecutive customers is tracked by constraints (4.48). If any station is not visited, i.e., all x_{ij}^ms are 0, then the SoC upon arrival at customer k will be the level at customer i, minus the energy consumed on arc (i,k). Otherwise, the recharged amount $(y_{ik}^d - y_{ik}^a)$ will be added to that value while the subtracted value will be the energy consumed on arcs (i,j) and (j,k). Similarly, the battery level upon arrival at station j after customer i is determined by constraints (4.49) and (4.50). In this case, the SoC upon arrival at the station and the departure from the station should lie between 10% and 90% of the battery capacity, which is stated by constraints (4.51). Constraints (4.52) and (4.56) establish the relationship between the binary variables z_{ikl} and w_{ikl} and the coefficients of the breakpoints in the piecewise approximation when an EV enters and leaves a station, respectively. Constraints (4.53)−(4.55) and (4.47)−(4.59) ensure that the coefficients related to the piecewise linear approximation of the charging function take correct values. Constraints (4.60) ensure that when arriving at a customer or at the depot, the SoC is at least 10% of the battery capacity. The vehicle departs from the depot with a SoC level equal to 90% the battery capacity, which is ensured by the constraint (4.61). Finally, (4.62)−(4.65) define the domains of the decision variables.

Having explained all components of the proposed matheuristic, its general framework is described in Algorithm 4.1. We denote by $f(x)$ the objective function value of solution x.

Algorithm 4.1. General Framework of the Matheuristic

Generate an initial solution x_0, $x_\Delta \leftarrow x_{best} \leftarrow x_{current} \leftarrow x_0$
Initialize the scores and probabilities of the operators, $iter \leftarrow 1$
while $iter <$ Maximum number of iterations **do**
  if $iter$ is a multiple of Δ and $f(x_\Delta) \neq f(x_{\Delta-previous})$ **then**
    for all routes in x_Δ **do**
      Remove all charging stations from the route
      Solve TD-EVRPTW for a given route, update the route in x_Δ
    end for
    $f(x_{\Delta-previous}) \leftarrow f(x_\Delta), x_{current} \leftarrow x_\Delta$
    if $f(x_\Delta) < f_{best}$ **then**
      $x_{best} \leftarrow x_\Delta, f_{best} \leftarrow f(x_\Delta)$
    end if
    $f(x_\Delta) \leftarrow \infty$
  else
    Select a Customer Removal operator and remove γ customers from $x_{current}$
    Apply *Route Refinement Procedure*
    Select a Customer Insertion operator and repair the solution
    if $f(x_{current}) < f_{previous}$ **then**
      $x_{previous} \leftarrow x_{current}, f_{previous} \leftarrow f(x_{current})$
      if $f(x_{current}) < f_{best}$ **then**
        $x_{best} \leftarrow x_{current}, f_{best} \leftarrow f(x_{current})$
      end if
    else
      Accept the solution using Simulated Annealing Criterion
      $x_{previous} \leftarrow x_{current}, f_{previous} \leftarrow f(x_{current})$
    end if
    if $f(x_{current}) < f(x_\Delta)$ **then**
      $f(x_\Delta) \leftarrow f(x_{current})$
    end if
  end if
  if $iter$ is a multiple of N_c **then**
    Update adaptive weights of CR and CI operators and calculate new selection probabilities
  end if
  $iter \leftarrow iter + 1$
end while
Return x_{best}

4.4.2.2. Preprocessing on Decision Variables for the Fixed Route Formulation

Since we know the customer sequence of the customers in the route, we can eliminate some decision variables. First, for each customer i in the route we calculate the earliest and latest possible departure times, namely $t_{i,early}$ and $t_{i,late}$, using the fact that vehicles depart from the depot at time 0 and have to return to the depot before l_0. Therefore, departing from a customer i earlier than $t_{i,early}$ or later than $t_{i,late}$ is impossible. Using these bounds, we can calculate the earliest and latest arrival times at the stations to be inserted between each customer pair and eliminate the x_{ij}^{m}s corresponding to the time interval outside of the limits just calculated.

4.5. Computational Experiments

This section presents the assumptions and calculations related to the waiting times and reports the results of our computational experiments. All experiments are conducted on an Intel Xeon E5 2.10 GHz processor virtual machine with 16 GB of RAM. The models described in Sections 4.3.2 and 4.4.2.1 as well as the matheuristic were coded in Java and the models were solved by CPLEX 12.6.2 with default settings. We used the EVRPTW instances of Schneider et al. (2014) with some problem-specific adaptations.

4.5.1. Experimental Design

Since we use a non-linear charging function and the EVRPTW data have a fixed charging rate, we need to adapt the rates. We used the piecewise linear charging function for fast chargers proposed in Montoya et al. (2017). The function has three pieces which means that we also need three different charging rates. We apply a scaling such that the recharging rate during the last piece is equal to the rate applied in Schneider et al. (2014) data. Furthermore, we assumed a battery whose SoC interval between 10% and 90% corresponds to the full capacity value used in Schneider et al. (2014). There are three types of instances defined according to the locations of customers, which are random, clustered and random clustered. These types are represented by R, C and RC in the names of the instances. Furthermore, for each instance type, there are two types of time windows which are narrow and wide. The number in the names of the instances begin with 1 or 2 depending on that whether they belong to the narrow or to the wide group.

The day is divided into five time intervals, namely morning, noon, late afternoon, evening, and night. However, since the FCFS property must hold, we need to define transition periods between each interval. It is assumed that the transition periods from a less crowded interval to a crowded interval last 30 minutes. Since the increase in the arrival rate is an outside factor, it is safe to assume that the increase can happen within 30 minutes. However, for the transitions from a crowded interval to a less crowded one, the transition period is bounded by the service time of the charger. The reason is that the number of vehicles that can be served is limited due to the charging capacity at the station. Hence, the ends of these transition periods are variable and they are determined by adding the time to serve the difference of the number of vehicles between the time periods to the end of the previous period. This means that from noon to late afternoon and from late afternoon to evening, the transition periods differ from each other and are not equal to 30 minutes as earlier in the day. After introducing the transition periods, the number of time intervals becomes eight. These intervals are defined as follows: [7:00,7:30), [7:30,9:30), [9:30,10:00), [10:00,15:30), [15:30, t_1), [t_1,19:00), [19:00, t_2), [t_2,7:00). Here, t_1 and t_2 are the ending times of the transient periods, calculated as explained above. The vehicles depart from the depot at 8:00 and must return to the depot by 20:00. Because the deadline of the depot is 20:00, we do not include the night-time interval in our study. Arrivals at the depot after 18:00 are penalized with an overtime wage. Since the benchmark instances are synthetic we convert the time data proportionally to the above setting. In each data we equal due date of the depot to 12 hours and determine the time intervals accordingly. So, the time intervals are different in each data set and scaled according to the due date of the depot.

In order to analyze the effect of the waiting times, we investigate different scenarios among which one does not involve waiting. For the other scenarios, we consider two types of patterns for the waiting times, namely time-independent (TI) and time-dependent (TD) waiting times. In the time-independent case, the waiting times are constant during the day. In the time-dependent case, we assume two types of transitions between time intervals, referred to as smooth and steep transitions. In the former type, the increase in the expected waiting time is smaller than that of the latter during the same transition time from an off-peak interval to a peak interval. Similarly, the decrease from a peak interval to an off-peak interval is smaller compared to the steep type. Whenever there is waiting time, we further use two scenarios where waiting times are short and long. The configurations for the time-dependent case are depicted in Figure 4.6. Note that the expected waiting time doubles from night to morning and from morning to noon, then halves from noon to late afternoon and from late afternoon to evening in the smooth transition case. In the steep transition case, it quadruples from night to morning, increases by a factor of 2.5 from morning to noon and decreases by a factor of two and five from noon to late afternoon and late afternoon to evening, respectively. Note also that the slopes of the transitions from noon to late afternoon and from late afternoon to evening are the same, which is the negative of the service rate. We assign waiting times to each interval assuming a 12-hour planning period. These waiting time values in minutes are given in Table 4.5 and illustrated in Figure 4.6. For each instance the waiting times for each interval are calculated considering the due date of the depot in the data. Since the service rate μ is the same for all time intervals and scenarios, the λ values are calculated accordingly.

Table 4.5. Average waiting time ($\overline{W}$) parameters for each scenario

Scenarios	Length of waiting times	Time Interval			
		Morning	Noon	Late afternoon	Evening
No waiting	-	0	0	0	0
Time-independent	Short	10	10	10	10
	Long	20	20	20	20
Time-dependent/ Smooth transitions	Short	10	20	10	5
	Long	20	40	20	10
Time-dependent/ Steep transitions	Short	20	50	25	5
	Long	40	100	50	10

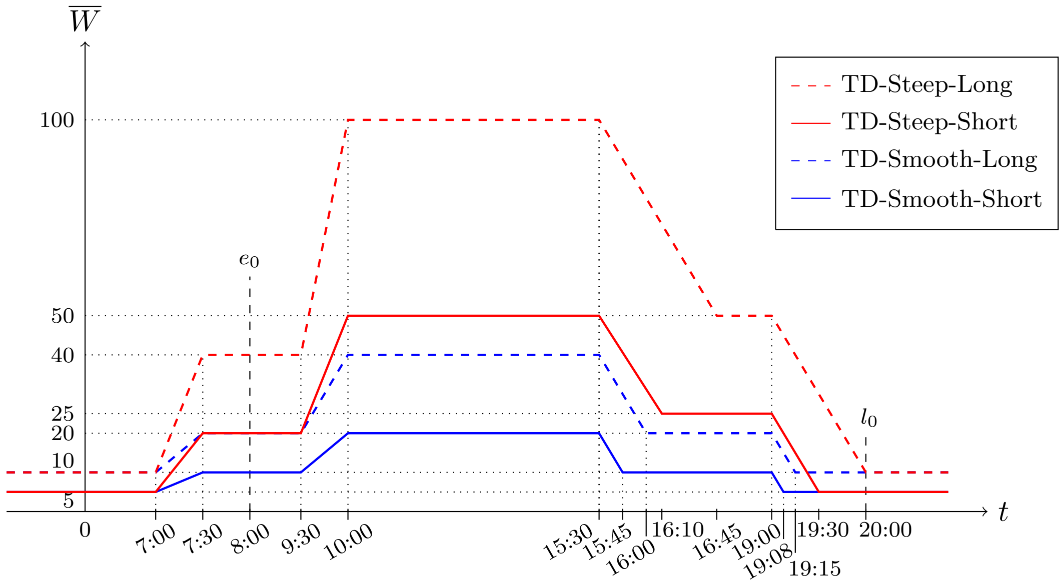


Figure 4.6. Average waiting times for each scenario

We adopt the objective function coefficients of Taş et al. (2013). However, we replace c_e and c_f to represent the cost figures of an EV instead of those of a diesel ICEV. In Feng and Figliozzi (2013), a conventional vehicle and an electric vehicle are compared for their fuel consumptions and purchase prices. The EV is three times more expensive than the conventional vehicle. Furthermore, using the consumption values reported in Feng and Figliozzi (2013) and the current electricity and fuel prices (www.eia.gov), we can conclude that cost per km of the conventional vehicle is 2.5 times higher than that of the EV. Hence, the cost values for our case are determined as follows: $c_e = 0.4, c_p = 1, c_f = 1200, c_d = 1$ and $c_o = 11/6$. In ALNS, we use the same parameter values as reported in Chapter 2.

4.5.2. Results on Small Instances

We first solved small instances with 5, 10 and 15 customers with CPLEX with a time limit of two hours. We used the TD-Steep-Long scenario for these experiments. Table 4.6 compares the performance of CPLEX with that of the matheuristic. The computational times are given in seconds. The value of $\%Imp$ is calculated as $(f_{CPLEX} - f_{Math})/f_{CPLEX}$, where f_{CPLEX} and f_{Math} stand for the solution values of CPLEX and the proposed matheuristic, respectively. The matheuristic was run 10 times and the best results are presented. The last column shows the average computational time. Since the copies of depots and stations should be created and we do not know how many of them will be in the optimal solution, we applied an iterative approach in CPLEX. First, we kept the number of depots fixed and increased the number of stations until the same or a worse optimal solution or an upper bound was obtained. We then repeated this procedure by gradually increasing the number of depots while the solution was improving. The overall best optimal solution or the best upper bound is reported along with its CPU time and the optimality gap. In the five-customer instances and some of the 10-customer instances, CPLEX was able to find optimal solutions within two hours. However, the other instances could not be solved to optimality. The heuristic was able to find an optimal solution for those instances with a zero gap. For the others, it either found the same bound as CPLEX, or improved it. In all instances, the heuristic performed faster than CPLEX.

4.5.3. Results on Large Instances

Desaulniers et al. (2016) highlighted the minor influence of wide time-window constraints on recharging decisions. Hence, we decided to use instances with narrow time windows in our experimental study, and we randomly selected four instances from each data set of C1, R1, and RC1 of the Schneider (2014) instances.

4.5.3.1. The Impact of Waiting on Total Cost and Its Components

We performed experiments for several scenarios of each instance and analyzed the impact of considering different waiting times at the stations by comparing the value of the objective function to that of the base case where waiting is ignored. In Table 4.7, we report the ratio of the corresponding objective function values calculated as $(OFV_{scenario}/OFV_{base})$, where $OFV_{scenario}$ and OFV_{base} are the objective function values of the best solutions over 10 runs. In the headings, “Sm” and “St” stand for Smooth and Steep transition scenarios, respectively, while the last letters “S” and “L” represent short and long waiting time cases, respectively.

Table 4.6. Results on small size instances

	CPLEX			Matheuristic			
Instance	Cost	#Veh	Time	Cost	#Veh	Time	% Imp.
C101C5	4234.59	2	469.70	4234.59	2	3.16	-
C103C5	2494.12	1	5.46	2494.12	1	2.08	-
C206C5	4883.26	1	9.58	4883.26	1	3.38	-
C208C5	3564.84	1	3.23	3564.84	1	1.92	-
R104C5	2770.12	2	3.00	2770.12	2	1.38	-
R105C5	2841.33	2	1.70	2841.33	2	1.02	-
R202C5	1755.46	1	4.74	1755.46	1	2.39	-
R203C5	2001.82	1	3.38	2001.82	1	2.36	-
RC105C5	2954.94	2	5.28	2954.94	2	2.63	-
RC108C5	4215.56	3	4.42	4215.56	3	1.75	-
RC204C5	2032.12	1	12.30	2032.12	1	1.42	-
RC208C5	1728.37	1	4.80	1728.37	1	1.83	-
C101C10	5990.07	2	7200	5342.34	2	8.14	10.81
C104C10	4149.23	2	7200	4149.23	2	22.03	-
C202C10	4929.78	1	7200	4929.78	1	13.14	-
C205C10	5959.2	1	425.56	5959.20	1	7.83	-
R102C10	4375.04	3	7200	4374.54	3	6.26	0.01
R103C10	2887.6	2	7200	2887.03	2	7.19	0.02
R201C10	2695.02	1	7200	2694.18	1	14.53	0.03
R203C10	2178.18	1	7200	2162.87	1	9.62	0.70
RC102C10	5792.04	4	7200	5792.04	4	6.33	-
RC108C10	4458.81	3	2810.71	4458.81	3	5.81	-
RC201C10	3562.49	1	7200	3453.67	1	11.66	3.05
RC205C10	3830.69	2	5986.4	3830.69	2	8.58	-
C103C15	6679.78	3	7200	6674.73	3	23.19	0.08
C106C15	6756.55	3	7200	6756.55	3	15.73	-
C202C15	8379.29	3	7200	7885.32	2	35.24	5.90
C208C15	6630.84	2	7200	6630.84	2	20.85	-
R102C15	5904.56	4	7200	5854.66	4	15.25	0.85
R105C15	5795.65	4	7200	5795.65	4	10.14	-
R202C15	4478.96	2	7200	4009.78	2	37.04	10.48
R209C15	4049.81	2	7200	4025.98	2	36.18	0.59
RC103C15	5786.35	4	7200	5786.35	4	13.65	-
RC108C15	5679.35	4	7200	4592.83	3	7.92	19.13
RC202C15	4393.06	2	7200	4335.84	2	20.96	1.30
RC204C15	3636.58	2	7200	3389.35	1	48.65	6.80

The results show that waiting at stations can increase the total cost by up to 11% and 26% for short- and long-waiting scenarios, respectively. As expected, the total cost is larger in the TD instances compared with the TI instances, when the waiting time is longer, and when the transitions are steeper. The waiting at the stations has a greater impact on the R and RC instances compared with the C instances. This is because the customers are dispersed in the random and random-and-clustered data and the vehicles need more frequent recharges (see Figure 4.7). On average, the total cost increases between 3% and 17% for different waiting scenarios.

Table 4.7. Impact of different waiting schemes on total cost

Instances	TI-S	TI-L	TD-Sm-S	TD-Sm-L	TD-St-S	TD-St-L
C101	1.01	1.03	1.01	1.05	1.08	1.12
C105	1.01	1.02	1.01	1.05	1.08	1.10
C107	1.01	1.02	1.02	1.03	1.04	1.10
C108	1.01	1.02	1.01	1.02	1.03	1.08
Average C	1.01	1.02	1.01	1.04	1.06	1.10
R102	1.07	1.07	1.06	1.09	1.14	1.25
R104	1.01	1.02	1.02	1.10	1.11	1.20
R109	1.08	1.09	1.08	1.10	1.10	1.26
R110	1.01	1.01	1.01	1.03	1.09	1.12
Average R	1.04	1.05	1.04	1.08	1.11	1.21
RC103	1.02	1.04	1.01	1.05	1.10	1.19
RC104	1.08	1.09	1.03	1.10	1.11	1.20
RC106	1.01	1.08	1.08	1.08	1.08	1.18
RC108	1.08	1.08	1.08	1.09	1.10	1.19
Average RC	1.05	1.07	1.05	1.08	1.10	1.19
Average All	1.03	1.05	1.04	1.07	1.09	1.17

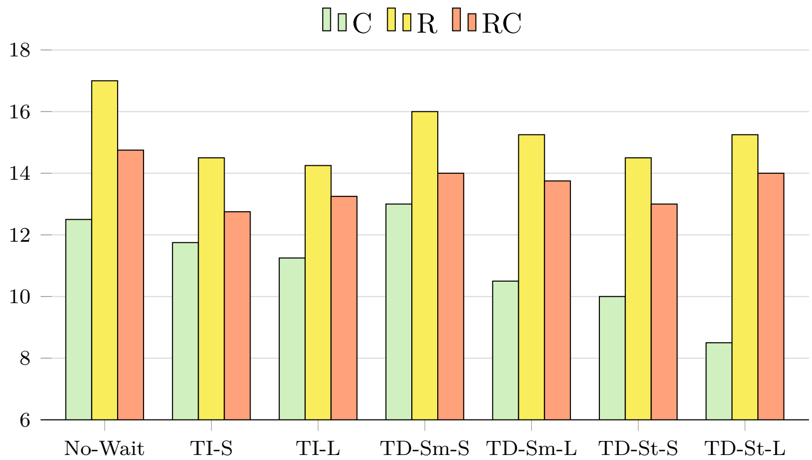


Figure 4.7. Average number of recharges for each data set

Figure 4.8 illustrates the average changes in total cost and cost components compared with the base scenario. The green, yellow, and orange-colored columns represent the average results for the C, R, and RC instances, respectively, whereas the blue-colored columns show the overall averages. In Figures 4.8(a)– 4.8(c) we see that waiting at the stations has a similar effect on the total cost, the cost of the vehicles, and the driver wages. On the other hand, Figures 4.8(d)– 4.8(e) reveal that the overtime wages and cost of late arrivals at customers increase more significantly in the C instances compared with the other two instance types. This is because the fleet size does not increase dramatically in the C instances and the vehicles make longer tours which causes more time window violations. Finally, Figure 4.8(f) shows that the cost of energy is relatively steady compared with the other cost components: the maximum increase is 7% in

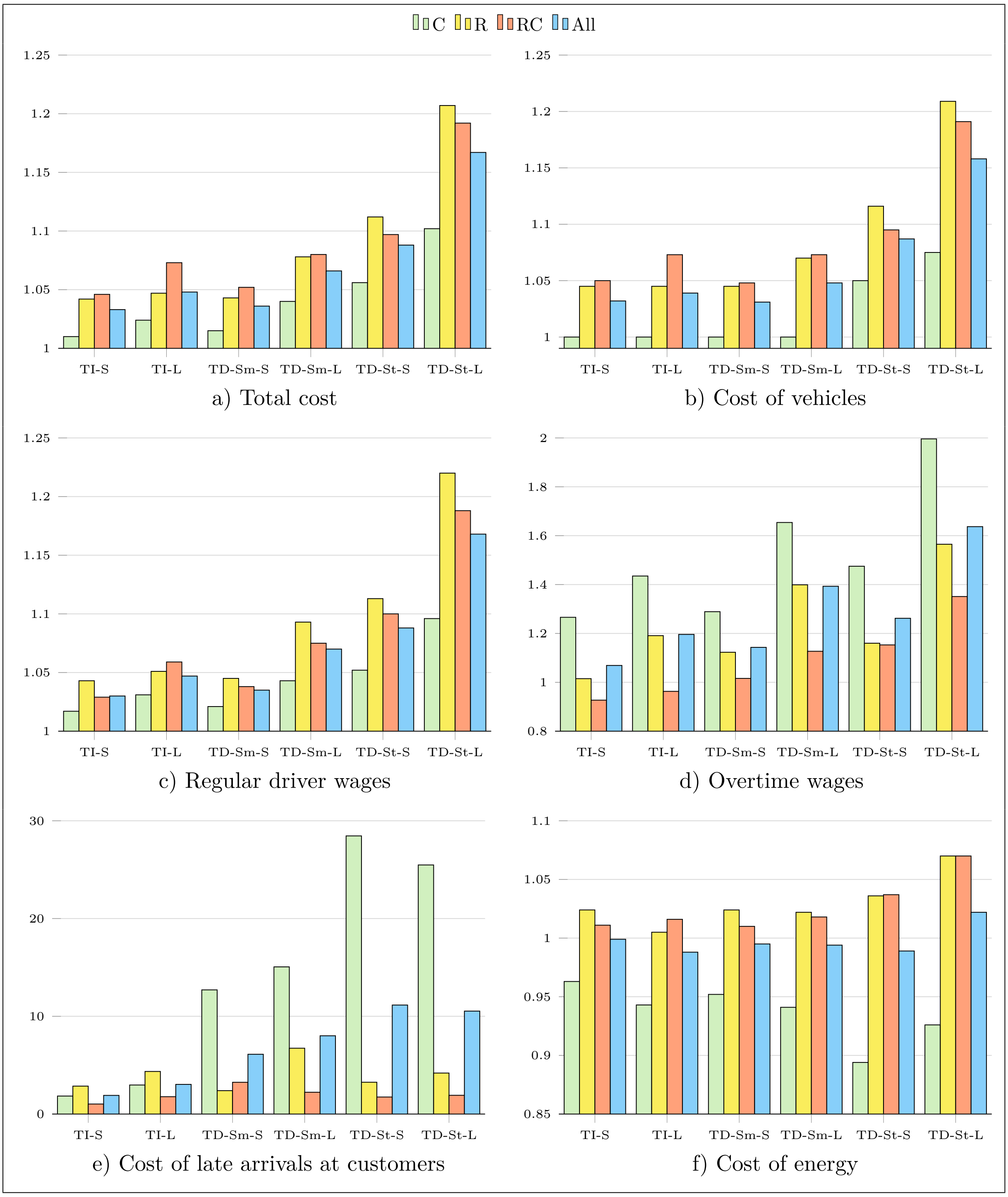


Figure 4.8. Comparison of different cost components for different waiting schemes

the worst case and it even decreases in certain scenarios. In particular, the reduced energy consumption, i.e. total traveled distance, in the C instances is rather surprising. To avoid waiting, planners avoid frequent recharges at the expense of increasing the fleet size. This, however, yields more compact tours and fewer visits between customers located in different cluster zones, hence this reduces the distance traveled.

Figure 4.9 shows the percentage contribution of each cost component to the objective function for the no-waiting and the TD-St-L scenarios. We only report the results of these two extreme cases, but the distributions of the cost components exhibit similar patterns in the intermediate scenarios as well. In Figure 4.9(a), we see that the vehicle cost is the major contributor to the total cost, followed by the driver wages in the R and RC instances. On the other hand, these two cost components are almost equal in the C instances. This difference is due to different time horizons in the data: the due date at the depot is about five times longer in the C instances. Since driver wages are proportional to the total travel time and the vehicles make longer tours, the total cost of wages is higher in the C instances. Figure 4.9(b) depicts results similar to those of Figure 4.9(a) in terms of the relationship between the cost components. However, the shares of cost of late arrivals and overtime wages increase slightly because of the waiting times which cause more delays in arrivals.

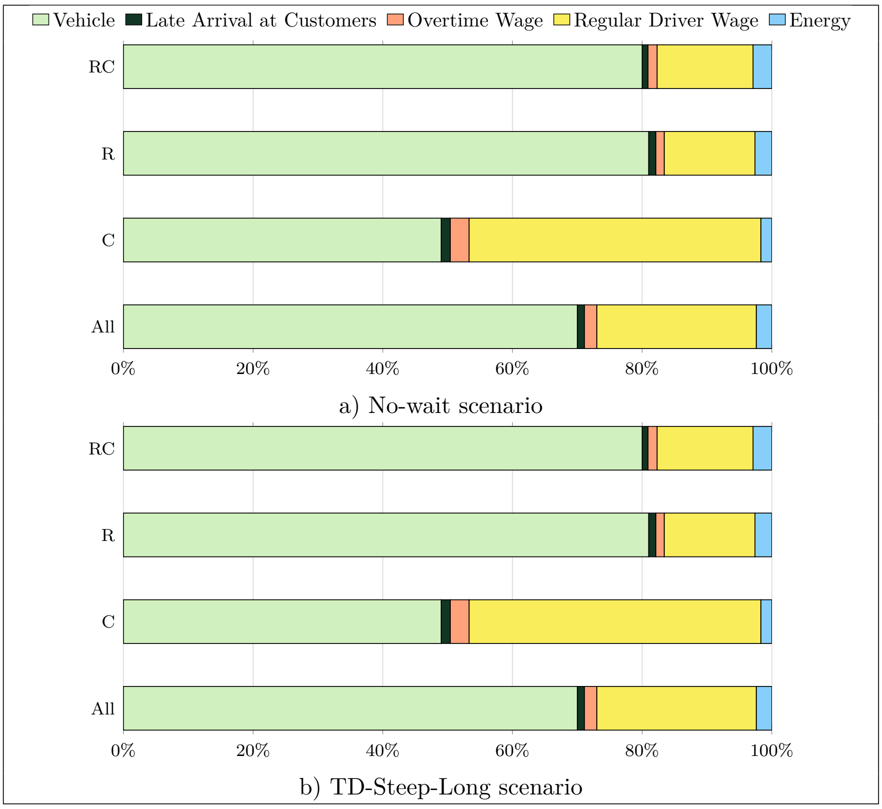


Figure 4.9. Distribution of the cost components for no-wait and TD-St-L scenarios

4.5.3.2. The Impact of Waiting at the Recharging Stations on the Decisions Made in Different Time Intervals

We now investigate how waiting at the recharging stations influences the recharging decisions made in different time intervals and the resulting cost behavior. Figure 4.10 provides a temporal analysis using the average values over all instances. Figure 4.10(a) shows that the vehicles do not recharge much during the morning and the evening. This is expected since they depart from the depot with full charge in the morning and they generally return to the depot without visiting any customer in the evening interval. The average number of recharges illustrated in Figure 4.10(b) shows a similar behavior. However, we observe more frequent recharges in the morning when the waiting is time-dependent. When we compare the short and long waiting cases for each scenario, we see that the number of recharges and the total amount of energy charged decreases during the afternoon when the waiting times are more significant, and increases in the morning and late afternoon intervals when the waiting times are shorter. We also note that both the number of recharges and the amount of energy recharged en route are smaller in the long waiting cases than in the short cases. This unexpected result is due to the increased fleet size in the former case, as depicted in Figure 4.8(b). To avoid long waiting times at the stations and high costs associated with late arrivals at the customers, the algorithm tends to add more vehicles to the fleet. As a result, on average each vehicle makes fewer stops and needs less recharging en route to complete its route. These results are more apparent when we compare the results of the no-wait scenario to those of the TD-St-L scenario. Figure 4.10(c) shows that substantial late arrival costs are incurred when there is more waiting at the stations. Since the vehicles rarely recharge in the morning, we observe almost no time window violations in this time interval. However, the violations are noticeable during the noon and late afternoon hours, particularly in the case of long waiting times. We also observe that some customers are served during the evening with a long delay in the TD-St-L scenario. In other scenarios, the evening interval is used to return to the depot, i.e., no customers are visited. However in this case, visits of some customers have to be postponed to the evening because of long waiting times. This generates late arrival cost in the evening interval, too. Figure 4.10(d) presents the total energy consumption (distance traveled) for each scenario. The total distance traveled is not affected by the changing waiting times, but the solutions differ in the number of vehicles and recharge schedules.

4.5.3.3. Sensitivity of the Solutions to the Late Arrival Penalty

We observed in Figure 4.8(e) that the cost of late arrivals at customers may increase dramatically in C-type instances. We therefore performed additional tests on these instances by increasing late arrival penalties 5-, 50-, and 100-fold to investigate the sensitivity of the results. Table 4.8 reports the average cost values and the average percentage change in each cost component for each late arrival penalty setting. The columns %Δ report the percentage difference in each component compared with the base case where the unit late arrival penalty is one. While almost all cost components increase, there is a significant decrease in late arrival and overtime costs as well as in the total lateness. As the late arrival penalty increases, the fleet size also increases to avoid late arrivals at customers. As a result, overtime wages decrease as well.

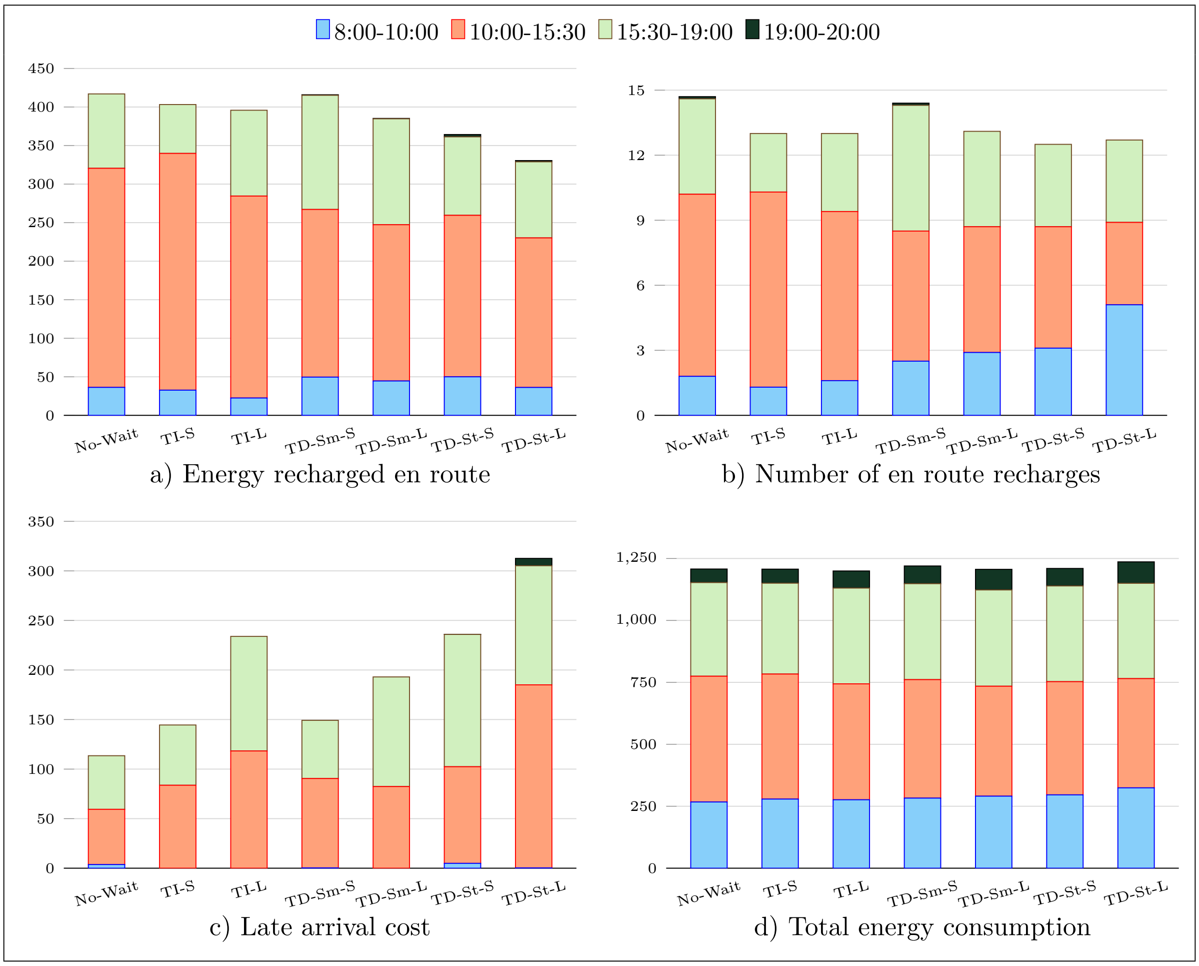


Figure 4.10. Temporal analysis of recharging decisions and costs

Table 4.8. Comparison of results for different late arrival penalty values

	Penalty						
	1	5		50		100	
	Value	Value	%Δ	Value	%Δ	Value	%Δ
Total Cost	24,610	25,603	4.03	26,511	7.72	26,766	8.76
Vehicle Cost	12,214	13,029	6.67	13,757	12.63	13,886	13.68
# of Vehicles	10.2	10.9	6.67	11.5	12.63	11.6	13.68
Late Arrival Cost	228	108	-52.36	28	-87.87	24	-89.5
Total Lateness	228	22	-90.47	0.6	-99.76	0.2	-99.89
Overtime Wage	627	564	-9.91	532	-15.15	532	-15.13
Driver Wage	11,077	11,421	3.1	11,709	5.7	11,827	6.77
Energy Cost	465	481	3.56	486	4.61	498	7.11

4.5.3.4. Computational Times

We now investigate the effect of different waiting scenarios on the computation times. Table 4.9 reports the average computational times of 10 runs for each scenario and data type, as well as the overall average of all type-1 data. The results reveal that the C instances require longer run times whereas RC instances are solved the fastest. Although the run time does not increase from short to long waiting cases, there is a noticeable rise from the no-wait to the TD-St-L scenario.

Table 4.9. Computation times for various waiting schemes (in minutes)

Data Types	No-Wait	TI-S	TI-L	TD-Sm-S	TD-Sm-L	TD-St-S	TD-St-L
C	99.2	124.5	108.0	111.1	119.2	126.7	132.8
R	95.2	84.9	84.2	104.0	93.6	102.7	91.0
RC	76.3	72.6	72.2	76.7	81.8	60.8	64.2
Average	90.2	94.0	88.1	97.3	98.2	96.7	96.0

As can be seen from Table 4.9, the computational times are relatively long. We therefore performed a sensitivity analysis to investigate the effect of the number of iterations on solution quality. Table 4.10 provides the percentage deterioration in solution quality and the reduction in run time for different numbers of iterations. The results are obtained by using the average values of all type 1 data considering all scenarios. The first column gives the number of iterations which are the breakpoints for the comparison. The second column shows the percentage deterioration in the objective function value at that iteration compared with that of the final solution obtained after 25,000 iterations. For instance, the cost of best-found solution at iteration 5,000 is 8.2% worse than the global best solution achieved after 25,000 iterations. The third column shows the percentage of the time spent at the corresponding breakpoint. For instance, the first 5,000 iterations take 14.8% of the total run time of 25,000 iterations. These results indicate that the algorithm could have been stopped after 15,000 iterations with only a 0.5% worsening in solution quality, and saving nearly half of the computational time. However, since we aimed at obtaining best benchmark results at the expense of longer run times, we carried our detailed experimental study with 25,000 iterations.

Table 4.10. Sensitivity of solution quality and run time to number of iterations

# of Iterations	%Deterioration	Relative Run Time
25,000	-	100.0%
20,000	0.1%	76.7%
15,000	0.5%	54.6%
10,000	2.5%	34.0%
5,000	8.2%	14.8%

4.5.3.5. Computations on R2 Type Instances

We used type 1 instances in our experiments since they have narrow time windows and we can better observe the effect of waiting times on routing decisions compared with the type 2 instances with wide time windows, where respecting the customer time windows is less of a concern. We also performed a limited set of experiments on R2 instances to gain some insights about the effect of the waiting times in such an environment. Table 4.11 presents a comparative analysis of results for R1- and R2-type data based on average cost figures across all waiting scenarios. Similar to our observations on the R1 instances, the total cost and its components in R2 instances increase when waiting is longer, both in the TI and TD settings, except for the cost of energy which does not display a discernible pattern. On the other hand, there are some fundamental differences between the two sets of results. First, the fleet size is not affected by waiting in the R2 instances and the same number of vehicles is used in all scenarios. We therefore conclude that the size of the customer time windows has a significant influence on the fleet size. Second, the costs of late arrivals and overtime wages are much smaller in the R2 instances than in the R1 instances in all scenarios. This can be considered as an expected consequence of the wider time windows. Even though late arrivals at customers and at the depot still exist, their magnitudes are negligible compared with those of the R1 instances. Finally, we see that the total travel distances are not affected by the size of the time windows as the total energy cost figures for each scenario are similar in both data types.

Table 4.11. Comparison of results for R1 and R2 instances

Type 1 Instances	No-Wait	TI-S	TI-L	TD-Sm-S	TD-Sm-L	TD-St-S	TD-St-L	Average
Total Cost	18,499	19,039	19,323	19,119	19,647	20,054	21,427	19,587
# of Vehicles	10.4	10.8	10.8	10.8	10.9	11.3	12.1	11
Late Arrival Cost	136.6	145.1	221.8	181.7	304.1	185.9	257	204.6
Overtime Wage	287.1	304.4	336.5	322.8	390.8	348.6	428.4	345.5
Driver Wage	5,091	5,205	5,286	5,232	5,371	5,440	5,747	5,339
Energy Cost	484.8	484.2	478.7	482.8	481.2	479.1	495.2	483.7
Type 2 Instances								
Total Cost	6,719	6,738	6,745	6,763	6,839	6,775	7,144	6,818
# of Vehicles	3	3	3	3	3	3	3	3
Late Arrival Cost	29.9	10.4	46.1	25.8	53.6	50.7	207.3	60.6
Overtime Wage	150.1	138.8	143.5	149	155.6	156.1	225.3	159.8
Driver Wage	2,461	2,501	2,502	2,530	2,569	2,530	2,680	2,539
Energy Cost	478	488.6	453.6	458.8	461.2	438.4	432.1	458.7

4.6. Conclusions

We have introduced the Electric Vehicle Routing Problem with Soft Time Windows and Time-Dependent Waiting Times at Recharging Stations (TD-EVRPSTW), where recharging stations have limited capacities and EVs may queue before being serviced. We considered five time intervals in a day (morning, noon, late afternoon, evening, and night) with different queue lengths. We used the M/G/1 queueing system equations to estimate the waiting times in each time interval. The EVs are allowed to serve the customers beyond their late service time by paying a penalty proportional to the length of the delay. Similarly, the EVs may return to the depot later than the depot closing time by paying overtime wages to the drivers. The recharging time is a non-linear function of the energy transferred and is approximated using a piecewise linear function. We formulated this problem as a mixed integer linear program and developed a matheuristic that couples ALNS with an exact solver. The ALNS uses known operators with problem-specific modifications, as well as a new customer removal operator. The routes obtained by the ALNS are enhanced by optimizing the recharging-related decisions while keeping the sequence of the visited customers fixed.

To test the performance of our method we adapted benchmark instances by considering six scenarios with different waiting characteristics. On small-size instances our method outperformed CPLEX both in solution quality and computational time. Since we do not have any benchmark results for the large instances, we provided managerial insights based on the best solutions we achieved. The results showed that waiting times may be crucial in routing decisions and they should be taken into account to compute feasible and better route plans. The distribution of the cost components in the objective function remained similar from a scenario to another. When the waiting times increase, the number of vehicles also increases to avoid long routes and frequent visits to recharging stations. Similarly, station visits during the crowded time intervals decrease while the total distance travelled does not change much in all cases. The increase in unit late arrival costs results in a larger fleet size and in a higher total late arrival cost. Finally, the waiting times do not affect the solutions significantly when the customer time windows are wide.

Chapter 5

Electric Vehicle Routing Problem with Time Windows and Stochastic Waiting Times at Recharging Stations

5.1. Introduction

Most of the papers in the literature assume that an EV can be recharged as soon as it arrives at a station. However, in practice, the chargers may be occupied and there may exist other EVs waiting for service. Hence, the EV may need to queue for some time before it starts recharging its battery. If the recharging stations are owned by the fleet operators, then the charging operations may be scheduled to prevent possible conflicts (Ding et al., 2015; Froger et al., 2017). On the other hand, if the stations are public, it is challenging to foresee the congestion in the stations.

In the stochastic Vehicle Routing Problems (VRPs), some parameters of the problem are not known in advance. The unknown parameters can be the customer demands, travel times, service times or customers' presence. The problems can be modeled as chance constrained programs or multi-stage stochastic programs. In the former approach, it is ensured that probabilistic constraints are satisfied with a probability greater than a threshold. However, there is not a recovery phase for the solutions having those constraints satisfied with a probability less than the threshold. In the latter approach, an a priori plan is constructed firstly. Then, the stochastic parameters are revealed through time and the decisions are made accordingly to reoptimize the initial solution for the realized data.

When the demands of the customers are unknown, for instance, after they are realized, the capacity of the vehicle may not be sufficient to perform the pickup or delivery of the subsequent customers in the route if the realized demands are more than expected. In this case, some corrective actions should be taken, i.e., the vehicle may return to the depot either to load goods for delivery or empty the vehicle to create space for pickup. When the travel times or service times are stochastic, the vehicles may not catch the time windows of the customers or the depot if these times are longer than expected. In this case, the corrective action may be skipping the customers whose time windows cannot be caught or allowing late arrivals with penalty. Finally, when the customers' presence is stochastic, there may exist some customers to be removed from or to be inserted to the planned solution. Then, the routing decisions may be altered accordingly. The objective function of the stochastic VRPs generally minimizes the transportation costs and the expected recourse costs.

In this study, we extend the EVRPTW by considering stochastic waiting times at the recharging stations. Stations are equipped with single chargers and partial recharging is allowed. We use an $M/G/1$ queuing system to model the waiting times, based on the assumptions discussed in Chapter 4. The problem is modeled as a two-stage stochastic program with recourse using scenarios. To calculate the probabilities and the expected costs, simulation is used. We propose an ALNS approach to solve the problem efficiently. We employ ALNS to solve both the first and the second stage problems.

The remainder of the chapter is organized as follows: Section 5.2 explains the related studies in the literature. Section 5.3 describes the problem and gives the mathematical formulations. Section 5.4 provides the details of the proposed solution method, while Section 5.5 presents the computational study. Finally, Section 5.6 concludes the study with some remarks.

5.2. Related Literature

Within the context of VRP, the stochasticity may be related to customer demands, service times, travel times, and customers' presence. Since our case is similar to that of stochastic service times at customers, we review the related literature and refer the interested reader to Gendreau et al. (2016) for a detailed survey of stochastic VRP. Sungur et al. (2010) study courier delivery problem with soft time windows. They use scenario-based stochastic programming with recourse and robust optimization to model the uncertainty in customers' presence and service time uncertainty, respectively. They model service times as lognormal random variables and their recourse action is skipping or inserting customers according to their presence and paying penalties for late arrivals. They propose a Tabu Search (TS) algorithm to solve the problem.

Zhang et al. (2013) address the stochastic VRP considering random travel and service times. Their recourse action is to allow late arrivals at the customers and the depot with penalty. They employ Iterated TS as the solution method. Li et al. (2010) consider a similar problem but they do not allow servicing customers earlier than their service start times. They use symmetric triangular distribution to model the service times and employ stochastic simulation to evaluate the objective function and calculate the probabilities. Their recourse action is to allow late arrivals with penalties. They also propose a TS solution method. Lei et al. (2012) study the capacitated VRP (CVRP) with stochastic service times and service times are assumed to be normal random variables. Their recourse action is also allowing late arrivals at the depot with penalty. They propose a Variable Neighborhood Search (VNS) approach to solve the problem.

Chen et al. (2014) study a capacitated arc routing problem with stochastic service times in the road network daily maintenance operations. They introduce confidence levels which correspond to the probability with which the stochastic constraint will hold. A chance-constrained programming model is developed and solved using branch-and-cut algorithm. Then, the problem is modelled as a two-stage stochastic programming with recourse where the recourse action is terminating the route where failure occurs, and the recourse cost is the penalty associated with the skipped customers. They present an ALNS algorithm to solve this problem.

Binart et al. (2016) consider the case of optional customers with the objective of minimizing total travel time while maximizing the optional customers visited. These visits are planned in the second stage. The random service times are drawn from a symmetric triangular distribution and dynamic programming is used to solve the problem. Errico et al. (2016) consider hard time windows in the presence of stochastic service times. The recourse action is skipping the current or the following customer with a penalty when infeasibility occurs. They also use symmetric triangular distribution for random service times and develop a branch-cut-and-price algorithm to solve the problem. Shi et al. (2018) study the delivery and pickup problem in home healthcare where the time windows are hard in the first stage and are relaxed with penalty in the second stage. Hence, the recourse action is allowing late arrivals at the customers and the depot with penalties. They use normal distribution to model the service times and employ simulation to evaluate the objective function and calculate the probabilities. They propose several heuristic methods such as Hybrid Genetic Algorithm, Simulated Annealing, and Bat and Firefly Algorithm.

Recently, Bruglieri et al. (2018) study the waiting times at the stations within the context of GVRP. AFVs are routed such that their refueling do not overlap in the stations. They minimize total distance and propose an exact method in which the routes are considered as composition of paths. Pelletier et al. (2018) consider a similar problem using EVs. They assume that recharging is performed only at the depot and develop a model to plan the depot charge schedules also considering the realistic charging process, time-dependent energy costs, battery degradation, grid restrictions, and facility-related demand charges. Ding et al. (2015) consider limited charging capacity in EV recharging stations to determine conflict-free routes. Their objective minimizes the total distance traveled and they propose a heuristic method which combines VNS and TS to solve the problem. Froger et al. (2017b) solve a similar problem in which the stations have limited number of chargers and an EV may need to wait before recharging if the chargers are busy recharging other EVs in the fleet. In this problem, the use of the chargers depends on the routing and charging decisions. They formulate the problem as a mixed integer linear program and propose a route-first assemble-second matheuristic to solve it. Their objective function minimizes the total time which includes driving, service, and charging times. Finally, Chapter 4 studies the time-dependent waiting times at the recharging stations where the objective is to minimize the total cost of drivers, EVs, energy, and late arrival penalties for customers. In this problem, the queue lengths at the recharging stations vary by the time of the day following different distributions. The study approximates the waiting times using their expected values during different time intervals and proposes an ALNS-based matheuristic approach to solve it. Kullman et al. (2018) model the uncertain availability of stations as a Markov decision process using $M/M/\psi_c$ queueing system and implement a stochastic dynamic programming method. Their objective is to minimize the total expected time consisting of travel, charging, and queueing times.

5.3. Problem Description and Formulation

The problem establishes a set of routes which are operated by a homogeneous fleet of EVs. Routes should cover all customers which have known demands and time windows. The customers should be visited within their time windows. If an EV arrives before the early service time, it waits until that time. On the other hand, arriving later than the late service time is not allowed. All EVs depart from the depot and should return to the depot before its due date. Furthermore, the SoC of the EVs should be nonnegative throughout the journey. The EVs may visit recharging stations to recharge their batteries in order to continue their routes. The crucial point of the problem is the waiting times at the stations before the recharging service, which are random variables. They are revealed at the time when an EV arrives at a station. Hence, if the waiting time is too long, the EV may not catch the time windows of the subsequent customers in the preplanned route, which is constructed before the EV begins its journey, and the route may become infeasible. In this case, some recourse actions should be taken to correct the solution and make it feasible. The problem can be formulated as a two-stage stochastic programming model. In the first stage, a set of routes is determined using expected waiting times at the stations. Next, the random queue times at the recharging stations are realized. Then, the second stage solution is the set of routes after the recourse actions are applied. The objective function minimizes the sum of the total first stage cost which includes vehicle acquisition cost, driver cost, energy cost, and the second stage cost corresponding to the expected cost of the recourse decisions. To keep the nature of the recourse action and the expected cost calculations simple, we assume that an EV may visit a recharging station at most once during its journey.

This is not a strong assumption considering the fact that waiting and recharging at station may take considerable amount of time during which the EV will remain idle.

5.3.1. Formulation of the First Stage Problem

Let V, F and K denote the set of customers, the set of recharging stations and the set of available EVs, respectively. We denote the depot as 0 or $N+1$ if it is at the beginning or at the end of a route. We define the sets $V_0 = V \cup 0$, $V_{n+1} = V \cup (n+1)$, $V' = V \cup F'$, $V_0{}' = V_0 \cup F$, and $V'_{n+1} = F \cup V_{n+1}$. Each arc (i,j) has a distance d_{ij} and a travel time t_{ij}. The energy is consumed at a rate of h and each traveled arc (i,j) consumes $h.d_{ij}$ of the remaining energy. The battery is recharged at a rate of g, i.e., one unit of energy is transferred in g time units. Each customer $i \in V$ has positive demand q_i, service time s_i and time window $[e_i, l_i]$. The cargo and battery capacities of the EVs are C and Q, respectively. When EV k arrives at recharging station i, it waits in the queue for ω_i^k time units, which is a random variable. The objective function consists of the total cost of energy, driver wages and the acquisition cost of EVs. c_e, c_d and c_f denote the unit energy cost, the cost of drivers per unit time and the fixed cost of the EVs' acquisition, respectively. $\widetilde{\omega}_i$ represents the expected waiting time at recharging station i. The decision variables, τ_i^k, u_i^k and y_i^k keep track of the service starting time, remaining cargo level and remaining SoC level at vertex $i \in V'_{n+1}$ visited by vehicle k, respectively, whereas the SoC level of EV k at the departure from station $i \in F$ is tracked by variables Y_i^k. Finally, binary decision variable x_{ij}^k takes value 1 if arc (i,j) is traversed by EV k and 0 otherwise. The notation of the first-stage model is provided in Table 5.1.

Table 5.1. Notation for the first-stage problem

Sets:	
V	Set of the customers
F	Set of the recharging stations
K	Set of the available EVs
V_0	Set of the departure depot and the customers
V_{n+1}	Set of the customers and the arrival depot
V'	Set of the customers and the recharging stations ($V \cup F$)
V'_0	Set of the departure depot, customers and stations ($V' \cup 0$)
V'_{n+1}	Set of the customers, stations, and the arrival depot ($V' \cup (N+1)$)
Parameters:	
d_{ij}	Distance from vertex i to vertex j
t_{ij}	Travel time from vertex i to vertex j
q_i	Demand of customer i
s_i	Time required to serve customer i
$[e_i, l_i]$	Service time window of customer i

Parameters:

C	Cargo capacity of the vehicles
Q	Battery capacity of the vehicles
$\widetilde{\omega}_i$	Expected waiting time at station i
c_e	Unit energy cost
c_d	Driver wage per unit time
c_f	Fixed vehicle acquisition cost

Decision variables:

τ_i^k	Service start time of vehicle k at vertex i
u_i^k	Remaining cargo in vehicle k at vertex i
y_i^k	Battery SoC of vehicle k upon its arrival at vertex i
Y_i^k	Battery SoC of vehicle k at its departure from station i
x_{ij}^k	1 if vehicle k traverses arc (i,j); 0 otherwise

The mathematical model of the first stage problem is formulated as follows:

$$f(x,\tau,y,Y) = minimize\ \ c_e \sum_{i\in V_0'} \sum_{j\in V_{n+1}'} \sum_{k\in K} d_{ij} x_{ij}^k + c_d \sum_{k\in K} \tau_{n+1}^k + c_f \sum_{j\in V'} \sum_{k\in K} x_{0j}^k + \sum_{k\in K} E[Q^k(x,\xi(\omega))] \tag{5.1}$$

subject to

$$\sum_{j\in V_{n+1}'} \sum_{k\in K} x_{ij}^k = 1 \qquad i \in V \tag{5.2}$$

$$\sum_{j\in V_{n+1}'} x_{ij}^k \le 1 \qquad i \in F, k \in K \tag{5.3}$$

$$\sum_{i\in V_0'} x_{ij}^k = \sum_{i\in V_{n+1}'} x_{ji}^k \qquad j \in V', k \in K \tag{5.4}$$

$$\tau_i^k + (t_{ij} + s_i)x_{ij}^k - l_0(1 - x_{ij}^k) \le \tau_j^k \qquad i \in V_0, j \in V_{n+1}', k \in K \tag{5.5}$$

$$\tau_i^k + t_{ij}x_{ij}^k + g(Y_i^k - y_i^k) + \widetilde{\omega}_i - M(1 - x_{ij}^k) \le \tau_j^k \qquad i \in F, j \in V_{n+1}, k \in K \tag{5.6}$$

$$e_j \sum_{i\in V_0'} x_{ij}^k \le \tau_j^k \le l_j \sum_{i\in V_0'} x_{ij}^k \qquad j \in V_{n+1}, k \in K \tag{5.7}$$

$$0 \le u_j^k \le u_i^k - q_i \sum_{k\in K} x_{ij}^k + C\left(1 - \sum_{k\in K} x_{ij}^k\right) \qquad i \in V_0', j \in V_{n+1}' \tag{5.8}$$

$$0 \leq u_j^k \leq C \sum_{i \in V_0'} x_{ij}^k \qquad j \in V_{n+1}', k \in K \tag{5.9}$$

$$0 \leq y_j^k \leq Y_i^k - (h \cdot d_{ij}) x_{ij}^k + Q(1 - x_{ij}^k) \qquad i \in V_0', j \in V_{n+1}', k \in K \tag{5.10}$$

$$0 \leq y_j^k \leq y_i^k - (h \cdot d_{ij}) x_{ij}^k + Q(1 - x_{ij}^k) \qquad i \in V, j \in V_{n+1}', k \in K \tag{5.11}$$

$$y_j^k \leq Y_j^k \leq Q \sum_{i \in V_0'} x_{ij}^k \qquad j \in V_{n+1}', k \in K \tag{5.12}$$

$$\sum_{i \in F} \sum_{j \in V_{n+1}'} x_{ij}^k \leq 1 \qquad k \in K \tag{5.13}$$

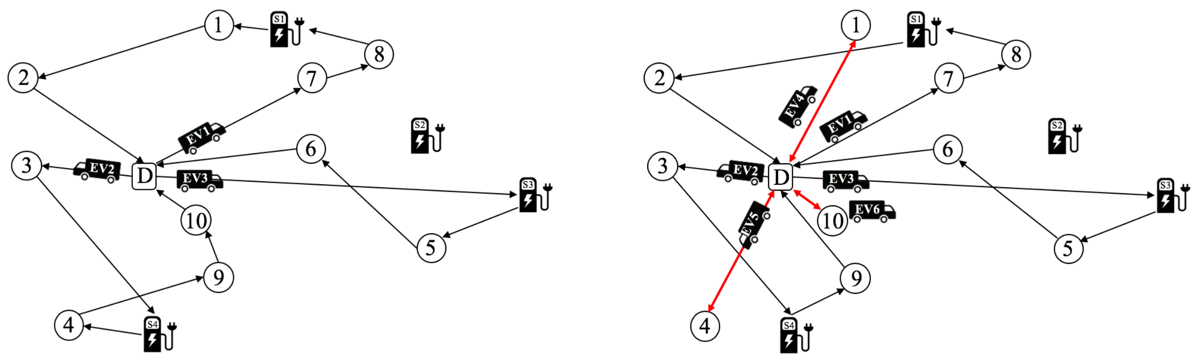
$$x_{ij}^k \in \{0,1\} \qquad i \in V_0', j \in V_{n+1}', k \in K \tag{5.14}$$

The objective function minimizes the sum of energy cost, vehicle acquisition cost, driver wage, and expected recourse cost. Constraints (5.2) ensure that each customer is visited exactly once whereas constraints (5.3) guarantee that each station is visited at most once by each EV. Constraints (5.4) ensure the connectivity of all vertices. Service start times of the EVs at a vertex after departing from the depot or any customer are tracked by constraints (5.5) while the service start times at a customer or arrival time at the depot after departing from a station are tracked by constraints (5.6). Constraints (5.7) make sure that service start times are within the time windows of the vertices. Constraints (5.8) and (5.9) observe the load on the EVs and ensure that total load does not exceed the cargo capacity of the vehicle. Battery SoC of the EVs are tracked by constraints (5.10) and (5.11). Constraints (5.12) set the lower and upper bounds of the SoC at the departure from a vertex. Constraints (5.13) limit each EV with a single recharge during its journey. Finally, constraints (5.14) define the binary decision variables.

$\widetilde{\omega}_i$ in constraints (5.6) represents the expected waiting time at recharging station i. Hence, in some realizations, it may lead to infeasibility in terms of time windows of the customers or the depot if the realized time is longer than the expected value. In this case, a corrective action should be taken in order to service the customers whose time windows are violated.

5.3.2. Recourse Action

When an EV visits a recharging station, the random waiting time in the queue is realized. If it is longer than the expected value, then the time windows of some customers visited following the station may not be caught if the EV continues the journey using the sequence of the first stage solution. In this case, a decision should be made to select a subset of the customers visited after recharging in the first stage solution. The customers in this subset are removed from the route and serviced by using additional EVs. This selection is made such that the remaining customers can be visited by satisfying the time window constraints. However, servicing the customers from the selected subset will incur a special service cost which includes serving each customer by a separate EV. On the other hand, since the route length may decrease due to removal of these customers, the driver wage and energy recharged at the station may decrease. Figure 5.1 illustrates the solutions of the first and second stage problems using a small instance involving 8 customers and 4 stations. Considering the expected waiting times at the stations, the first stage solution involves 3 routes. However, after the random times are realized, some customers become infeasible to visit because of the waiting times which happen to be longer than expected. EV1 faces a long queue in S1 and it cannot catch the time window of customer 1. So, it decides to skip that customer and continue its route with customer 2 which can still be visited. Hence, for EV1 the recourse cost includes serving customer 1 with a separate EV minus the savings from the energy cost and driver wage since the route involves one less customer. However, EV2 has to skip both customer 4 and customer 10 paying special service penalty for these two customers. Finally, EV3 does not make any changes in its route since the actual waiting time at S3 allows visiting customers 5 and 6 within their time windows. Hence, recourse cost for the third route is zero.



a. Optimal solution of the first stage problem b. Solution of the second stage problem

Figure 5.1. Illustration of the recourse action and the resulting solution after the recourse

Sometimes, although the infeasible customers are skipped from the solution, the EV may still be late for the depot. This happens especially when the EV visits a recharging station right before returning to the depot. In this case, late arrival to the depot is allowed with a penalty which is an overtime wage paid to the driver. Hence, the recourse cost includes also the overtime cost. Figure 5.2 illustrates this case for a route which includes 4 customers. The EV arrives at the depot at time 100 which is later than the due date, 90. Since the station is visited just before the depot, this recharging should be made to return to the depot and cannot be skipped. Hence, the only corrective action in this case would be allowing late arrival to the depot with a penalty.

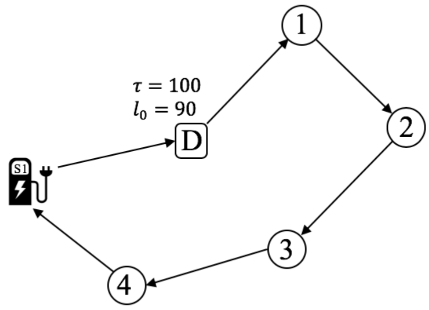
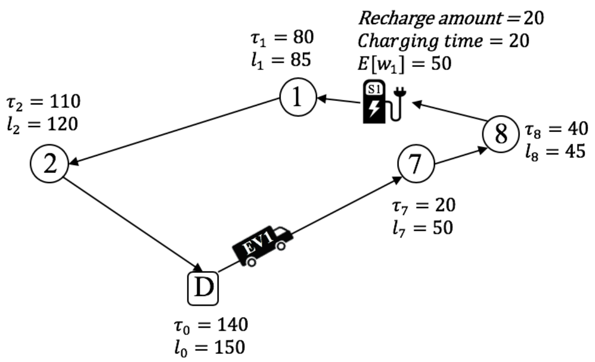
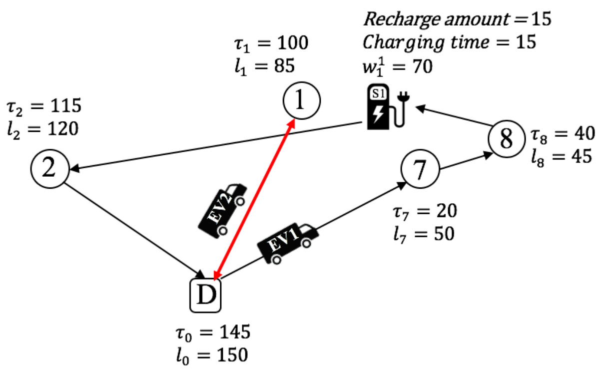


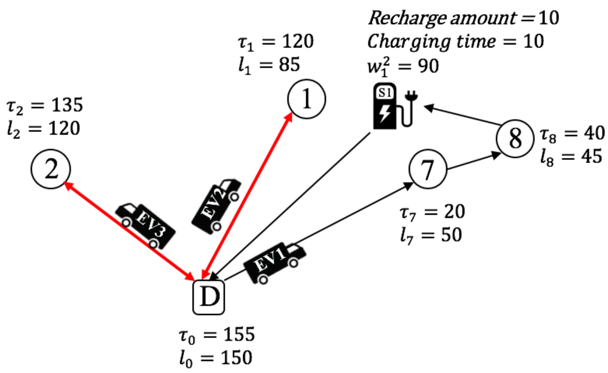
Figure 5.2. A route in which the EV arrives at the depot later than its due date



a. First stage solution



b. Second stage solution of scenario 1



c. Second stage solution of scenario 2

Figure 5.3. First stage solution and the second stage solutions of two different scenarios

Figure 5.3 shows the details of the first route in Figure 5.1 as well as two different final solutions according to two different scenarios. The first stage solution is constructed considering the expected waiting time at the station, which is 50, and is illustrated in Figure 5.2.a. In the first scenario given in Figure 5.2.b, the realized waiting time is 70, 20 units longer than expected. Since customer 1 has only 5 units of slack time, the EV cannot arrive at this customer before its late service time. So, customer 1 is serviced by another vehicle and the EV continues its route with customer 2. Because customer 1 is skipped, there is a decrease in the energy needed to return to the depot and the recharge amount at station 1 is decreased accordingly. The recharging time also decreases and the EV arrives at customer 2 and the depot on time.

In the second scenario, as shown in Figure 5.2.c, the actual waiting time is 40 time units longer than expected and is equal to 90. Customer 1 is again skipped since the arrival time would be 120, which is later than its late service time of 85. If the EV travels to customer 2 from the station, then the recharge amount is decreased since it will not visit customer 1. Considering the time saved from the recharging, the arrival time at customer 2 is updated to 135, which is still later than its late service time. Hence, customer 2 is also skipped and the vehicle directly returns to the depot following the station. In this case, the recharge amount is again decreased since one more customer is removed from the route. On the other hand, since the vehicle arrives at the depot 5 time units after its due date, an overtime wage is incurred. In this scenario, the recourse cost is then serving customers 1 and 2 with separate vehicles, overtime wage paid to the driver minus the cost of 10 units of energy saved by skipping customers 1 and 2.

5.3.3. Formulation of the Second Stage Problem

When the first stage problem is solved, we have the optimal solution and the second stage model reoptimizes this solution route by route considering the actual waiting times. Let i_k^* be the index of the recharging station visited in route k. Let V_k denote the set of customers visited by vehicle k according to the first stage solution. Let $\underline{V_k} \subset V_k$ and $\bar{V}_k \subset V_k$ denote the subset of customers which are visited before and after the recharging station. Let c_i be the cost of serving customer i with a separate EV. The randomness of the waiting times at the recharging stations is modeled using a set of scenarios. A scenario represents the joint realization of the waiting times at all visited recharging stations. For the second stage problem, $\xi^s = (\omega_i^s)$ denotes the realization of the random waiting time at station i under scenario $s = 1, \dots, N$. Here we have an additional continuous decision variable to keep track of the lateness if an EV returns to the depot after its due date. l_{n+1}^{sk} denotes the lateness of vehicle k under scenario s. Table 5.2 presents the notation of the second-stage problem for vehicle k and scenario s.

Table 5.2. Notation for the second-stage problem

Sets:

V_k	Set of customers on route k
$V_{k,0}$	Set of departure depot and customers on route k

Sets:

i_k^*	Index of the recharging station on route k
$\underline{V}_k$	Set of customers visited before recharging at i_k^*
$\underline{V}_{k,0}$	Set of departure depot and customers visited before recharging at i_k^*
$\underline{V'}_k$	Set of recharging station i_k^* and customers visited before recharging
$\bar{V}_k$	Set of customers visited after recharging at i_k^*
$\bar{V}'_k$	Set of recharging station i_k^* and customers visited after recharging at i_k^*
$\bar{V}_{k,n+1}$	Set of customers visited after recharging at i_k^* and arrival depot
$V'_{k,n+1}$	Set of recharging station i_k^*, all customers and arrival depot

Parameters:

$\omega_{i_k^*}^{sk}$	Realized waiting time at recharging station i_k^*
c_i	Cost of serving customer i with a separate EV
c_o	Overtime wage per unit time

Decision variables:

$\bar{\tau}_i^{sk}$	Service start time at vertex i
$\bar{u}_i^{sk}$	Remaining cargo level at vertex i
$\bar{y}_i^{sk}$	Battery SoC upon arrival at vertex i
$\bar{Y}_i^{sk}$	Battery SoC at the departure from station i
$\bar{x}_{ij}^{sk}$	1 if the vehicle traverses arc (i,j); 0 otherwise
l_{n+1}^{sk}	Lateness of the depot

The mathematical model of the second-stage problem for EV k under scenario $s = 1, \dots, N$ is given below.

$$Q^k(x, \xi^s) =$$

$$\begin{aligned} minimize \quad & c_d \sum_{k \in K} \left(\bar{\tau}_{n+1}^{sk} - \tau_{i_k^*}^{k}\right) + c_e \sum_{k \in K} \sum_{i \in V_{k,0}} \sum_{j \in V'_{k,n+1}} \left(\bar{x}_{ij}^{sk} - x_{ij}^{k}\right) \\ & + c_i \sum_{k \in K} \sum_{i \in \bar{V}'_k} \sum_{j \in \bar{V}_{k,n+1}} \left(1 - \bar{x}_{ij}^{sk}\right) + (c_o - c_d) \sum_{k \in K} l_{n+1}^{sk} \end{aligned} \tag{5.15}$$

$$subject\ to$$

$$\sum_{i \in \bar{V}'_k} \bar{x}_{ij}^{sk} \leq 1 \qquad j \in \bar{V}_k \tag{5.16}$$

$$\sum_{i \in \bar{V}'_k} \bar{x}_{i,n+1}^{sk} = 1 \tag{5.17}$$

$$\sum_{i \in \bar{V}'_k} \bar{x}_{ij}^{sk} = \sum_{i \in \bar{V}_{k,n+1}} \bar{x}_{ji}^{sk} \qquad j \in \bar{V}_k \tag{5.18}$$

$$\bar{x}_{ij}^{sk} = x_{ij}^{k} \qquad i \in \underline{V}_{k,0}, j \in \underline{V'}_k \tag{5.19}$$

$$\bar{\tau}_i^{sk} + \sum_{k \in K} (t_{ij} + s_i)\, \bar{x}_{ij}^{sk} - l_0 \left(1 - \sum_{k \in K} \bar{x}_{ij}^{sk} \right) \leq \bar{\tau}_j^{sk} \qquad i \in \bar{V}_k, j \in \bar{V}_{k,n+1} \qquad (5.20)$$

$$\tau_{i_k^*}^{sk} + t_{i_k^* j} \bar{x}_{i_k^* j}^{sk} + g\left(\bar{Y}_{i_k^*}^{sk} - \bar{y}_{i_k^*}^{sk} \right) + \omega_{i_k^*}^{sk} - M(1 - \bar{x}_{i_k^* j}^{sk}) \leq \bar{\tau}_j^{sk} \qquad j \in \bar{V}_{k,n+1} \qquad (5.21)$$

$$e_j \leq \bar{\tau}_j^{sk} \leq l_j \qquad j \in \bar{V}_k \qquad (5.22)$$

$$\bar{\tau}_{n+1}^{sk} - l_0 \leq l_{n+1}^{sk} \qquad (5.23)$$

$$0 \leq \bar{y}_j^{sk} \leq \bar{Y}_{i_k^*}^{sk} - (h \cdot d_{i_k^* j}) \bar{x}_{i_k^* j}^{sk} + Q(1 - \bar{x}_{i_k^* j}^{sk}) \qquad j \in \bar{V}_{k,n+1} \qquad (5.24)$$

$$0 \leq \bar{y}_j^{sk} \leq \bar{y}_i^{sk} - (h \cdot d_{ij}) \bar{x}_{ij}^{sk} + Q(1 - \bar{x}_{ij}^{sk}) \qquad i \in \bar{V}_k, j \in \bar{V}_{k,n+1} \qquad (5.25)$$

$$\bar{x}_{ij}^{sk} \in \{0,1\} \qquad i \in, j \in \bar{V}_{n+1}^{k'} \qquad (5.26)$$

$$\bar{\tau}_i^{sk}, \bar{y}_i^{sk}, \bar{Y}_i^{sk}, l_{n+1}^{sk} \geq 0 \qquad i \in \underline{V'}_k, j \in \underline{V}_{k,n+1} \qquad (5.27)$$

The objective function of the second stage problem minimizes the total cost of drivers, energy, skipped customers and late arrivals at the depot. Note that, the driver wages and energy costs of the first stage problem are subtracted to prevent double counting. Constraints (5.16)−(5.18) are connectivity and flow conservation constraints. Since the sequence of the customers visited prior to recharging does not change in the second stage, the values of the flow variables related to these customers are fixed by constraints (5.19). Constraints (5.20) and (5.21) keep track of the time when EV k departs from a customer/depot and from station i_k^*. $\omega_{i_k^*}^{k}$ is now the realized waiting time when EV k arrives at station i_k^*. Constraints (5.22) set the customer service time windows. Constraints (5.23) determine the lateness when the vehicle arrives at the depot after its due date. Constraints (5.24) and (5.25) keep track of the SoC and determine the energy transferred at the station if any customer is removed from the route. Finally, (5.26) and (5.27) define the domain of the decision variables.

5.3.4. Modeling the Waiting Times

In this study, we assume an $\mathrm{M/G/1}$ queueing system at recharging stations. The arrivals of the EVs at stations follow a Poisson distribution with mean λ and the service time may be drawn from any distribution with known mean and standard deviation. Here, service rate is the recharging rate. In the first stage model, we use expected waiting times which are calculated using the $\mathrm{M/G/1}$ queueing system equations. The random variable ω_i is replaced with its expected value $\mathbb{E}[\omega_i]$. However, in the second stage, we generate random waiting times following the properties of their distribution.

5.4. Solution Methodology

We propose an ALNS approach to solve the problem. In the first stage, an initial solution is constructed first using a greedy algorithm and improved using the destroy and repair mechanisms. In the second stage, we utilize another procedure along with stochastic simulation to determine the final solution. The stochastic simulation is used to calculate probabilities in some operators and to calculate the expected cost of the second stage solution.

5.4.1. Stochastic Simulation for Computing the Expected Values and Probabilities

In this study, the waiting times at the recharging stations are random variables. Second stage problem relies on the realized values of these random variables and the first stage problem uses the expected value of the objective function of the second stage problem. In addition, in some operators of the ALNS, some probability measures have to be calculated. We use simulation to calculate these probabilities and the expected costs. This technique performs sampling in the probability space and estimates the probabilities based on the law of large numbers. Li et al. (2010), Shi et al. (2018) and Gutiemez et al. (2018) also used this technique in their studies. The procedures will be explained in sections 5.4.2 and 5.4.5.

5.4.2. Destroy Operators

We use two types of destroy operators to remove customers and recharging stations from a solution.

5.4.2.1. Customer Removal (CR) Operators

We use random, worst-distance, worst-time, Shaw, proximity-based, demand-based, time-based, zone, random route, and greedy route removals, which are commonly used in the literature. In addition, we implement probabilistic worst removal, which is introduced in Chen et al. (2014) with some problem specific adaptations.

Probabilistic worst removal: This operator aims to remove the customers that potentially cause delay for their successor customers. For each customer, the probability that the time window of its successor customer will be violated is calculated. Since the random waiting times are revealed at the recharging stations, this probability is zero for the customers that are visited before the recharging station. After calculating the probabilities, γ customers having the highest probabilities are removed from their routes. Procedure 5.1 provides the details of the simulation procedure to calculate this probability for customer i, i.e., $P(i)$.

After the customer removal operation, an idle station check operation is performed for each changed route. If there is a station in the route, it may become idle since the route is now shorter due to the absence of some customers. In this case, these stations are also removed from their routes.

Procedure 5.1 Simulation to calculate probability that the next customer is infeasible

Set $iter \leftarrow 1, p \leftarrow 0$ and N to be a sufficiently large number
Determine the index of the station visited, i.e., s
while $iter \leqslant N$ **do**
  Generate a waiting time, ω_s from sample space according to the probability distribution of station s.
  Using ω_s update the arrival times of the customers from s to the successor of i, i.e., $i+1$.
  if $\tau_{i+1} > l_{i+1}$ **then**
    $p = p + 1$
  end if
  $iter = iter + 1$
end while
$P(customer\ i{+}1\ is\ infeasible) = p/N$
Return $P(i) = P(customer\ i{+}1\ is\ infeasible)$

5.4.2.2. Station Removal (SR) Operators

We use random station removal which is introduced in Chapter 2 as well as the following new operator that we propose:

Longest Waiting Time Removal: Since the stations have different waiting time distributions, some have longer expected waiting times. Hence, removing such stations would help decrease the violations and the objective function value. This operator selects σ stations having the longest waiting times and removes them from the solution.

5.4.3. Repair Operators

Similar to the destroy operators, we use two types of repair operators specific to customers and recharging stations.

5.4.3.1. Customer Insertion (CI) Operators

Customer insertion is applied after the customer removal to reinsert the removed customers to the partial solution. We adapt the greedy insertion of Chapter 2 and also use two probabilistic insertion operators inspired from Chen et al. (2014).

Deterministic Greedy Insertion: This operator does not consider the stochastic part of the problem. It does not include the recourse cost in the evaluation of an insertion. For all (i, j) pairs in the partial solution, the removed customer is inserted between i and j and the increase in the deterministic part of the objective function is calculated. Specifically, the recourse cost of the first stage solution is not calculated, i.e., $Q(x, \xi(\omega))$ term in the objective function is excluded. After determining the costs of all feasible insertions, the customer that leads to the least cost increase is inserted to its least costly position. After each insertion, the insertion costs of other removed customers to that modified route are updated and this procedure is repeated until all removed customers are reinserted to the solution.

Probabilistic Greedy Insertion: In this insertion, the expected cost of the second stage problem is also considered in the evaluation of an insertion. For each customer removed, all possible positions in the fleet are determined and the first stage costs as well as the expected recourse costs are calculated. The calculation of the expected recourse cost will be explained in Section 5.4.5. Then, the insertion cost becomes the difference between the first stage costs before and after the insertion plus the expected recourse cost. The customer with the lowest cost is selected and it is inserted to its lowest cost position.

Probabilistic Greedy Insertion with Confidence: This operator is an extension to the deterministic greedy insertion. For each customer and its feasible insertion positions, the insertion cost is determined by calculating the first stage objective function value. Moreover, the probability that the route is still feasible after the insertion of the removed customer into that position is also calculated as described in Procedure 5.2. Then, the positions that have a probability lower than the confidence level δ are discarded. Among other alternatives, the customer having the least cost is selected and the least cost insertion is performed.

Sometimes, customers may need recharging stations to be inserted along with them since the battery is not sufficient to cover this extension. In this case, a station is inserted by Greedy Station Insertion mechanism which is proposed in Chapter 2.

Procedure 5.2 Simulation to calculate probability that the route is feasible

Set $iter \leftarrow 1, p \leftarrow 0$ and N to be a sufficiently large number
Determine the index of the station visited on route k, i.e., s
while $iter \leqslant N$ **do**
  Generate a waiting time, ω_s from sample space according to the probability distribution of station s.
  Using ω_s update the arrival times of the vertices visited after s.
  for all vertex i, visited after the station **do**
    if $\tau_i \geqslant l_i$ **then**
      $p = p + 1$
      break
    end if
  end for
  $iter = iter + 1$
end while
$P(route\ k\ is\ feasible) = (1 - p)/N$
Return $P(k) = P(route\ k\ is\ feasible)$

5.4.3.2. Station Insertion (SI) Operators

After the station removal, the solution may become infeasible with respect to battery SoC. Then, a station insertion method is applied to repair the solution. We use *Deterministic Best Station Insertion* and *Probabilistic Best Station Insertion.* The former is proposed in Chapter 2 and here the insertion is performed considering only the first stage objective function values. Hence, insertion cost of a station to a position is the difference of the first stage costs of the solutions before and after the insertion. The station which has the least insertion cost is inserted to its best position. On the other hand, Probabilistic Best Station Insertion considers the recourse costs, as well. For each feasible insertion, total expected cost is also calculated and the insertion cost becomes the difference between the first stage costs plus the expected recourse cost. The station which increases the total cost least is then inserted to its best position. This new mechanism is outlined in Procedure 5.3.

Procedure 5.3 Simulation to calculate expected recourse cost and construct second stage solution

Set $iter \leftarrow 1, C_k \leftarrow 0$ and N to be a sufficiently large number
Determine the index of the station visited on route k, i.e., s
Determine the arrival time at the depot, i.e., τ_{n+1}
while $iter \leqslant N$ **do**
  Generate a waiting time, ω_s from sample space according to the probability distribution of station s.
  Using ω_s update the arrival times of the vertices visited after the station.
  for all customer i, visited after the station **do**
    if $\tau_i > l_i$ **then**
      Remove customer i from the route
      Update the arrival times of the vertices after customer i
      Determine the energy saved after removal of i, i.e., Δ_{y_i}
      $C_k = C_k + c_i - c_e \Delta_{y_i}$
    end if
  end for
  $iter = iter + 1$
  if $\tau_{depot} > l_0$ **then**
    $C_k = C_k + c_d(l_0 - \tau_{n+1}) + c_o(\tau_{depot} - l_0)$
  else
    $C_k = C_k + c_d(\tau_{depot} - \tau_{n+1})$
  end if
end while
$E[C_k] = C_k/N$
Return $E[C_k]$ and *the resulting route*

5.4.4. Initial Solution

The initial solution is constructed by putting all customers in a list randomly and applying the Deterministic Greedy Insertion method.

5.4.5. Solving the Second Stage Problem

Given a first stage solution, ALNS is utilized to solve also the second stage problem. The second stage problem relies on the realized values of the random waiting times and the first stage problem uses the expected objective function value of the second stage problem. Procedure 5.3 provides the steps for calculating the expected recourse cost and constructing the second stage solution.

If the first stage solution is infeasible, i.e., one of the customers cannot be visited within its time window, then the customers visited after the recharging are checked in an iterative way starting from the first customer visited following the station. If the EV arrives at customer i after its late service time, then this customer is skipped with penalty c_i. After removing this customer from the route, the arrival times of the subsequent vertices are updated, and the same procedure is applied until all customers become feasible. This simulation is performed N times with different waiting times belonging to different scenarios. Finally, the expected cost is calculated as the average of all the simulated costs.

5.4.6. Waiting Time Adjustment

The algorithm finds the first stage solutions assuming fixed waiting times at the recharging stations. If the fixed value is much less than the average waiting time, then the first stage cost will be low, but the recourse cost will be high since many customers will be skipped due to long realized waiting times. This is a risk seeking strategy and it may yield low quality solutions if the realized waiting times are long. On the other hand, a risk averse strategy may be followed by assuming much longer waiting times than the averages. This will increase the cost of the first stage decisions because the fleet size may grow to cover all customers in the presence of long queue times. However, the recourse cost will be low. So, if the fixed waiting time at a station is increased, the first stage cost will be higher, and the recourse cost will be lower. On the contrary, if it is decreased, the first stage cost will be lower, and the recourse cost will be higher. So, based on this tradeoff we propose an adaptive mechanism to adjust these fixed waiting times after every N_w iterations. Let $\mathbb{E}[Q]$ and $\mathbb{E}[\hat{Q}]$ denote the total expected costs of the best-found solutions in the last N_w iterations and previous N_w iterations, respectively. If $\mathbb{E}[Q] > \mathbb{E}[\hat{Q}]$ then the quality of the deterministic solution deteriorates in terms of estimating the stochastic solution because the total cost has increased. In other words, the fixed waiting times underestimate the real values and may be increased in a more risk averse setting. In contrast, if $\mathbb{E}[Q] < \mathbb{E}[\hat{Q}]$, then we conclude that the fixed times overestimate the real values and may be decreased. In this way, we tune the waiting times in an attempt to reduce the total cost. The tuning is performed by multiplying the average waiting times with a constant α calculated as follows:

$$\alpha = \frac{\mathbb{E}[Q]}{\mathbb{E}[\hat{Q}]} \tag{5.28}$$

If α is less than 1, it means that the waiting times will be decreased for the next N_w iterations. Conversely, if it is greater than 1, they will be increased for the next N_w iterations. In the latter case, the current solution may become infeasible when the routes are updated using new waiting times. Since the waiting times are increased, some customers which are visited after the station may not be covered within their time windows. In this case, the Solution Correction procedure is applied to make the current solution feasible.

Solution Correction: Similar to the procedure when obtaining the final solution after random waiting times are revealed, this mechanism detects the customers whose time windows are violated. These customers are then removed from their routes and put in a list. Sometimes, the EV may be late for the depot, as well. In these cases, the customer visited before the depot is also removed even if the EV arrives at that customer on time. If the EV is still late, then the removal continues with the next customers until the route becomes feasible. Then the Deterministic Greedy Insertion is applied for the removed customers to include them in the solution.

The general structure of the proposed metaheuristic is outlined in Procedure 5.4. $f(x)$ represents the objective function of the first stage solution x.

5.5. Computational Study

We test the performance of the proposed ALNS using 100-customer EVRPTW-SP instances of Desaulniers et al. (2016). This study also assumed that each vehicle may visit at most one recharging station during its journey and its data set is compatible with our case. As highlighted in Desaulniers et al. (2016) and Chapter 4, wide time windows do not have much effect on recharging decisions since they can be easily satisfied. So, we focus on type-1 instances where the customers have narrow time windows in order to better observe the influence of the waiting times at recharging stations. The data set includes three different configurations, namely, C, R and RC. In type C problems, customers are geographically distributed within clusters whereas in type R problems, they are located randomly. In type RC problems, they are both clustered and randomly distributed. The metaheuristic is coded in Java and all experiments are conducted on an Intel Core i7-8700 CPU 3.2 GHz processor with 16 GB of RAM.

Procedure 5.4 General Structure of the Proposed Metaheuristic

Generate an initial solution x_0, $x_{best} \leftarrow x_{current} \leftarrow x_{previous} \leftarrow x_0$
Initialize the scores and probabilities of the operators, $iter \leftarrow 1$
Generate random utilization levels ρ_i for each station $i \in F$ using the uniform distribution $U[\underline{\rho} - \bar{\rho}]$
Calculate the expected waiting times at the stations and set $\omega_i \leftarrow \mathbb{E}[\omega_i]$ for all $i \in F$
while $iter <$ Maximum number of iterations **do**
  if $iter$ is a multiple of Δ **then**
    Select a Station Removal Operator and remove γ_s stations from $x_{current}$
    Select a Station Insertion Operator and repair the solution
  else
    Select a Customer Removal operator and remove γ_c customers from $x_{current}$
    Select a Customer Insertion operator and repair the solution
  end if
  Calculate the expected recourse cost, $\mathbb{E}[Q]$, using stochastic simulation
  $f(x_{current}) \leftarrow f(x_{current}) + \mathbb{E}[Q]$
  if $f(x_{current}) < f_{previous}$ **then**
    $x_{previous} \leftarrow x_{current}, f_{previous} \leftarrow f(x_{current})$
    if $f(x_{current}) < f_{best}$ **then**
      $x_{best} \leftarrow x_{current}, f_{best} \leftarrow f(x_{current})$
    end if
  else
    Accept the solution using Simulated Annealing Criterion
  end if
  if $iter$ is a multiple of N_c **then**
    Update adaptive weights of CR and CI operators and calculate new selection probabilities
  end if
  if $iter$ is a multiple of N_s **then**
    Update adaptive weights of SR and SI operators and calculate new selection probabilities
  end if
  if $iter$ is a multiple of N_w **then**
    Calculate the change factor α for the waiting times and set $\omega_i \leftarrow \omega_i \times \alpha, \ \forall i \in F$
    if $\alpha > 1$ **then**
      Apply *Solution Correction*
    end if
  end if
  $iter \leftarrow iter + 1$
end while
Return x_{best}

We also perform an analysis on the maximum number of iterations by recording the iteration number where the best solution is obtained. In many cases, the best solution is found close to 25,000 iterations. Hence, the algorithm is terminated after 25,000 iterations.

5.5.1. Problem Settings

We assume that the recharging (service) times follow exponential distribution with parameter μ. The arrivals of EVs at stations follow a Poisson distribution with mean λ. Hence, the queueing systems at the recharging stations become $M/M/1$ systems. We assume that the batteries are operated between $10\% - 90\%$ of their capacities to improve the battery life

(Pelletier et al., 2017). Hence, if the recharge amount of different EVs is assumed to be distributed uniformly, then the average recharge amount will be 40% of the capacity and average recharging time will be the time required to increase the SoC by 40%. So, $\mathbb{E}[\text{Service time}] = 0.4 \times g \times Q$ and $\mu = 1/\mathbb{E}[\text{Service time}]$. In $M/M/1$ systems, the waiting time in the queue is 0 with probability $1 - \rho$ whereas with probability ρ it is an exponential random variable with parameter $\mu(1 - \rho)$, where $\rho = \lambda/\mu$ is the utilization rate of the charger. Hence, upon arrival at a station, the expected waiting time in the queue can be calculated by the formula $\lambda/\mu(\mu - \lambda)$. In the first stage model, we use the expected waiting times calculated with this formula. However, in the second stage, we generate random waiting times following the properties of their distribution.

We assume that all stations have identical chargers, i.e., their service rates are the same. However, the arrival rate λ may vary from one station to another depending on different factors such as location, traffic volume, and availability of the stations. Hence, we assign different arrival rates to each station. We choose two utilization levels and use them as bounds of a uniform distribution, i.e. $[\underline{\rho}, \overline{\rho}]$. Then, we generate a utilization level for each station using this distribution. In this way, a unique arrival rate is assigned to each station. In this study, we use $[30\%, 70\%]$ as the utilization bounds.

The probabilities and expected total costs are calculated using simulation which generates N replications. Hence, the precision of the values depends on this parameter. If it is large, then the precision would be better, but it leads to higher computational effort. We set $N = 1000$ as in Li et al. (2010).

5.5.2. Parameter Tuning

We used the same parameter values as in Chapter 2 for the ALNS. However, the number of iterations after which the average waiting times are adjusted, N_w, as well as the threshold confidence level, δ, which is used in Probabilistic Greedy Insertion with Confidence, are new parameters. Hence, we tuned these parameters using the following subset of instances: C101, C104, R102, R105, RC103, RC107. We apply a similar methodology as in Ropke and Pisinger (2006a, b). For each parameter, we determine 10 candidate values and we perform 10 runs for each value and each instance. We calculate the average objective function values of 10 runs as well as their deviations from the minimum value attained in the computations. Then, the average of the average deviations is calculated for each parameter value and the one with the minimum deviation is selected. We first determine the value of δ, next N_w is tuned using the selected value of δ. Table 5.3 shows the average deviations for each parameter value. The best performing values are indicated in bold. Based on these initial experiments, we set $\delta = 80\%$ and $N_w = 300$.

Table 5.3. Parameter tuning

Parameter	Values Tested										
δ	Value	10%	20%	30%	40%	50%	60%	70%	**80%**	90%	95%
	Deviation	871.3	902.0	1013.7	1042.1	1069.7	896.1	993.7	**866.1**	922.0	1046.8
N_w	Value	100	200	**300**	400	500	600	700	800	900	1000
	Deviation	1056.7	1107.3	**1000.3**	1136.4	1133.7	1067.6	1107.0	1182.0	1222.9	1242.0

5.5.3. Results

For this problem, there are no benchmark results. Hence, we may compare our results with the costs obtained by first solving the deterministic problem with expected waiting times, and then computing the expected cost of recourse associated with the best-found solutions. To obtain the deterministic solutions we run the ALNS algorithm without the stochastic components. We use mean waiting time values and the search is based on the first stage objective function value. Hence, we do not consider the recourse costs. Then we compute the expected recourse cost using the best-found solutions. The comparison of these solutions with those obtained by the proposed heuristic is given in Table 5.4. *Det(X)* and *Q(X)* stand for the deterministic cost of the first stage solution and the expected recourse cost. $f_d(X)$ and $f_s(X)$ are the total costs of the deterministic and stochastic approaches. *Imp*(%) shows the percent improvement achieved by the stochastic approach compared to the deterministic approach. It is calculated as $(f_d(X) - f_s(X))/f_d(X)$. The best results as well as the average computational times of 10 runs are presented.

The results show that, in all types of instances, the solutions of the stochastic approach are better than those of the deterministic approach. This improvement comes from the decreased recourse costs in the stochastic case. Although the deterministic cost increases in all instances, since the recourse costs are much smaller, total cost decreases. The average improvement achieved is 13%. On the other hand, computational times are much shorter in the deterministic case. This is due to the lack of simulation of random variables both in the ALNS operators and in the objective function evaluation.

5.5.3.1. Sensitivity of Results to Utilization Levels at the Stations

We also perform experiments with relatively higher and lower utilization levels to investigate the sensitivity of the solutions to the average waiting times. $[\underline{\rho}, \overline{\rho}]$ is set equal to $[10\%, 50\%]$ and $[50\%, 85\%]$ for the low and high utilization settings, respectively. The results are provided in Table 5.5 and Table 5.6. In the low utilization setting, the improvement obtained by the stochastic methodology decreases to 8% on average. Since the chargers are idle more often in this setting compared to the moderate setting, the realized waiting times are lower. So, infeasible solutions are observed more rarely. On the other hand, when the utilization levels are higher, the stations are more occupied and finding solutions considering the recourse costs becomes more crucial in order to reduce high penalties incurred due to the skipped customers. The improvement over the deterministic method is 16%.

Table 5.4. Comparison of best deterministic and stochastic solutions

Instance	Deterministic				Stochastic				*Imp* (%)
	Det(X)	*Q(X)*	*f_d(X)*	Time(sec)	*Det(X)*	*Q(X)*	*fs(X)*	Time(sec)	
C101	29,197	8,879	38,076	36.9	30,605	3,456	34,061	338.2	11%
C102	28,349	8,367	36,716	78.7	29,710	2,064	31,774	517.9	13%
C103	27,299	5,089	32,388	120.3	28,012	2,917	30,930	1,063.7	5%
C104	25,821	6,620	32,441	164.2	25,664	2,296	27,960	942.5	14%
C105	27,881	7,804	35,686	48.4	29,916	2,263	32,179	381.3	10%
C106	28,877	7,213	36,090	60.5	29,533	2,203	31,736	413.6	12%
C107	27,352	7,039	34,392	62.1	29,633	1,727	31,360	479.9	9%
C108	27,227	6,417	33,644	77.6	29,197	1,139	30,337	511.6	10%
C109	26,752	7,552	34,303	108.6	27,397	1,613	29,010	611.6	15%
C - Average	**27,639**	**7,220**	**34,859**	**84**	**28,852**	**2,187**	**31,039**	**584**	**11%**
R101	30,037	8,283	38,319	45.9	31,648	2,516	34,164	499.3	11%
R102	25,679	7,571	33,250	72.5	25,863	4,278	30,141	662.2	9%
R103	21,515	8,277	29,792	94.7	22,996	3,103	26,099	702.9	12%
R104	17,402	7,757	25,159	119.4	18,749	3,271	22,021	844.5	12%
R105	24,319	8,137	32,456	55.0	25,953	3,280	29,233	645.7	10%
R106	21,506	8,349	29,855	82.3	22,965	3,424	26,388	663.8	12%
R107	18,785	8,533	27,318	105.2	21,676	2,173	23,848	768.4	13%
R108	17,281	7,132	24,413	122.2	18,561	3,521	22,082	834.4	10%
R109	21,501	8,414	29,915	78.4	21,734	4,304	26,038	671.6	13%
R110	18,703	5,683	24,386	106.0	20,011	2,558	22,569	762.2	7%
R111	18,772	6,928	25,700	102.8	20,053	3,071	23,124	783.0	10%
R112	17,423	6,833	24,255	132.8	18,675	3,759	22,434	858.7	8%
R - Average	**21,077**	**7,658**	**28,735**	**93**	**22,407**	**3,271**	**25,678**	**725**	**11%**
RC101	26,021	8,075	34,096	50.5	27,579	2,368	29,947	508.2	12%
RC102	24,472	6,912	31,384	73.5	24,697	2,261	26,957	507.4	14%
RC103	20,299	7,494	27,793	87.1	21,806	1,485	23,290	578.7	16%
RC104	17,680	7,223	24,904	101.4	18,875	836	19,711	635.3	21%
RC105	23,158	11,166	34,324	66.4	24,744	2,023	26,767	587.9	22%
RC106	21,571	9,012	30,583	66.3	23,253	1,035	24,288	629.7	21%
RC107	18,830	8,394	27,225	87.3	20,383	1,071	21,454	600.9	21%
RC108	18,852	7,748	26,600	98.2	18,911	2,097	21,008	628.8	21%
RC - Average	**21,360**	**8,253**	**29,613**	**79**	**22,531**	**1,647**	**24,178**	**585**	**19%**
Average All	**23,359**	**7,710**	**31,069**	**85**	**24,597**	**2,368**	**26,965**	**631**	**13%**

5.5.3.2. Effect of Parameter N_w

We propose a new adaptive mechanism which adjusts the fixed waiting times used in the ALNS after each N_w iterations. To investigate the contribution of this mechanism to the solution quality, we perform experiments keeping expected waiting times fixed throughout the search. Table 5.7 gives the results obtained in both cases. The values in column "*Diff. (%)*" show the percentage difference between the total costs. Negative values indicate that the cost obtained in the "without adjustment" setting is better than that obtained in the "with adjustment" setting. The average improvement obtained by adjusting the waiting time is 1%. Although the adjustment deteriorates the solution in few instances, its contribution can be up to 4.1%. Note that all RC-type problems benefited from this mechanism.

Table 5.5. Results for low utilization levels at stations

Instance	Deterministic				Stochastic				*Imp* (%)
	Det(X)	*Q(X)*	$f_d(X)$	Time(sec)	*Det(X)*	*Q(X)*	*fs(X)*	Time(sec)	
C101	28,714	5,914	34,628	40	29,236	2,061	31,296	320	10%
C102	26,971	7,242	34,212	81	27,176	2,348	29,524	438	14%
C103	25,798	6,362	32,160	128	27,309	1,982	29,292	776	9%
C104	24,967	3,583	28,550	165	25,189	904	26,094	827	9%
C105	25,593	5,174	30,768	48	27,200	2,411	29,610	393	4%
C106	25,428	5,757	31,185	64	27,044	2,336	29,381	408	6%
C107	25,327	4,444	29,770	65	26,529	2,341	28,870	425	3%
C108	25,050	5,594	30,644	82	26,789	1,486	28,275	489	8%
C109	24,719	4,044	28,763	112	24,772	2,323	27,095	610	6%
C - Average	**25,841**	**5,346**	**31,187**	**87**	**26,805**	**2,021**	**28,826**	**521**	**7%**
R101	28,646	5,418	34,064	48	28,798	2,480	31,278	415	8%
R102	24,277	5,344	29,622	75	25,785	1,671	27,456	512	7%
R103	20,098	5,197	25,294	98	21,493	2,316	23,808	636	6%
R104	17,254	3,875	21,128	129	17,371	1,866	19,237	728	9%
R105	22,950	4,304	27,254	58	23,119	2,739	25,858	436	5%
R106	20,094	5,338	25,432	81	21,524	2,076	23,600	499	7%
R107	18,627	3,648	22,275	104	18,728	2,436	21,165	616	5%
R108	15,921	4,017	19,938	121	17,303	1,570	18,873	745	5%
R109	20,128	4,376	24,504	77	21,536	1,348	22,885	510	7%
R110	17,302	3,910	21,212	107	18,658	1,447	20,105	625	5%
R111	17,437	4,404	21,841	104	18,735	1,524	20,259	610	7%
R112	17,240	3,139	20,380	130	17,284	2,036	19,321	799	5%
R - Average	**19,998**	**4,414**	**24,412**	**94**	**20,861**	**1,959**	**22,820**	**594**	**6%**
RC101	24,593	6,607	31,200	51	26,082	1,312	27,394	372	12%
RC102	23,023	5,249	28,272	74	23,209	1,476	24,685	421	13%
RC103	18,918	4,091	23,009	90	20,248	1,257	21,505	466	7%
RC104	17,469	3,991	21,460	102	18,812	536	19,348	594	10%
RC105	21,701	5,446	27,146	67	23,227	1,300	24,527	450	10%
RC106	20,228	5,171	25,399	66	21,778	851	22,629	438	11%
RC107	17,473	3,276	20,748	86	18,910	817	19,727	525	5%
RC108	17,464	5,272	22,736	97	17,576	990	18,565	606	18%
RC - Average	**20,108**	**4,888**	**24,996**	**79**	**21,230**	**1,067**	**22,297**	**484**	**11%**
Average All	**21,982**	**4,883**	**26,865**	**87**	**22,965**	**1,683**	**24,648**	**533**	**8%**

Table 5.6. Results for high utilization levels at stations

Instance	Deterministic				Stochastic				*Imp* (%)
	Det(X)	*Q(X)*	$f_d(X)$	Time(sec)	*Det(X)*	*Q(X)*	*fs(X)*	Time(sec)	
C101	34,337	9,347	43,685	39	34,615	4,138	38,753	375	11%
C102	32,109	11,577	43,685	76	32,533	2,657	35,191	642	19%
C103	30,437	9,658	40,096	120	31,625	3,923	35,547	1,312	11%
C104	29,285	8,900	38,185	169	28,172	2,879	31,050	1,016	19%
C105	32,349	7,710	40,059	51	32,407	2,904	35,311	557	12%
C106	31,651	10,486	42,137	59	30,563	2,823	33,385	458	21%
C107	31,164	9,595	40,759	63	30,147	3,807	33,954	452	17%
C108	30,097	9,573	39,670	81	29,809	3,372	33,181	495	16%
C109	28,991	7,901	36,892	106	29,437	2,691	32,129	598	13%
C - Average	**31,158**	**9,416**	**40,574**	**85**	**31,034**	**3,244**	**34,278**	**656**	**15%**
R101	33,086	10,903	43,989	50	35,953	5,490	41,443	645	6%
R102	28,770	10,015	38,785	72	31,582	3,916	35,498	747	8%
R103	23,127	8,987	32,115	96	24,534	4,251	28,785	772	10%
R104	20,124	8,184	28,308	129	21,613	1,762	23,374	946	17%
R105	27,277	11,448	38,726	60	28,853	4,160	33,012	714	15%
R106	24,344	9,213	33,556	86	24,539	5,774	30,313	738	10%
R107	21,610	9,297	30,908	105	22,986	4,878	27,865	974	10%
R108	20,019	10,358	30,377	124	20,036	4,861	24,897	956	18%
R109	23,154	12,803	35,956	76	25,908	5,786	31,694	858	12%
R110	20,240	7,935	28,175	109	21,475	3,400	24,876	915	12%
R111	21,564	10,646	32,210	106	24,257	1,741	25,997	1,011	19%
R112	20,186	10,764	30,949	136	21,388	4,092	25,479	1,023	18%
R - Average	**23,625**	**10,046**	**33,671**	**96**	**25,260**	**4,176**	**29,436**	**858**	**13%**
RC101	29,068	11,095	40,163	53	30,728	3,411	34,139	600	15%
RC102	26,159	8,471	34,630	72	26,276	3,328	29,604	611	15%
RC103	22,001	11,437	33,438	86	23,302	2,943	26,245	650	22%
RC104	20,399	10,361	30,760	105	21,792	1,053	22,845	698	26%
RC105	26,071	12,487	38,558	68	27,676	3,009	30,685	764	20%
RC106	23,333	10,511	33,843	66	23,430	3,182	26,611	645	21%
RC107	21,683	8,022	29,704	89	21,875	2,116	23,991	688	19%
RC108	21,745	11,492	33,237	96	23,194	904	24,098	749	27%
RC - Average	**23,807**	**10,484**	**34,292**	**79**	**24,784**	**2,493**	**27,277**	**675**	**21%**
Average All	**26,197**	**9,982**	**36,179**	**87**	**27,026**	**3,304**	**30,331**	**730**	**16%**

Table 5.7. The results with and without adjusting waiting times

Instance	With Adjustment				Without Adjustment				*Diff*. (%)
	Det(X)	*Q(X)*	$f_d(X)$	Time(sec)	*Det(X)*	*Q(X)*	*fs(X)*	Time(sec)	
C101	30,605	3,456	34,061	338	30,557	3,688	34,245	356	0.5%
C102	29,710	2,064	31,774	518	30,023	1,862	31,885	487	0.3%
C103	28,012	2,917	30,930	1,064	28,127	2,836	30,963	790	0.1%
C104	25,664	2,296	27,960	942	26,069	1,817	27,886	1,036	-0.3%
C105	29,916	2,263	32,179	381	30,424	1,883	32,307	410	0.4%
C106	29,533	2,203	31,736	414	29,960	1,064	31,025	411	-2.2%
C107	29,633	1,727	31,360	480	29,894	1,582	31,476	453	0.4%
C108	29,197	1,139	30,337	512	29,149	1,933	31,082	546	2.5%
C109	27,397	1,613	29,010	612	28,807	1,178	29,985	634	3.4%
C - Average	**28,852**	**2,187**	**31,039**	**584**	**29,223**	**1,983**	**31,206**	**569**	**0.6%**
R101	31,648	2,516	34,164	499	31,661	2,507	34,167	464	0.0%
R102	25,863	4,278	30,141	662	25,940	4,517	30,458	763	1.1%
R103	22,996	3,103	26,099	703	23,045	2,803	25,848	715	-1.0%
R104	18,749	3,271	22,021	844	20,086	1,710	21,796	849	-1.0%
R105	25,953	3,280	29,233	646	25,919	3,972	29,891	613	2.2%
R106	22,965	3,424	26,388	664	22,999	3,537	26,536	688	0.6%
R107	21,676	2,173	23,848	768	21,667	2,492	24,159	808	1.3%
R108	18,561	3,521	22,082	834	18,641	2,906	21,548	975	-2.4%
R109	21,734	4,304	26,038	672	23,061	2,902	25,962	692	-0.3%
R110	20,011	2,558	22,569	762	20,064	2,264	22,328	756	-1.1%
R111	20,053	3,071	23,124	783	20,147	3,056	23,202	831	0.3%
R112	18,675	3,759	22,434	859	20,054	2,901	22,955	952	2.3%
R - Average	**22,407**	**3,271**	**25,678**	**725**	**22,774**	**2,964**	**25,738**	**759**	**0.2%**
RC101	27,579	2,368	29,947	508	27,663	3,034	30,696	508	2.5%
RC102	24,697	2,261	26,957	507	24,731	2,303	27,033	529	0.3%
RC103	21,806	1,485	23,290	579	21,844	2,304	24,147	613	3.7%
RC104	18,875	836	19,711	635	18,997	1,521	20,519	634	4.1%
RC105	24,744	2,023	26,767	588	24,744	2,398	27,141	610	1.4%
RC106	23,253	1,035	24,288	630	23,303	994	24,297	528	0.0%
RC107	20,383	1,071	21,454	601	20,284	1,984	22,268	610	3.8%
RC108	18,911	2,097	21,008	629	20,254	1,431	21,685	673	3.2%
RC - Average	**22,531**	**1,647**	**24,178**	**585**	**22,727**	**1,996**	**24,723**	**588**	**2.4%**
Average All	**24,597**	**2,368**	**26,965**	**631**	**24,908**	**2,314**	**27,222**	**639**	**1.0%**

5.6. Conclusions

In this study, we introduced the Electric Vehicle Routing Problem with Time Windows and Stochastic Waiting Times at Recharging Stations. We modeled the problem as a two-stage stochastic program with recourse. The randomness is incorporated using scenarios. We developed an ALNS method by introducing some problem-specific mechanisms. We tested the performance of the proposed method by comparing its results with those obtained by solving the deterministic problem using the proposed framework without the stochastic components and computing the expected recourse cost. We used stochastic simulation to calculate the probabilities and evaluate the objective function values. The results showed that proposed method is efficient in finding good solutions in reasonable amount of time.

Future research on this topic may address obtaining the distribution functions for each node in the route and calculating the expected values exactly. Since the waiting times at the stations are random variables, the arrival times at the subsequent customers are random, too. Because each arrival time is affected by the waiting time at the station. Hence, if their distribution functions are determined, then the expected arrival times can be calculated instead of simulating them. Moreover, we assumed that EVs may recharge at most once. This assumption may be relaxed by allowing multiple visits to stations during an EV’s journey. The stations may also have different recharge technologies, such as fast and slow chargers with different distributions. Then, the EVs have to decide which recharging stations to visit as well as the charger types according to their queue lengths. Finally, we used exponentially distributed service and arrival times. Different distributions may be used to represent the arrival and service processes at the recharging stations.

Chapter 6

Conclusions

This thesis investigates four variants of the electric vehicle routing problem with time windows. It is assumed that a homogeneous fleet of BEVs is used in all the problems.

Chapter 2 studies the most basic version where the EVs are allowed to recharge their batteries up to any level. The objective is to minimize the total distance with the minimum fleet size. To solve the problem, an Adaptive Large Neighborhood Search approach is proposed. It uses some efficient mechanisms from the literature as well as new operators which are developed specific to the problem. The proposed method improves some of the best-known solutions from the literature by 0.4% on average, and it provides new benchmark results for the EVRPTW with partial recharges.

Chapter 3 studies a special case of the problem where the stations are equipped with different types of chargers which differ by the recharging rates and unit recharging costs. The objective is minimizing total energy cost with the minimum fleet size. Two mathematical formulations are proposed to represent the charger types. To solve the large instances, a matheuristic approach is developed. It is based on ALNS and the solution of a mathematical model by a mixed integer programming solver. ALNS is used to search the solution space. After a predetermined number of iterations, the routes of the best solution found within these iterations are optimized in terms of the recharging decisions. To be used in the route optimization, another mathematical model is developed. The results show that using multiple chargers is advantageous to save from the energy cost. This is because of the presence of time windows. When recharging takes less time, more customers can be visited by the same EV. It is also pointed out that having fast chargers has little effect on the solutions of the instances with wide time windows.

Chapter 5 considers the possibility that EVs may wait in the queue at the recharging stations before being recharged due to limited number of chargers. Since some time intervals are more crowded due to the rush hours, these waiting times are assumed to be dependent to the time of the day. Then, the planning horizon is split into five time intervals, namely morning, noon, late afternoon, evening, and night and each one is assigned a waiting time. Hence, the stations have deterministic waiting times which depend on the time of the day. It is assumed that the stations have M/G/1 queueing system and the average waiting time equations of M/G/1 are used to determine the waiting times. The linear recharging time function is also relaxed, and its nonlinear behavior is approximated with a piecewise linear function. The problem is modeled as a mixed integer linear program and a matheuristic approach is proposed to solve it. The methodology is similar to one in the previous chapter. ALNS searches the solution space and a mixed integer programming solver is utilized to optimize the routes of the best solution found within a predetermined number of iterations. ALNS is also equipped with problem-specific mechanisms. The results show that the waiting times are essential for the problem and they should be considered in the planning.

Finally, Chapter 6 studies the EVRPTW in the presence of stochastic waiting times at the recharging stations. Unlike the previous case where the waiting times are known in advance, here the EVs have the information about the waiting time at a station when they arrive at that station. Hence, an a priori route may become infeasible if the planned station has longer queue than expected since the customers and the depot have time windows. Then the solution may be repaired by skipping the customers whose time windows are missed with a penalty and paying an overtime wage to the driver if the EV is late for the depot, as well. The problem is modeled as a two-stage stochastic program with recourse. The randomness of the waiting times is modeled using a set of scenarios. The objective is to minimize the total cost which includes energy cost, driver wage, vehicle operating cost and expected recourse cost which comes from correcting the infeasible solution. To solve the problem, an ALNS algorithm is developed. To evaluate the expected value of the objective function and calculate some probabilities, simulation is used. Results show that planning with expected waiting times usually gives poor results and ALNS performs well since it considers the stochasticity to guide the search.

The solution methodologies used in this thesis are based on heuristic approaches. In two chapters, the heuristics are combined with a mixed integer programming solver. Hence, future research may focus on exact solution methodologies which are rarely addressed in the EVRP literature. In addition, it is assumed in all problems that the recharging stations are already installed, and they are public stations. However, the company which owns the EV fleet may want to optimize the locations of the stations such that its transportation costs are minimized. This extension may be addressed in a future research. Furthermore, the energy consumption rate is also assumed constant but in reality, it is affected by various factors, such as ambient temperature, slope of the road, and load of the vehicle. It is also known that the EVs have regenerative braking which enables the battery to be recharged while going downhill. These energy features may be included in the planning and more realistic results may be obtained.

Appendix A. ALNS Parameters

All parameters are summarized in Table A.1 and the results of the parameter tuning procedure for solving the EVRPTW-PR are given in Table A.2. The tuning sequence is from top to bottom. All parameters were initially set to the values given in the "Initial Value" column. We first considered ten different values for σ_2 shown in the first row and performed ten runs on the six instances selected for tuning. Next, we calculated the average percentage deviation 'Dev%' of the average solution achieved with that value from the best solution found in all runs. Then, we determined 0.22% as the least deviation and fixed the value of the parameter σ_2 to 20, which had achieved this best average performance. We repeated this procedure for the remaining parameters in the order given until all parameter values had been tuned. The best parameter values (the ones yielding the minimum deviations) are indicated in bold. Note that the initial values in Table A.2 are the values we determined when we applied the parameter tuning for solving the EVRPTW. We do not give the detailed setting for that case since the tuning approach and the values considered are same.

Table A.1. Notation and description of the parameters

σ_2	score of the better solution
N_C	# of iterations between which adaptive weights of CR and CI algorithms are updated
ρ	roulette wheel parameter
σ_1	score of the best solution
σ_3	score of the worse solution
ϕ_1	first Shaw parameter
ϕ_2	second Shaw parameter
ϕ_3	third Shaw parameter
ϕ_4	fourth Shaw parameter
ε	cooling rate of SA
μ	initial temperature control parameter of SA
κ	worst removal determinism factor
η	Shaw removal determinism factor
n_Z	number of zones in zone removal
N_{SR}	# of iterations between which SR is performed
N_S	# of iterations between which adaptive weights of SR and SI algorithms are updated
m_r	route removal upper bound
N_{RR}	# of iterations between which route removal algorithms are performed
n_{RR}	# of consecutive iterations during which route removal algorithms are performed

Table A.2. Parameter tuning

Parameter		Initial Value	Values Tested								
σ_2	Value	6	0	2	4	9	12	14	16	18	**20**
	Dev%	0.30	0.27	0.25	0.28	0.27	0.28	0.25	0.25	0.28	**0.22**
N_C	Value	500	50	100	150	**200**	250	300	350	400	450
	Dev%	0.28	0.33	0.27	0.25	**0.23**	0.32	0.30	0.27	0.25	0.32
ρ	Value	0.2	0.05	0.1	0.15	**0.25**	0.3	0.35	0.4	0.45	0.5
	Dev%	0.30	0.33	0.25	0.30	**0.25**	0.27	0.32	0.27	0.25	0.30
σ_1	Value	33	**25**	30	35	40	45	50			
	Dev%	0.32	**0.25**	0.32	0.32	0.37	0.30	0.28			
σ_3	Value	21	3	6	9	12	13	15	24	27	30
	Dev%	0.37	0.42	0.40	0.45	0.47	0.42	0.45	0.43	0.47	0.48
ϕ_1	Value	5	**0.5**	1	3	7	9	11	13	15	
	Dev%	0.55	**0.55**	0.60	0.55	0.58	0.63	0.57	0.65	0.63	
ϕ_2	Value	1	0.25	3	5	7	9	11	**13**	15	
	Dev%	0.57	0.62	0.57	0.62	0.63	0.60	0.62	**0.55**	0.67	
ϕ_3	Value	13	**0.15**	1	3	5	7	9	11	15	
	Dev%	0.60	**0.53**	0.58	0.53	0.57	0.55	0.53	0.62	0.53	
ϕ_4	Value	0.25	1	2	3	4	5	6	7	8	9
	Dev%	0.57	0.62	0.58	0.62	0.68	0.62	0.63	0.58	0.67	0.62
ε	Value	0.9995	0.999	0.9991	0.9992	0.9993	**0.9994**	0.9996	0.9997	0.9998	0.9999
	Dev%	0.48	0.57	0.48	0.43	0.53	**0.38**	0.42	0.48	0.42	0.42
μ	Value	0.05	0.1	0.15	0.2	0.25	0.3	0.35	**0.4**	0.45	0.5
	Dev%	0.55	0.60	0.58	0.58	0.65	0.58	0.58	**0.55**	0.58	0.62
κ	Value	1	2	3	**4**	5	6				
	Dev%	0.57	0.63	0.58	**0.55**	0.60	0.57				
η	Value	10	2	4	6	8	**12**				
	Dev%	0.58	0.55	0.60	0.62	0.63	**0.55**				
n_Z	Value	15	5	7	9	11	13	19	21	**25**	30
	Dev%	0.60	0.60	0.65	0.55	0.58	0.63	0.60	0.57	**0.53**	0.58
N_{SR}	Value	10	20	30	40	50	**60**	70	80	90	100
	Dev%	0.75	0.78	0.82	0.80	0.80	**0.72**	0.82	0.73	0.77	0.77
N_S	Value	1000	1500	2000	2500	3000	3500	4000	4500	5000	**5500**
	Dev%	0.70	0.78	0.75	0.77	0.77	0.83	0.73	0.80	0.77	**0.70**
m_r	Value	0.4	**0.3**	0.5	0.6						
	Dev%	0.58	**0.57**	0.67	0.65						
N_{RR}	Value	6000	**2000**	2500	3000	3500	4000	4500	5000	5500	6500
	Dev%	0.85	**0.77**	0.87	0.78	0.77	0.85	0.80	0.78	0.80	0.88
n_{RR}	Value	1000	750	**1250**	1500	1750	2000	2250	2500	2750	3000
	Dev%	0.43	0.43	**0.40**	0.45	0.53	0.47	0.48	0.48	0.43	0.48

Appendix B. Outline of the Matheuristic

In Algorithm B.1, we provide the pseudocode of the proposed methodology.

Algorithm B.1: Pseudocode of the proposed matheuristic

Generate an initial solution, current solution ← initial solution
Initialize the scores and probabilities of the operators, $iter \leftarrow 1$
while *termination criterion is not met* **do**
  if $iter$ *is a multiple of* N_{SR} **then**
    Select an SR operator and remove stations
    Select an SI operator and repair solution
  else if $iter$ *is a multiple of* N_{RR} **then**
    for n_{RR} iterations **do**
      Select RRR or GRR operator and remove customers
      Select a CI operator and repair solution
      if $iter$ *is a multiple of* N_{SR} **then**
        Select an SR operator and remove stations
        Select an SI operator and repair solution
      end if
      Accept/reject the solution using Simulated Annealing criterion
      $iter \leftarrow iter + 1$
      if $iter$ *is a multiple of* Ω **then**
        Apply Route Enhancement to the best solution found in Ω iterations
      end if
    end for
  else
    Select a CR operator and remove customers
    if *the destroyed solution is infeasible* **then** Perform Greedy Station Insertion
    Select a CI operator and repair solution
  end if
  Accept/reject the solution using Simulated Annealing criterion
  $iter \leftarrow iter + 1$
  if $iter$ *is a multiple of* Ω **then**
    Apply Route Enhancement to the best solution found in Ω iterations
  end if
  if $iter$ *is a multiple of* Ω **then** Apply Route Enhancement to current best solution
  if $iter$ *is a multiple of* N_C **then** Update adaptive weights of CR and CI operators
  if $iter$ *is a multiple of* N_S **then** Update adaptive weights of SR and SI operators
end while

SR: Recharging Station Removal; CR: Customer Removal; RRR: Random Route Removal
SI: Recharging Station Insertion; CI: Customer Insertion; GRR: Greedy Route Removal

Appendix C. Comparison with Felipe et al. (2014)

The detailed results of the experiments for the problem of Felipe et al. (2014) using FORT instances are presented in Tables C.1-C.3. The results were obtained by using IA2 with Ω=200 and by implementing two configurations: (A) 25,000 iterations of ALNS for initialization and 10,000 iterations of matheuristic; (B) 2500 iterations of ALNS and 1000 iterations of matheuristic. "*S*" shows the number of stations in the problem, "TC" and "$Time$" refer to the total cost and computation time (in seconds), respectively, and "$\%\ Imp$" represents the percentage improvement achieved by our matheuristic for each configuration and calculated as $(\mathrm{FORT} - \mathrm{Matheuristic})/\mathrm{FORT}$. Note that for the instances where TC of FORT is shown with "–" Felipe et al. (2014) did not report any results.

Table C.1. Comparison of results on 100-customer instances of Felipe et al. (2014)

		FORT	Matheuristic (A)			Matheuristic (B)		
S	*Instance*	*TC*	*TC*	*Time*	*% Imp*	*TC*	*Time*	*% Imp*
9	p1	73.42	65.12	197	11.31	65.43	22	10.88
	p2	65.97	60.49	173	8.31	60.81	19	7.82
	p3	66.98	62.95	175	6.01	61.46	21	8.24
	p4	70.96	65.41	191	7.83	65.44	19	7.78
	p5	78.22	73.82	190	5.63	75.27	19	3.77
	p6	72.16	65.72	174	8.93	65.94	21	8.63
	p7	73.87	65.40	171	11.46	67.33	20	8.86
	p8	62.70	57.21	186	8.76	57.44	21	8.40
	p9	72.45	62.80	186	13.32	63.15	21	12.83
	p10	66.70	60.21	168	9.73	60.21	18	9.73
5	p1	73.79	65.29	197	11.52	65.47	32	11.28
	p2	66.58	60.92	168	8.51	60.95	30	8.46
	p3	66.92	63.12	189	5.68	63.28	33	5.44
	p4	71.74	68.24	183	4.89	65.88	30	8.17
	p5	82.02	76.28	165	7.00	76.37	29	6.88
	p6	73.59	66.25	167	9.98	66.36	31	9.83
	p7	74.00	67.65	180	8.59	67.91	30	8.23
	p8	62.74	57.21	187	8.81	57.23	30	8.78
	p9	73.28	65.37	196	10.79	65.54	31	10.56
	p10	71.19	60.91	164	14.44	63.46	29	10.86

Table C.2. Comparison of results on 200-customer instances of Felipe et al. (2014)

		FORT	Matheuristic (A)			Matheuristic (B)		
S	*Instance*	*TC*	*TC*	*Time*	*% Imp*	*TC*	*Time*	*% Imp*
9	p1	-	105.54	768	-	110.49	80	-
	p2	108.24	96.40	757	10.94	97.47	83	9.95
	p3	110.21	97.34	827	11.68	99.06	77	10.12
	p4	108.69	95.78	794	11.87	99.54	77	8.42
	p5	117.44	103.65	781	11.74	106.33	75	9.46
	p6	111.26	98.17	869	11.76	101.66	83	8.62
	p7	110.42	98.48	773	10.81	102.46	78	7.21
	p8	102.37	93.66	813	8.51	94.65	84	7.54
	p9	110.08	97.41	821	11.51	101.62	86	7.68
	p10	114.71	101.06	780	11.90	104.86	73	8.59
5	p1	124.11	110.48	706	10.98	114.43	113	7.80
	p2	110.15	98.49	788	10.59	101.97	126	7.43
	p3	109.64	99.18	718	9.54	103.34	117	5.74
	p4	112.65	99.03	769	12.09	100.29	122	10.97
	p5	121.81	104.88	703	13.90	109.69	119	9.95
	p6	115.05	102.23	831	11.14	103.10	127	10.38
	p7	113.76	102.52	805	9.88	103.32	124	9.17
	p8	106.70	92.42	835	13.38	96.71	129	9.36
	p9	113.46	100.32	792	11.58	100.75	129	11.21
	p10	116.23	103.12	749	11.28	106.60	117	8.28

Table C.3. Comparison of results on 400-customer instances of Felipe et al. (2014)

		FORT	Matheuristic (A)			Matheuristic (B)		
S	*Instance*	*TC*	*TC*	*Time*	*% Imp*	*TC*	*Time*	*% Imp*
9	p1	198.48	179.56	2715	9.53	182.81	333	7.90
	p2	196.50	176.34	2928	10.26	180.94	341	7.92
	p3	195.72	177.34	2631	9.39	183.52	335	6.23
	p4	190.28	172.00	3292	9.61	179.11	329	5.87
	p5	192.67	176.13	2880	8.59	181.20	348	5.95
	p6	200.66	179.52	2810	10.53	184.82	311	7.89
	p7	194.40	176.17	2903	9.38	182.97	318	5.88
	p8	194.92	176.03	3103	9.69	183.16	332	6.03
	p9	198.16	175.12	2976	11.63	182.46	334	7.92
	p10	-	177.88	3123	-	183.30	311	-
5	p1	206.60	182.70	3265	11.57	189.73	545	8.17
	p2	201.98	180.38	3393	10.69	181.00	542	10.39
	p3	207.55	183.56	2780	11.56	188.32	539	9.27
	p4	192.96	177.07	3765	8.24	187.02	555	3.08
	p5	199.21	177.34	3386	10.98	184.44	556	7.42
	p6	206.75	184.41	3200	10.80	188.15	505	9.00
	p7	205.96	182.73	3261	11.28	190.41	541	7.55
	p8	195.06	181.10	3203	7.16	191.48	509	1.84
	p9	207.38	184.38	3173	11.09	191.94	531	7.44
	p10	208.33	182.41	3046	12.44	189.35	503	9.11

Bibliography

Adulyasak, Y., Cordeau, J-F., Jans, R. (2014). Optimization-based adaptive large neighborhood search for the production routing problem. Transportation Science 48(1): 20−45.

Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., Margaritis, D. (2014). Electric vehicle routing problem with industry constraints: Trends and insights for future research. Transportation Research Procedia, 3: 452−459.

Aksen, D., Kaya, O., Salman, F. S., Tüncel, Ö. (2014). An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. European Journal of Operational Research, 239(2): 413−426.

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

Artmeier, A., Haselmayr, J., Leucker, M. and Sachenbacher, M. (2010). The optimal routing problem in the context of battery-powered electric vehicles. 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability, Bologna, Italy.

Azi, N., Gendreau, M., Potvin, J-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers & Operations Research 41: 167−173.

Barco, J., Guerra, A., Muñoz, L., Quijano, N. (2013). Optimal routing and scheduling of charge for electric vehicles: Case study. Technical Report. Available online: <http://arxiv.org/pdf/1310.0145.pdf> (last accessed 10.04.2018)

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

Bramel, J., Simchi-Levi, D. (1995). A location based heuristic for general routing problems. Operations Research 43(4): 649−660.

Braysy, O., Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transportation Science 39(1): 104−118

Breunig, U., Baldacci, R., Hartl, R. F., Vidal, T. (2018). The Electric Two-echelon Vehicle Routing Problem. Technical Report, available at: https://arxiv.org/pdf/1803.03628.pdf

Bruglieri, M., F. Pezzella, O. Pisacane, S. Suraci. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electronic Notes in Discrete Mathematics, 47: 221−228.

Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O. (2016). A new mathematical programming model for the green vehicle routing problem. Electronic Notes in Discrete Mathematics, 55: 89−92.

Bruglieri, M., Mancini, S., Pisacane, O. (2018). Solving the green vehicle routing problem with capacitated alternative fuel stations. In Proceedings of 16th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Paris, France: 196−199.

CALSTART. (2013). “Technologies, challenges and opportunities: I-710 Zero-emission freight corridor vehicle systems”. Available at: http://www.calstart.org/Libraries/I-710_Project/Technologies_Challenges_and_Opportunities_I-710_Zero-Emission_Freight_Corridor_Vehicle_Systems.sflb.ashx (Last accessed on October 27, 2015)

Çatay, B., Keskin, M. (2017). The impact of quick charging stations on the route planning of electric vehicles. Proceedings of the 2017 IEEE Symposium on Computers and Communications: 152−157.

Chan, C. C. (2002). The state of the art of electric and hybrid vehicles. Proceedings of the IEEE, 90(2): 247−275.

Chan, C.C. (2007). The state of the art of electric, hybrid, and fuel cell vehicles, Proceedings of the IEEE 95(4) :704−718.

Chen, J. F., Wu, T. H. (2006). Vehicle routing problem with simultaneous deliveries and pickups. The Journal of the Operational Research Society 57(5): 579−587.

Chen, L., Hà, M.H., Langevin, A., Gendreau, M. (2014). Optimizing road network daily maintenance operations with stochastic service and travel times. Transportation Research Part E: Logistics and Transportation Review, 64: 88−102.

Çimen, M., Soysal, M. (2017). Time-dependent green vehicle routing problem with stochastic vehicle speeds: An approximate dynamic programming algorithm. Transportation Research Part D: Transport and Environment, 54: 82−98.

Conrad, R. G., Figliozzi, M. A. (2011). The recharging vehicle routing problem. In: Doolen, T. and Van Aken, E. (eds.) Proceedings of the 2011 Industrial Engineering Research Conference.

Coplon-Newfield, G. and Park. S.J. (2017). Corporate Fleets Making the Switch to Electric Vehicles. https://www.ecowatch.com/corporate-fleets-electric-vehicles-2499038220.html (last accessed: 26.07.2018)

Dayarian, I., Crainic, T. G., Gendreau, M., Rei, W. (2016). An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 95: 95−123.

Demir, E., Bektaş, T., Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. European Journal of Operational Research 223(2): 346−359.

den Boer, E., Aarnink, S., Kleiner, F., Pagenkopf, J. (2013). Zero emission trucks: An overview of state-of-the-art technologies and their potential. Available at: https://www.cedelft.eu/publicatie/zero_emission_trucks/1399. (Last accessed on October 27, 2015)

Desaulniers, G., Errico, F., Irnich, S. and Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. Operations Research, 64(6): 1388−1405.

Ding, N., Batta, R. and Kwon, C. (2015). Conflict-Free Electric Vehicle Routing Problem with Capacitated Charging Stations and Partial Recharge. Technical Report, available at: http://www.acsu.buffalo.edu/~batta/Nan%20Ding.pdf

Edelstein, S. (2016). Netherlands joins Norway in plans to end new gas, diesel car sales by 2025, <http://www.greencarreports.com/news/1103507_netherlands-joins-norway-in-plans-to-end-new-gas-diesel-car-sales-by-2025> (last accessed 10.04.2018)

Electrification Coalition. (2013). State of the plug-in electric vehicle market. Available at: https://www.pwc.com/gx/en/automotive/industry-publications-and-thought-leadership/assets/pwc-ec-state-of-pev-market-final.pdf. (Last accessed on October 27, 2015)

Emeç, U., Çatay, B., Bozkaya, B. (2016). An adaptive large neighborhood search heuristic for an e-grocery delivery routing problem. Computers & Operations Research, 69: 109−125.

Erdogan, S., Miller-Hooks, E. (2012). A green vehicle routing problem. Transportation Research Part E 48: 100−114.

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

European Commission White Paper on Transport 2011, Roadmap to a single European transport area: Towards a competitive and resource-efficient transport system. <http://ec.europa.eu/transport/themes/strategies/2011_white_paper_en.htm> (last accessed 09.10.2017)

Fairley, P. (2017). 10 Giant Companies Commit to Electric Vehicles, Sending Auto Industry a Message. Inside Climate News, https://insideclimatenews.org/news/19092017/electric-cars-ev100-coalition-charging-fleet-ikea-dhl (last accessed 23.08.2018)

Felipe, Á., Ortuño, M. T., Righini, G., Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transportation Research Part E, 71: 111−128.

Feng, W. and Figliozzi, M. (2013). An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. Transportation Research Part C: Emerging Technologies 26: 135−145.

Fleischmann, B., Gnutzmann, S., Sandvoß, E. (2004). Dynamic vehicle routing based on online traffic information. Transportation Science, 38(4): 420−433.

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., Stobbe, M. (2017). A metaheuristic for the time-dependent pollution-routing problem. European Journal of Operational Research, 259(3): 972−991.

Froger, A., Mendoza, J.E., Jabali, O., Laporte, G. (2017a). New Formulations for the Electric Vehicle Routing Problem with Nonlinear Charging Functions. Technical Report, available at; https://hal.archives-ouvertes.fr/hal-01559507/file/CIRRELT-2017-30.pdf

Froger, A., Mendoza, J.E., Jabali, O., Laporte, G. (2017b). A matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations. Technical Report, available at; https://hal.archives-ouvertes.fr/hal-01559524/document

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

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

Goeke, D. and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. European Journal of Operational Research, 245(1): 81−99.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European Journal of Operational Research, 254(1): 80−91.

Greenhouse gas emission statistics. (2016). <http://ec.europa.eu/eurostat/statistics-explained/index.php/Greenhouse_gas_emission_statistics> (last accessed 10.04.2018)

Haghani, A., Jung, S.A. (2005). Dynamic vehicle routing problem with time-dependent travel times. Computers & Operations Research, 32(11): 2959−2986.

Hashimoto, H., Yagiura, M., Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. Discrete Optimization, 5(2): 434−456.

Hemmelmayr, V. C., Cordeau, J-F., Crainic, T.G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research 39(12): 3215−3228.

Hiermann, G., Hartl, R.F., Puchinger, J. and Vidal, T. (2018). Routing a Mix of Conventional, Plug-in Hybrid, and Electric Vehicles. European Journal of Operational Research, 272(1): 235−248

Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. European Journal of Operational Research, 252(3): 995−1018.

Hof, J., Schneider, M., Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. Transportation Research Part B: Methodological, 97: 102−112.

European Commission Statistical Pocketbook 2017, available at: https://ec.europa.eu/transport/sites/transport/files/pocketbook2017.pdf (Last accessed on December 15, 2018)

Huang, Y., Zhao, L., Van Woensel, T. and Gross, J.P., (2017). Time-dependent vehicle routing problem with path flexibility. Transportation Research Part B: Methodological, 95: 169−195.

Ioannou, M., Kritikos, M., Prastacos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows, The Journal of the Operational Research Society 52(5): 523−537.

Jabali, O., Van Woensel, T., de Kok, A. (2012). Analysis of travel times and CO_2 emissions in time-dependent vehicle routing. Production and Operations Management, 21(6): 1060−1074.

Jie, W., Yang, J., Zhang, M., Huang, Y. (2018). The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology. European Journal of Operational Research, doi: 10.1016/j.ejor.2018.07.002

Jung, S., Haghani, A. (2001). Genetic algorithm for the time-dependent vehicle routing problem. Transportation Research Record: Journal of the Transportation Research Board, 1771: 164−171.

Khan, S. (2016). Germany pushes to ban petrol-fuelled cars within next 20 years. The Independent, <http://www.independent.co.uk/news/world/europe/germany-petrol-car-ban-no-combustion-diesel-vehicles-2030-a7354281.html> (last accessed 12.11.2018)

Koç, Ç., (2016). A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. Transportation Research Part C: Emerging Technologies, 68: 265−284.

Koç, Ç., Bektaş, T., Jabali, O. and Laporte, G. (2016). The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm. European Journal of Operational Research, 248(1): 33−51.

Koç, Ç., Jabali, O., Laporte, G. (2018). Long-haul vehicle routing and scheduling with idling options. Journal of the Operational Research Society. 69(2): 235−246.

Kopfer, H. and Vornhusen, B. (2017). Energy vehicle routing problem for differently sized and powered vehicles. Journal of Business Economics: https://doi.org/10.1007/s11573-018-0910-z.

Kullman, N.D., Goodson, J.C., Mendoza, J.E. (2018). Dynamic electric vehicle routing with mid-route recharging and uncertain availability. Technical report, available at: https://hal.archives-ouvertes.fr/hal-01814644.

Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research 7: 285−300.

Lee, D.Y., V.M. Thomas, M.A. Brown. (2013). Electric urban delivery trucks: Energy use, greenhouse gas emissions, and cost-effectiveness, Environmental science & technology 47(14): 8022−8030.

Leggieri, V., Haouari, M. (2017). A practical solution approach for the green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 104: 97−112.

Lei, H., Laporte, G. and Guo, B. (2012). A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. Top, 20(1): 99−118.

Li-ying, W., Yuan-bin, S. (2015). Multiple Charging Station Location-Routing Problem with Time Window of Electric Vehicle. Journal of Engineering Science and Technology Review, 8(5): 190−201.

Li, C., Ding, T., Liu, X., Huang, C. (2018). An electric vehicle routing optimization model with hybrid plug-in and wireless charging systems. IEEE Access, 6: 27569−27578.

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

Li, Y., Chen, H., Prins, C. (2016). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. European Journal of Operational Research, 252(1): 27−38.

Lin, J., Zhou, W., Wolfson, O. (2016). Electric vehicle routing problem. Transportation Research Procedia 12, 508−521.

Luo, Z., Qin, H., Zhang, D., Lim, A. (2016). Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. Transportation Research Part E: Logistics and Transportation Review, 85: 69−89.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laporte, G. (2018). The green mixed fleet vehicle routing problem with partial battery recharging and time windows. Computers & Operations Research (in press).

Malandraki, C., Daskin, M. (1992) Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. Transportation Science, 26(3): 185−200.

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

Montoya, A., Gueret, C., Mendoza, J. E., Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological, 103: 87−110.

Muller, L.F. (2009). An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. Proceedings of the VIII Metaheuristics International Conference (MIC 2009), Hamburg, Germany.

Muller, L.F., Spoorendonk, S., Pisinger, D. (2012). A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. European Journal of Operational Research 218(3): 614−623.

Nahum, O.E., Hadas, Y. (2009). Developing a model for the stochastic time-dependent vehicle-routing problem. In International Conference on Computers and Industrial Engineering, CIE 2009. Troyes, France: IEEE: 118−123.

Nesterova, N., H. Quak, S. Balm, I. Roche-Cerasi, T. Tretvik. (2013). Project FREVUE deliverable D1.3: State of the art of the electric freight vehicles implementation in city logistics. TNO and SINTEF. European Commission Seventh framework programme.

Omidvar, A., Tavakkoli-Moghaddam, R. (2012). Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. Proceedings of the IEEE International Energy Conference and Exhibition, Florence, Italy: 1089−1094.

Paz, J., Granada-Echeverri, M., Escobar, J. (2017). The multi-depot electric vehicle location routing problem with time windows. International Journal of Industrial Engineering Computations, 9(1): 123−136.

Paz, J., Granada-Echeverri, M., Escobar, J., (2018). The multi-depot electric vehicle location routing problem with time windows. International Journal of Industrial Engineering Computations, 9(1): 123−136.

Pelletier, S., Jabali, O. and Laporte, G., (2016). 50th anniversary invited article—goods distribution with electric vehicles: review and research perspectives. Transportation Science, 50(1): 3−22.

Pelletier, S., Jabali, O., Laporte, G. (2018). Charge scheduling for electric freight vehicles. Transportation Research Part B: Methodological, 115: 246−269.

Pelletier, S., Jabali, O., Laporte, G., Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. Transportation Research Part B: Methodological, 103, 158−187.

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

Pisinger, D., Ropke, S. (2010). Large neighborhood search. In: Gendreau M, Potvin J-Y, (eds.) Handbook of Metaheuristics 146: 399−419.

Pollet, B.G., I. Staffell, J.L. Shang. (2012). Current status of hybrid, battery and fuel cell electric vehicles: From electrochemistry to market prospects, Electrochimica Acta, 84: 235−249.

Potvin, J.-Y., Xu, Y., Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. Computers & Operations Research, 33(4): 1129−1137.

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

Ribeiro, G.M., Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Computers & Operations Research 39(3): 728−735.

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

Ropke, S., Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. European Journal of Operational Research 171(3): 750−775.

Sassi, O., Cherif, W. R., Oulamara, A. (2014). Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. Technical report, available at: https://hal.archives-ouvertes.fr/hal-01083966/

Schiffer, M., Walther, G. (2017). The electric location routing problem with time windows and partial recharging. European Journal of Operational Research, 260(3): 995−1013.

Schiffer, M., Schneider, M., Laporte, G. (2018a). Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. European Journal of Operational Research, 265(2): 517−532.

Schiffer, M., Walther, G. (2018b). Strategic planning of electric logistics fleet networks: A robust location-routing approach. Omega, 80: 31−42.

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

Schneider, M. Stenger, A., Goeke, D. (2014). The electric vehicle routing problem with time windows and recharging stations. Transportation Science 48(4): 500−520.

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

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming, Springer, New York: 417−431.

Shi, Y., Boudouh, T., Grunder, O. and Wang, D. (2018). Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. Expert Systems with Applications, 102: 218−233.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2): 254−265.

Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M. and Zhong, H. (2010). A model and algorithm for the courier delivery problem with uncertainty. Transportation science, 44(2): 193−205.

Suzuki, Y. (2014). A variable-reduction technique for the fixed-route vehicle-refueling problem. Computers & Industrial Engineering 67: 204−215.

Sweda, T.M., Dolinskaya, I.S., Klabjan, D. (2017a). Adaptive routing and recharging policies for electric vehicles. Transportation Science, 51(4): 1326−1348.

Sweda, T.M., Dolinskaya, I.S., Klabjan, D. (2017b). Optimal recharging policies for electric vehicles. Transportation Science 51:2: 457−479.

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

Taş, D., Gendreau, M., Jabali, O. and Laporte, G., (2016). The traveling salesman problem with time-dependent service times. European Journal of Operational Research, 248(2): 372−383.

Toth, P., Vigo, D. (2002). An overview of vehicle routing problems. In: Toth P, Vigo D, (eds.) The Vehicle Routing Problem, SIAM, Philadelphia: 1−26.

Touati-Moungla, N., Jost, V. (2012). Combinatorial optimization for electric vehicles management. Journal of Energy and Power Engineering 6(5): 738−743.

U.S. Department of State Policy Statement & Executive Summary <http://www.state.gov/documents/organization/203933.pdf> (last accessed 10.04.2018)

U.S. Department of Transportation, Bureau of Transportation Statistics. <http://www.bts.gov/> (last accessed 10.04.2018)

U.S. Energy Information Administration. https://www.eia.gov (last accessed: 26.07.2018)

U.S. Environmental Protection Agency, Sources of Greenhouse Gas Emissions. <http://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions> (last accessed 10.04.2018)

United Nations, Framework Convention on Climate Change, Paris Agreement, <http://unfccc.int/paris_agreement/items/9485.php> (last accessed 10.04.2018)

United States Climate Action Report 2014, available at: https://2009-2017.state.gov/documents/organization/219038.pdf (Last accessed on December 15, 2018)

United Nations, Framework Convention on Climate Change, Paris Agreement, <http://unfccc.int/paris agreement/items/9485.php> (last accessed 18.07.2018)

United States Environmental Protection Agency, 2016, available at: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions (Last accessed on October 25, 2015)

Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N. (2008). Vehicle routing with dynamic travel times: A queueing approach. European Journal of Operational Research, 186(3): 990−1007.

Villegas, J., Guéret, C., Mendoza, J.E. and Montoya, A. (2018). The technician routing and scheduling problem with conventional and electric vehicle. Technical report, available at: \\https://hal.archives-ouvertes.fr/hal-01813887/document

Wang, H. and Cheu, R. L. (2012). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations, Journal of the Transportation Research Board 2352, 1−10.

Wang, H., Shen, J. (2007). Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. Applied Mathematics and Computation 190, 1237−1249

Wang, Y. W., Lin, C. C., Lee, T. J. (2018). Electric vehicle tour planning. Transportation Research Part D: Transport and Environment, 63: 121−136.

Wen, M., Linde, E., Ropke, S., Mirchandani, P. and Larsen, A., (2016). An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. Computers & Operations Research, 76: 73−83.

White Paper on Transport (2011). Roadmap to a single European transport area: Towards a competitive and resource-efficient transport system, Publications Office of the European Union, Luxembourg.

Worley, O., Klabjan, D. (2012). Simultaneous vehicle routing and charging station siting for commercial electric vehicles. In IEEE International Electric Vehicle Conference, Greenville, SC: 1−3.

Wyman, O. (2018). Governmental incentives to encourage electric-vehicle sales and use. https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2018/march/Governmental-Incentives-To-Encourage-%20Electric-Vehicle-Sales-And-Use-exhibit.pdf (last accessed 18.07.2018)

Yang, J., Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. Computers & Operations Research, 55: 217−232.

Yilmaz, M. and Krein, P.T. (2013). Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles. IEEE Transactions on Power Electronics, 28(5): 2151−2169.

Zhang, J., Lam, W.H. and Chen, B.Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service. Networks and Spatial Economics, 13(4): 471−496.

Zhang, X., Yao, J., Liao, Z., Li, J. (2018a). The Electric Vehicle Routing Problem with Soft Time Windows and Recharging Stations in the Reverse Logistics. In International Conference on Management Science and Engineering Management: 171−182. Springer, Cham.

Zhang, S., Gajpal, Y., Appadoo, S. S., Abdulkader, M. M. S. (2018b). Electric Vehicle Routing Problem with Recharging Stations for Minimizing Energy Consumption. International Journal of Production Economics, 203: 404−413.