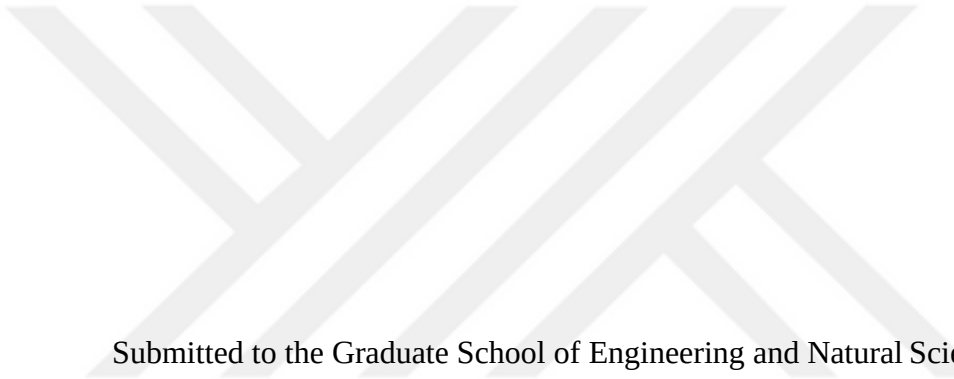


**ROUTE PLANNING OF ELECTRIC FREIGHT VEHICLES BY
CONSIDERING INTERNAL AND ENVIRONMENTAL CONDITIONS**

by
Sina Rastani



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

Sabancı University

Aug 2020

ROUTE PLANNING OF ELECTRIC FREIGHT VEHICLES BY
CONSIDERING INTERNAL AND ENVIRONMENTAL CONDITIONS

APPROVED BY:

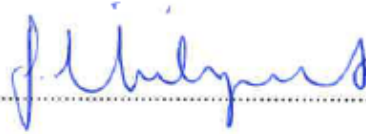
Prof. Dr. Bülent Çatay
(Thesis Supervisor)



Asst. Prof. Dr. Duygu Taş Küten



Prof. Dr. Tonguç Ünlüyurt



Asst. Prof. Dr. Umman Mahir Yıldırım



Asst. Prof. Dr. Tuğçe Yüksel



DATE OF APPROVAL: 27/Aug/2020



SINA RASTANI 2020 ©

All Rights Reserved

ABSTRACT

ROUTE PLANNING OF ELECTRIC FREIGHT VEHICLES BY CONSIDERING INTERNAL AND ENVIRONMENTAL CONDITIONS

Sina Rastani

Industrial Engineering Ph.D. Dissertation, Aug 2020

Dissertation Supervisor: Prof. Bülent Çatay

Keywords: electric vehicle routing, energy consumption, metaheuristics, green logistics

Electric freight vehicles have strong potential to reduce emissions stemming from logistics operations; however, their limited range still causes critical limitations. Range anxiety is directly related to the total amount of energy consumed during trips. There are several operational factors that affect the energy consumption of electric vehicles and should be considered for accurate route planning. In this thesis, we investigate the effect of ambient temperature, cargo weight, road gradient, and regenerative braking process on the fleet composition, energy consumption, and routing decisions in last-mile delivery operations. First, we consider the influence of ambient temperature on the energy consumption of the vehicle. Cabin heating or cooling may significantly increase the energy discharged from the battery during the trip and reduce the driving range. Additionally, cold temperatures decrease battery efficiency and cause performance losses. We formulate this problem as a mixed-integer linear program and solve the small-size instances using a commercial solver. For the large-size instances we resort to an Adaptive Large Neighborhood Search method. We also provide a case study based on the real data provided by Ekol Logistics in their Adana operations. Then, we propose new preprocessing techniques to reduce the problem size and enhance the computational performance of the solution methods. Furthermore, we develop an algorithm that can be used to identify if a problem instance is infeasible. Our experimental study validates the performance of the proposed preprocessing techniques and feasibility check algorithm. Next, we take into account the effect of cargo weight on the energy consumption and routing decisions. We formulate three alternative mathematical models and investigate their effectiveness. We also develop a Large Neighborhood Search (LNS) method by

using an exact method to repair the partial solution. Finally, we tackle the problem involving cargo weight and road gradient by considering regenerative braking. Considering the road gradient, a loaded vehicle going uphill will consume significantly more energy. On the other hand, when it travels downhill it can recharge its battery through recuperation. For this problem, we introduce a new dataset generated using the benchmark data from the literature. We adapt our LNS and perform an extensive computational study using the generated data. Overall, our results show that the route plans made without considering any of these factors may lead to inefficiencies, unforeseen costs, and disruptions in logistics operations.



ÖZET

İÇSEL VE ÇEVRESEL KOŞULLARI GÖZ ÖNÜNDE BULUNDURARAK ELEKTRİKLİ YÜK ARAÇLARININ ROTA PLANLAMASI

Sina Rastani

Endüstri Mühendisliği, Ağustos 2020

Tez Danışmanı: Prof. Dr. Bülent Çatay

Anahtar Kelimeler: elektrikli araç rotalama, enerji tüketimi, metasezgisel, yeşil lojistik

Elektrikli araçların kullanımı lojistik operasyonlarından kaynaklanan zararlı gazların salımının azaltılmasında önemli bir potansiyel sunar. Ancak, bu araçların menzillerinin kısa olması yaygın kullanımlarını sınırlayan en önemli faktördür. Menzil kaygısı, yolculuk sırasında tüketilen toplam enerji miktarı ile doğrudan ilişkilidir. Doğru rota planlaması için elektrikli araçların enerji tüketimini etkileyen operasyonel etkenlerin dikkate alınması gerekir. Bu tezde bu etkenlerden ortam sıcaklığı, yük ağırlığı, yol eğimi ve rejeneratif frenleme ele alınmıştır. Araç kabininin ısıtılması veya soğutulması yolculuk boyunca tüketilen enerjiyi yüksek ölçüde artırabilir ve buna bağlı olarak da aracın menzilini kısaltır. Bunun yanında, çok soğuk hava koşullarının batarya verimini azalttığı ve performans kaybına neden olduğu bilinmektedir. Ayrıca, araçta taşınan yükün ağırlığına bağlı olarak enerji tüketimi de artmaktadır. Yol eğimi göz önünde bulundurulduğunda, yokuş yukarı giden yüklü bir aracın enerji tüketimi düz yolda ilerleyen bir araca göre daha fazla olacaktır. Öte yandan, araç yokuş aşağı hareket ettiğinde ise geri kazanım yoluyla bataryasını şarj edebilmektedir. Bu çalışmada, kentsel lojistik operasyonlarında bu etkenlerin araç filosu kompozisyonuna, toplam enerji tüketimine ve rotalama kararlarına nasıl etki ettikleri incelenmektedir. Farklı problemler için bu etkenleri göz önünde bulunduran matematiksel programlama modelleri sunulmakta, küçük boyutlu problemler için bir eniyileme yazılımı ile bu modeller çözülürken büyük boyutlu problemleri çözmek için Geniş Komşuluk Arama metasezgisel yaklaşımından faydalanılmaktadır. Bu kapsamda, problem boyutunu küçültmek ve çözüm yöntemlerinin hesaplama performansını artırmak için yeni ön işleme yöntemleri de önerilmektedir. Ayrıca, bir problemin olurlu olup olmadığını belirlemek için bir

algoritma geliştirilmiştir. Sunulan modellerin ve geliştirilen çözüm yöntemlerinin performansı literatürdeki veri setleri kullanılarak kapsamlı deneysel çalışmalarla incelenmiştir. Ayrıca, literatürde yol eğimini içeren bir veri seti bulunmadığı için buna yönelik yeni bir veri seti sunulmuştur. Elde edilen sonuçlar, bu etkenler dikkate alınmadan yapılan rota planlarının lojistik operasyonlarında öngörülmeven maliyetlere ve aksaklıklara yol açabileceğini göstermektedir. Ayrıca, önerilen ön işleme yöntemlerinin ve olurluluk kontrol algoritmasının etkinlikleri yapılan deneylerle gösterilmiştir.



ACKNOWLEDGEMENTS

Foremost, I would like to express my deepest gratitude to my advisor Prof. Bülent Çatay for his support, patience, motivation, and immense knowledge. I have learnt many things from him and being his student has been an honor and pleasure. I could not have imagined having a better advisor and mentor for my Ph.D. study.

I am much grateful to Professors Gündüz Ulusoy, Kemal Kılıç, Tonguç Ünlüyurt, Nilay Noyan, Ilker Birbil, Güvenç Sahin, Baris Balcıoglu, Murat Kaya and Kerem Bülbül 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 Professors Tonguç Ünlüyurt, Tuğçe Yüksel Umman Mahir Yıldırım, Duygu Taş Küten for their valuable advises and insightful comments throughout my study. They have always supported me with their positive energy.

I would like to express my special thanks to Deniz Mortazavi who has stood by me all the time. She has always been much understanding to me and encouraged me to do the right. Along with her, I would like to express my appreciation to my family for always motivating me. Without their supports, I would not start graduate studies.

I would like to thank Milad Hassani for his valuable support and continuous encouragement throughout my study. Many thanks to my friends Amin Ahmadi Digehsara, Siamak Naderi Varandi, Merve Keskin, Sonya Javadi, Nozir Shokirov, Ece Naz Duman, Saeedeh Basir and Yasaman Karimian for always supporting me. We have shared not only the Logistics Lab but also many joyful moments together. Also, special thanks to Sahand, Ehsan, Faraz, Nasim, Arash, Naeime, Vahid, Ali, Meysam, Mahsa, Navid, Pegah, Mohamadreza and Farzad for all their supports and great moments we had during my Ph.D. studies.

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 thank Ekol Logistics for providing the data for the case study.

This thesis was supported by The Scientific and Technical Research Council of Turkey through Grant #118M412 and by Sabanci University Internal Research Project I.A.CF-17-01699.

Finally, I would like to express my appreciation to all health workers working hard during the 2020 Covid-19 Pandemic to save lives.





TABLE OF CONTENTS

List of Tables.....	xi
List of Figures.....	xii
List of Abbreviations.....	xiv
1. Introduction	1
1.1. Advantages and disadvantages of using EVs.....	2
1.2. Overview of the Fuel and Energy Consumption Approaches.....	2
1.3. Electric Vehicle Routing Problem	4
1.4. Thesis Organization	5
2. Effects of Ambient Temperature on the Route Planning of Electric Freight Vehicles	7
2.1. Introduction to Electric Vehicle Routing Problem with Time Windows and Literature Review	7
2.2. Problem description and mathematical model	10
2.2.1. Temperature effect on energy consumption	11
2.2.2. Mathematical formulation.....	14
2.3. Solution methodology	18
2.4. Computational study	19
2.4.1. The influence of ambient temperature on routing decisions in small-size instances	20
2.4.2. The influence of ambient temperature on routing decisions in large-size instances	22
2.5. Case study	24
2.6. Discussion and conclusion	28

3. Speed-up Techniques for Solving the Electric Vehicle Routing Problem with Time Windows.....	30
3.1. Introduction to Electric Vehicle Routing Problem with Time Windows.....	30
3.2. Problem statement.....	32
3.3. Network reduction	35
3.3.1. Connectivity of the depot to a customer	36
3.3.2. Connectivity of a customer to another customer	37
3.3.3. Connectivity of a customer to the depot	39
3.3.4. Valid inequalities	41
3.4. Feasibility check algorithm.....	41
3.5. Numerical results	43
3.5.1. Analysis of network reduction.....	44
3.5.2. Performance on small-size problems.....	45
3.5.3. Performance on large-size problems.....	47
3.5.4. Identifying infeasibility.....	51
3.6. Concluding remarks	52
4. Electric Vehicle Routing Problem with Time Windows and Cargo Weight....	53
4.1. Introduction to Electric Vehicle Routing Problem with Time Windows considering cargo load	53
4.2. Problem definition and formulations	55
4.2.1. Energy consumption function.....	55
4.2.2. Mathematical models.....	56
4.3. Large Neighborhood Search algorithm.....	62
4.3.1. Destroy operators.....	63
4.3.2. Repair operators.....	64
4.3.3. Repair-opt operator	66
4.4. Experimental study	67
4.4.1. Analysis on the performance of the proposed models.....	68

4.4.2.	Analysis on the effect of load on the route planning	70
4.4.3.	Analysis on the repair-opt operator	71
4.5.	Conclusions and future directions.....	72
5.	Electric Vehicle Routing Problem with Time Windows considering Cargo Weight, Road Gradient and Regenerative Braking	73
5.1.	Introduction.....	73
5.2.	Problem description	74
5.2.1.	Energy consumption considering road gradient and regenerative braking	75
5.2.2.	Problem formulation	76
5.3.	Experimental design and computational study	77
5.3.1.	Data Generation	78
5.3.2.	Analysis on the effect of road gradient and regenerative braking on the route planning	80
5.4.	Conclusions and future research	82
6.	Concluding Remarks.....	84
	Appendix A Results for benchmark instances.....	87
	Appendix B Optimal route plans for different temperatures in Adana.....	90
	Appendix C Distance data of the distribution network in Adana.....	91
	Appendix D Parameters	92
	Appendix E Detailed result analyzing the performance of repair-opt operator	93
	Bibliography.....	95

LIST OF TABLES

Table 2.1 Energy consumption at different temperatures	12
Table 2.2 Mathematical notation for EVRPTW with ambient temperature	15
Table 2.3 Number of infeasible problems in small-size dataset for different temperature conditions.....	20
Table 2.4 The influence of ambient temperature on route plans in small-size instances	21
Table 2.5 Average results for large-size problems	22
Table 2.6 The influence of ambient temperature on route plans in large-size instances	23
Table 2.7 Case study data: time windows and demands of the customers	24
Table 3.1 Mathematical notation for EVRPTW	33
Table 3.2 Average customer-network densities after preprocessing	45
Table 3.3 Results for small-size instances in the mild case.....	45
Table 3.4 Results for small-size instances in the intermediate case	46
Table 3.5 Results for small-size instances in the intense case.....	46
Table 3.6 Result for large-size instances in mild case	48
Table 3.7 Result for large-size instances in the intermediate case	49
Table 3.8 Result for large-size instances in the intense case	50
Table 3.9 Summary of results for large-size instances	51
Table 3.10 Average results for identifying infeasibilities.....	52
Table 4.1 Comparison of results obtained using different models for the load-dependent case.....	68
Table 4.2 Result for small-size instances obtained using GUROBI and LNS for Load Independent and Load-dependent cases	69
Table 4.3 Result for large-size instances obtained using LNS for Load-Independent and Load-Dependent cases	70
Table 4.4 Average results for solving small and large-size instances without and with considering repair-opt operator in LNS algorithm	72
Table 5.1 Result for small-size instances using LNS for Level, Nearly Level, Very Gentle	

Slope cases	81
Table 5.2 Result for large-size instances using LNS for Level, Nearly Level, Very Gentle	
Slope cases	82



LIST OF FIGURES

Figure 2.1 Energy consumption vs. ambient temperature for Nissan Leaf	13
Figure 2.2 Optimal route plans that change according to varying temperatures	14
Figure 2.3 Monthly daytime highest/average/lowest temperatures in Adana during 2017	24
Figure 2.4 Geographical area.....	26
Figure 2.5 Optimal route plans at different ambient temperature conditions.....	27
Figure 3.1 Conditions for the connectivity of the depot to customer i	37
Figure 3.2 Conditions for the connectivity of customer i to customer j	39
Figure 3.3 Conditions for the connectivity of customer j to depot n+1.....	40
Figure 3.4 Feasibility check algorithm	43
Figure 3.5 Summary of results for small-size instances	47
Figure 5.1. Scatter network for “c101” with 100 customers	79
Figure 5.2. Network for "c101" with 100 customers.....	80

LIST OF ABBREVIATIONS

GHG: Global Greenhouse Gas	1
VRP: Vehicle Routing Problem.....	1
BEV: Battery Electric Vehicle.....	1
EV: Electric Vehicle	1
ICEV: Internal Combustion Engine Vehicle	1
PRP: Pollution Routing Problem	3
ALNS: Adaptive Large Neighborhood Search.....	3
SA: Simulated Annealing	4
TS: Tabu Search.....	4
EVRPTW: Electric Vehicle Routing Problem with Time Windows.....	5
MILP: Mixed-Integer Linear Programming	5
LNS: Large Neighborhood Search	5
AFV: Alternative Fuel Vehicle.....	7
AFS: Alternative Fuel Station.....	7
GVRP: Green Vehicle Routing Problem	7
VRPTW: Vehicle Routing Problem with Time Windows.....	8
VNS: Variable Neighborhood search	8
HVAC: Heating, Ventilation and Air Conditioning	9
SoC: State of Charge.....	10
MSI: Multi-Station Insertion	18
LDVRP: Load Dependent Vehicle Routing Problem.....	54

1. INTRODUCTION

Transportation sector is responsible from 14% of global anthropogenic emissions and 23% of energy-related global greenhouse gas (GHG) emissions around the world (Edenhofer et al., 2014; Raadal et al., 2011). About 75% of transport-related emissions can be attributed to road transport (International Energy Agency, 2017). Road transport is also a major source of air pollutants, particularly NO_x and PM_{2.5}. To reduce negative effects and mitigate emissions, governments are setting ambitious targets. The European Commission targets 60% reduction in transport-related GHG emissions by 2050 compared to 1990 levels (European Commission, 2011).

Urban transport is particularly important because road vehicles are mostly used in high population areas, which causes the concentration of emissions in the cities (International Energy Agency, 2016). In Europe, urban transport constitutes 23% of transport-related emissions, 6% of which is due to urban freight transport, i.e. transportation of goods (European Commission, 2013). City logistics, therefore, has a significant portion in the urban transport emissions and the EC targets “CO₂ free city logistics” by 2030 (European Commission, 2013). In addition, several cities have issued plans to ban domestic sales of new diesel and gasoline-powered cars as of 2025 in the Scandinavian countries and as of 2030 in most of the European countries (DW, 2018).

Replacing internal combustion engine vehicles (ICEVs) with battery electric vehicles (BEVs) is one of the most promising approaches to achieve these targets. This thesis aims to develop effective models and solution methods for the route planning of BEVs. Throughout this thesis, we will refer to a commercial BEV as EV (Electric Vehicle) in line with the Vehicle Routing Problem (VRP) literature.

1.1. Advantages and disadvantages of using EVs

The EVs use only electricity as an energy source, hence, they constitute a good alternative to gasoline and diesel-powered vehicles. Furthermore, EVs provide cost benefits during operation with lower energy consumption per distance traveled due to more efficient powertrains (Wu et al., 2015). Limited driving range, long recharge durations, inadequate charging infrastructure, and high acquisition costs are the major drawbacks of EVs. Logistics companies might have advantages regarding these drawbacks since they have the chance to plan their itineraries, therefore charging times and durations, and install their own charging stations at their depots (Giordano et al., 2017).

1.2. Overview of the Fuel and Energy Consumption Approaches

The amount of fuel consumed by a vehicle that causes pollution depends on load, speed, slope, weather conditions, acceleration, air density, vehicle's frontal area and other factors. A variety of models which are mostly based on simulation have been presented to calculate fuel consumption such as aaSIDRA and aaMOTION (Akcelik and Besley 2003) and the Comprehensive Modal Emission Model (Barth et al., 2005), which have been used to test various strategies for CO₂ reduction (Barth and Boriboonsomsin 2008). Palmer proposed an integrated routing and emission model for the freight vehicles and discussed the effect of speed on polluting under different congestion and time window scenarios. However, he did not consider vehicle loads in his model (Palmer 2007).

A vehicle routing and scheduling problem with time windows was addressed by Maden et al. which depends on the time of travel. They solved a case study of a fleet of delivery vehicles in the UK by applying a heuristic and reported up to 7% saving in CO₂ emissions (Maden et al., 2010). A similar problem to that Maden et al. was studied by Jabali et al. which tries to obtain optimal speed with respect to emission. They calculated the emission with a nonlinear function of speed. They did not consider other factors that affect the vehicle's emission. An iterative TS was proposed to solve VRP instances taken from the literature (Jabali et al., 2012). Hsu et al. addressed a VRP with energy considerations.

The problem's aim is to distribute perishable foods with means of vehicles with refrigerators. The objective is to minimize transportation, inventory, energy, and violations of time windows costs (Hsu et al., 2007).

Fuel and energy consumption were studied in VRP and Electric Vehicle Routing Problem (EVRP) recently, as it is becoming more and more important to decrease emission (in conventional vehicles) and increase driving range with a limited battery capacity in the electric fleet. In the VRP literature Bektaş and Laporte (2011) introduced the pollution routing problem (PRP) which is an extension of classical VRP that considers not only traveled distance, but also the amount of greenhouse emissions, fuel, travel times, and their costs. They stated that speed and load have the most imperative effect on the amount of pollution emitted by a vehicle. They used a function introduced by Barth et al. for calculating emission costs (Barth et al., 2005; Barth and Boriboonsomsin 2009). They showed that fuel consumption as a function of speed is a U-shape function which means that fuel consumption decreases and then increases as the speed increases. Bektaş and Laporte proposed mathematical models for PRP with or without time windows and illustrated their computational experiments performed on realistic instances (Bektaş and Laporte 2011).

Demir et al. (2011) analyzed and numerically compared several available freight transportation vehicle emission models. As a result, they showed U-shape diagrams corresponding to fuel consumption for three types of vehicles under various speed levels estimated by the engine power module (Demir et al., 2011). Suzuki (2011) developed an approach for the time-constrained, multiple-stop, truck-routing problem which minimizes the distance a vehicle should travel with a heavy load in a given tour by sequencing the customer visits such that heavier items are unloaded first while lighter items are unloaded later, and it considers the amount of fuel burned during the time a truck is detained at customer sites. This problem is a kind of load-dependent problem and minimizes fuel consumption and emission (Suzuki 2011).

Demir et al. (2012) proposed an Adaptive Large Neighborhood Search (ALNS) heuristic for solving the pollution routing problem which has time windows and determines the speed of each vehicle on each route segment in order to minimize a function which considers fuel, emission, and driver costs (Demir et al., 2012). Demir et al. (2014)

addressed an extension for the PRP named bi-objective PRP which includes two objective functions that minimize fuel consumption and driving time which are conflicting and are thus considered separately. They presented an ALNS combined with a speed optimization procedure to solve the bi-objective PRP (Demir et al., 2014).

Wu et al. (2015) studied electric vehicles' energy consumption. Their analyses showed that the EV is more efficient when driving on in-city routes than driving on freeway routes. Moreover, they analyzed the relations among the EV's power, the vehicle's speed, acceleration, and the roadway grade. They proposed an analytical EV power estimation model (Wu et al., 2015).

Suzuki (2016) addressed a PRP model that needs fewer user inputs. Their model incorporates only a subset of all factors affecting trucks' fuel consumption. Their solution approach treats the PRP which is a single-objective problem, as a dual-objective problem that minimizes distance traveled and vehicle payload. By means of Simulated Annealing (SA), a Pareto frontier for this dual-objective problem was approximated. Then a Tabu Search (TS) algorithm which explores only the regions near the frontier was applied in order to improve each component of the frontier (Suzuki 2016).

1.3. Electric Vehicle Routing Problem

The Electric Vehicle Routing Problem (EVRP) is an extension of the well-known Vehicle Routing Problem (VRP) where the fleet consists of EVs. The aim of VRP is to determine the minimum cost routes that serve a set of customers with known demands. The utilization of an EV fleet in logistics operations reduces the tailpipe emissions and help companies achieve their sustainability objectives while decreasing the operational costs. On the other hand, limited battery capacity, recharging strategies, and long charging durations bring additional complexity to the problem. These challenges have attracted the interest of many researchers and studies on EVRP has recently gained momentum.

1.4. Thesis Organization

In chapter 2 we consider the impact of ambient temperature on the fleet sizing, battery recharging and routing decisions within the context of EVRPs in logistics operations. Particularly, we focus on the EVRP with time windows (EVRPTW) by allowing partial charging. Ambient temperature can cause a rise in energy consumption of EVs since cabin heating or cooling may significantly increase the energy discharged from the battery during the trip and reduce the driving range. Additionally, cold temperatures decrease battery efficiency and cause performance losses. First, we present the Mixed-Integer Linear Programming (MILP) formulation of the problem. Next, we perform an extensive computational study based on benchmark data from the literature. For solving the small-size instances we use a commercial solver (CPLEX). For solving the large-size instances we employ an ALNS algorithm. We show how the fleet compositions and route plans change under different weather conditions using benchmark data from the literature as well as real data from a logistics company. This study is published in Transportation Research Part D: Transport and Environment as “Effects of ambient temperature on the route planning of electric freight vehicles” by Sina Rastani, Tuğçe Yüksel and Bülent Çatay.

Chapter 3 presents some reduction techniques and develops a preprocessing procedure to reduce the graph, hence the number of decision variables in EVRPTW. Furthermore, we propose an algorithm to identify whether a problem instance is feasible or not. Extensive computational tests are performed to investigate the performance of the proposed approaches. This study is submitted to Computers & Operations Research as “Speed-up techniques for solving the electric vehicle routing problem with time windows”.

In the 4th chapter, we address the load-dependent variant of EVRPTW with partial recharges by taking into account the energy consumption associated with the cargo carried on the vehicle. Carrying more load by an EV causes more energy consumption. We present the MILP formulation of the problem and perform an extensive experimental study to investigate the influence of load on the routing decisions. We solve small-size instances using a commercial solver (GUROBI), and for the large-size instances, we develop a Large Neighbourhood Search (LNS) algorithm. The results show that cargo

weight may create substantial changes in the route plans and fleet size. Additionally, we equipped the proposed LNS method with exact insertion operator by joining LNS metaheuristic coded in Python with GUROBI to obtain better solutions. This work is accepted to be published in ICLS 2020 edited volume by Springer as “Electric Vehicle Routing Problem with Time Windows and Cargo Weight”.

Chapter 5 studies EVRPTW with partial recharges by taking into account the energy consumption associated with the road gradient and the cargo carried on the vehicle. Traveling on an arc with a positive road gradient requires more energy comparing to an arc in a flat network. As the gradient of an arc slightly rises, the energy consumption per unit distance increases which can extremely increase the energy consumption on that arc. Furthermore, in the operations where the EVs deal with heavy loads, the effect of road gradient on the energy consumption intensifies since an EV moving uphill with heavy load requires more energy in order to finish its journey. On the other hand, if an EV traverses on an arc with a negative road gradient where the driver needs to push the brake pedal in order to travel with a constant speed, energy can be saved on the battery because of the regenerative braking technology. We generate data based on the benchmark datasets in the literature by assigning altitude to each node in the network. Clustering techniques are used to elevate the nodes in order to have a consistent dataset. We present an LNS algorithm to solve the small and large-size instances. Results show that considering road gradient along with cargo load can significantly change the routing decisions and it can make the problem infeasible in the networks with steep road slope.

Finally, concluding remarks and future directions of research are presented in the last chapter of the thesis.

2. EFFECTS OF AMBIENT TEMPERATURE ON THE ROUTE PLANNING OF ELECTRIC FREIGHT VEHICLES

2.1. Introduction to Electric Vehicle Routing Problem with Time Windows and Literature Review

The Electric Vehicle Routing Problem (EVRP) is an extension of the well-known Vehicle Routing Problem (VRP) where the fleet consists of EVs. The aim of VRP is to determine the minimum cost routes that serve a set of customers with known demands. The utilization of an EV fleet in logistics operations reduces the tailpipe emissions and help companies achieve their sustainability objectives while decreasing the operational costs. On the other hand, limited battery capacity, recharging strategies, and long charging durations bring additional complexity to the problem. These challenges have attracted the interest of many researchers and studies on EVRP has recently gained momentum.

Conrad and Figliozzi (2011) is the first study that considers an EV fleet within the context of VRP. In this problem, EVs are recharged at selected customer locations at a fixed cost. The objective is minimizing the fleet size and a total cost function associated with recharges, distance, and service time. Erdoğan and Miller-Hooks (2012) generalized the problem by considering alternative fuel vehicles (AFVs) and introduced the Green Vehicle Routing Problem (GVRP). The authors assumed that the fuel is consumed proportional to the distance, the tank is fully refueled at the alternative fueling stations (AFSs), and the refueling time is constant. The objective is to minimize the total distance. Wang and Cheu (2013) addressed a similar problem for an electric taxi fleet by also assuming full recharge strategy.

Schneider et al. (2014) introduced EVRP with Time Window (EVRPTW) by also assuming full recharge strategy. They formulated the mathematical programming model and proposed three hybrid Variable Neighborhood Search (VNS) and TS algorithms. They tested the performance of their algorithms using benchmark instances for GVRP and Multi-Depot VRP with Inter-Depot Routes. They also generated a new data set based on the well-known Solomon (1987) Vehicle Routing Problem with Time Windows (VRPTW) data. Afroditi et al. (2014) also developed a mathematical model for EVRPTW with full recharges and provided insights about the trends in the literature. Bruglieri et al. (2015) relaxed the full recharge assumption and developed a Variable Neighborhood Search Branching method to solve small-size instances. Keskin and Çatay (2016) also allowed partial recharges and implemented an Adaptive Large Neighborhood Search (ALNS) method by introducing several new removal and insertion mechanisms specific to the problem. Desaulniers et al. (2016) also attacked EVRPTW and attempted to solve four variants optimality using branch-price-and-cut algorithm.

Several extensions of EVRP and EVRPTW have been addressed in the literature such as the utilization of a mixed fleet of EVs and ICEVs (Goeke and Schneider, 2015; Sassi et al., 2015, Macrina et al., 2018, Hiermann et al., 2019), heterogeneous fleet of EVs (Hiermann et al., 2016), fast charging technologies (Felipe et al., 2014; Çatay and Keskin, 2017; Keskin and Çatay, 2018), nonlinear charging function (Montoya et al., 2017; Froger et al., 2019), battery swap stations (Yang and Sun, 2015; Hof et al., 2017; Paz et al., 2018), and time-dependent waiting times at stations (Keskin et al., 2019). In addition, EV fleets have also been considered within the framework of Location Routing Problem (Worley et al., 2012; Hof et al., 2017; Schiffer et al., 2018), Two-echelon VRP (Jie et al., 2019), and Two-stage EVRP (Basso et al., 2019) that integrates path finding with route planning.

In a parallel setting, Montoya et al. (2016) used a two-phase heuristic for solving GVRP. Bruglieri et al. (2016) also tackled GVRP and presented a three-index formulation to reduce the number of decision variables in the problem and proposed a method to eliminate the dominated stations. Recently, Bruglieri et al., (2018) developed a path-based exact approach to solve small size GVRP instances. For larger instances, they converted their exact method to a heuristic approach. Koç and Karaoğlu (2016) also introduced a new GVRP formulation with fewer constraints and decision variables, and implemented a Simulated Annealing (SA) method to solve it. Leggieri and Haouari (2017) presented a

new formulation for EVRPTW and proposed a reduction procedure to speed up solving the problem. A comprehensive survey about the use of EVs in distribution operations is provided in Pelletier et al. (2016).

Most of the aforementioned studies assume a constant rate of energy consumption per unit distance traveled. However, the EV energy consumption varies with operating conditions such as driving style (speed and acceleration), road profile, vehicle load, and weather. Among these, ambient temperature has a significant effect on EV's performance. Yuksel and Michalek (2015) showed that, compared to mild climate regions, energy consumption of EVs can rise, which can result in up to 41% decrease in the driving range. Temperature affects energy consumption due to heater use and decreased battery efficiency in cold temperatures, and increased use of air conditioning in hot temperatures. Neubauer and Wood (2014) showed that EV energy consumption can increase by 24% due to heating, ventilation, and air conditioning (HVAC) used in cold climates. Yi et al. (2018) studied the impact of ambient temperature on the energy consumption and demand for charging of an autonomous EV. Their aim is to determine the path from an origin node to a destination node as well as the recharging time at each intermediate charging station node. Using a taxi pick-up and drop-off dataset from New York City, they observed that in hot and cold temperatures the energy consumption and charging demand of the fleet can increase by 20% and 60%, respectively. Temperature effect in EVs is more dominant at cold weather compared to diesel/gasoline counterparts because EVs do not have the option to use excess engine temperature for cabin heating. Extreme temperatures might therefore cause considerable changes in route planning. Depending on the weather conditions, a larger fleet of EVs may be needed on certain seasons/days in order to perform the desired logistics operations and/or the EVs may need more frequent recharges because of the increase in energy consumption. In extreme conditions, it may even not be possible to find a feasible route plan.

The aim of this chapter is to investigate the impact of ambient temperature on routing decisions of EVs in logistics operations. Particularly, we focus on the EVRPTW by allowing partial charging (Keskin and Çatay, 2016). To the best of our knowledge, this is the first study that investigates the influence of ambient temperature on the fleet sizing, battery recharging and routing decisions within the context of EVRPs. Our contributions to the literature are twofold: (i) we extend the mathematical model of EVRPTW by

incorporating the effect of temperature in the energy consumption of the vehicles; and (ii) show how the fleet compositions and route plans change under different weather conditions using benchmark data from literature as well as real data from a logistics company.

The remainder of the article is organized as follows: Section 2.2 depicts the problem and formulates the mathematical programming model. Section 2.3 describes the methodology employed to solve it. Section 2.4 presents the computational results and discusses the influence of the ambient temperature on route plans and energy consumptions. Section 2.5 presents a case study based on the last mile delivery operations of Ekol Logistics in Southern Turkey. Final remarks and future research directions conclude this chapter.

2.2. Problem description and mathematical model

In this chapter, we address EVRPTW which involves a homogeneous fleet of EVs and a set of customers whose demands, time windows, and service durations are known. Similar to the previous studies the battery state of charge (SoC) decreases proportional to the distance traveled; however, we also take into account the effect of ambient temperature on the energy consumption during the trip. In addition, we allow partial recharging and its duration depends on the amount of energy transferred. Fully recharging the battery can shorten its lifespan (Sweda et al., 2017) and it is a common practice in the real world to operate within the first phase of recharging where the energy transferred is a linear function of the recharge duration in order to prolong the battery life (Pelletier et al., 2017). So, without loss of generality, we assume that the energy recharged at the stations is linear function of time. We assume that the EV can be recharged at most once between two consecutive customers, which is the practical case in urban logistics.

The change in energy consumption with temperature is contingent on the duration of the trip and effect of temperature on charging efficiency is not considered. It is assumed that the driver turns off the heating/cooling equipment in the recharging stations, as it is a time-consuming process. Without loss of generality, the energy consumption related to on-board auxiliary systems are neglected in this chapter. In addition, we assume that EVs

are recharged overnight and depart from the depot with full battery. In line with the EVRPTW literature, we adopt a hierarchical objective function where the primary objective is to minimize the fleet size whereas the secondary objective is to minimize total energy consumption (Schneider et al., 2014; Keskin and Çatay 2016).

2.2.1. Temperature effect on energy consumption

To estimate how energy consumption changes with temperature, we use a similar approach as in Yuksel and Michalek (2015) and construct a model based on real-world data collected from Nissan Leaf drivers over more than 7000 trips across North America. Publicly available data reports average driving range with respect to temperature and includes no other information on the trip and driver profiles. Energy consumption in kilowatt-hour per mile (kWh/mile) versus ambient temperature is shown in Figure 2.1. In their study, Yuksel and Michalek use a model obtained by fitting a single curve to the available data. To improve accuracy, we divide data into two and fit two separate polynomial curves for data points below and above 22 °C as shown by blue and red curves in Figure 2.1. The functional relationships between energy consumption per unit distance and temperature can be given as follows:

$$h_{LEAF}(T) = \begin{cases} 0.3392 - 0.005238 T - 0.0001078 T^2 + 1.047 \cdot 10^{-5} T^3 + \\ \quad 3.955 \times 10^{-7} T^4 - 1.362 \times 10^{-8} T^5 - 3.109 \times 10^{-10} T^6, & T < 22^\circ C \\ 0.4211 - 0.01627 T + 0.0004229 T^2, & T \geq 22^\circ C \end{cases} \quad (2.1)$$

where h_{LEAF} is in kWh/mile when T is in °C. Note that Nissan Leaf is a light duty passenger vehicle; however, we assume that the same kind of relation with temperature holds for all sizes of commercial vehicles used in logistics operations as well. In addition, since there is no further information available about the driver and trip profiles, we follow the same assumption in Yuksel and Michalek (2015) and we attribute the efficiency change given in Figure 2.1 only to the ambient temperature.

A similar study performed by National Renewable Energy Laboratory (NREL) for a fleet consisting of medium-duty EVs reports slightly higher energy consumption at cold temperatures (Duran et al., 2014). Their results show that the average energy consumption

of medium-duty EVs running in NY almost doubled in January 2013 when the average minimum temperature observed is -20°C as compared to May 2013 when the minimum temperature is around 18°C. According to the Nissan Leaf based data we used, the ratio between the similar temperatures is 1.52 (i.e. 1.52 times more energy consumption in -21°C compared to 22°C). Therefore, our results might be slightly more on the conservative side. However, the data on Nissan Leaf investigates hot weather as well as cold climate, therefore we found it more reliable to use for our purposes.

According to Eq. (2.1) the minimum energy consumption occurs at 22°C, corresponding to 0.27 kWh/mile. We used this as our base case and normalized the energy consumption at other temperatures for other vehicles using Eq. (2.2). In addition, we assumed that the additional energy consumption, compared to the base case, arises from the using heating/cooling equipment and battery efficiency drop in cold temperatures.

$$h_{VEH}(T) = \frac{h_{LEAF}(T)}{h_{LEAF}(22^{\circ}C)} \cdot h_{VEH}(22^{\circ}C) \quad (2.2)$$

where $h_{VEH}(T)$ is the normalized energy consumption of the commercial vehicle under consideration at temperature T and $h_{VEH}(22^{\circ}C)$ is the actual energy consumption of the commercial vehicle at 22°C (or without temperature effects).

Assuming that one unit of energy is consumed to travel one unit of distance in the base case, i.e. $h_{VEH}(22^{\circ}C) = 1$, the energy consumption at other temperatures are shown in Table 2.1. Temperature effect on energy consumption at 8°C has the similar impact as at 27°C. Same phenomenon can be observed for other cold/hot temperature pairs as presented in Table 2.1.

Table 2.1 Energy consumption at different temperatures

Temperature (°C)	Condition	Energy Consumption (per unit distance)
22	Mild	1.00
8 or 27	Intermediate	1.09
0 or 33	Intense	1.27
-21 or 38	Extreme	1.52

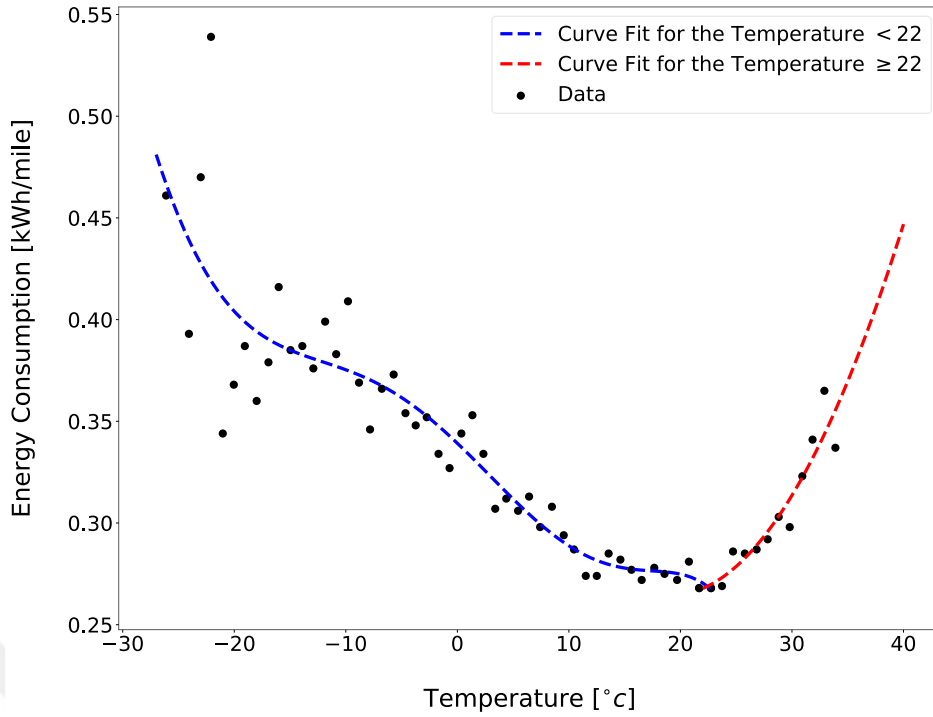


Figure 2.1 Energy consumption vs. ambient temperature for Nissan Leaf

In Figure 2.2, we present a simple example that illustrates how temperature affects the optimal route plans and how it can increase the energy consumption or cause infeasibility. The example involves one depot equipped with a charger, two customers and one station. All customers must be served, and the vehicle tours should start from and terminate at the depot. For the sake of simplicity, we do not consider the cargo capacity and time-window constraints. The battery capacity of the EV is three units. The distances are symmetric and the numbers on the arcs represent the energy consumptions. The directed arcs with solid line show the optimal routes. Figure 2.2 (a) demonstrates the network and optimal route in a mild temperature (22°C). We assume that the vehicle travels at constant speed and as there is no need for cooling/heating in this temperature, it consumes one unit of energy per unit distance traveled. So, the numbers on the arcs in this network also show the distances. In the optimal solution, one EV serves both customer by cruising a total distance of 3 units and consuming 3 units of energy. In Figure 2.2 (b)-(d), the temperature is colder, so heating equipment causes more energy consumption in comparison with the mild case. Figure 2.2 (b) shows the optimal route plan for the intermediate case when the temperature is 8°C. The consumption rate per unit distance is 1.09 in this case and one EV serves both customers by traveling a total distance of 3.9 units and consuming 4.25 units of energy. Note that the EV needs a recharging at the station in order to continue its tour. In Figure 2.2 (c) as the ambient temperature effect is more intense (the consumption rate

is 1.27), each customer is served by a dedicated EV and the total energy consumption is 5.08. Figure 2.2 (d) depicts the case where the weather is too cold (-21°C) and the temperature effect is extreme (the consumption rate is 1.52). We see that customer 1 can still be served by a dedicated EV; however, no EV can visit customer 2 and reach the station or return to the depot without running out of battery. So, the problem becomes infeasible.

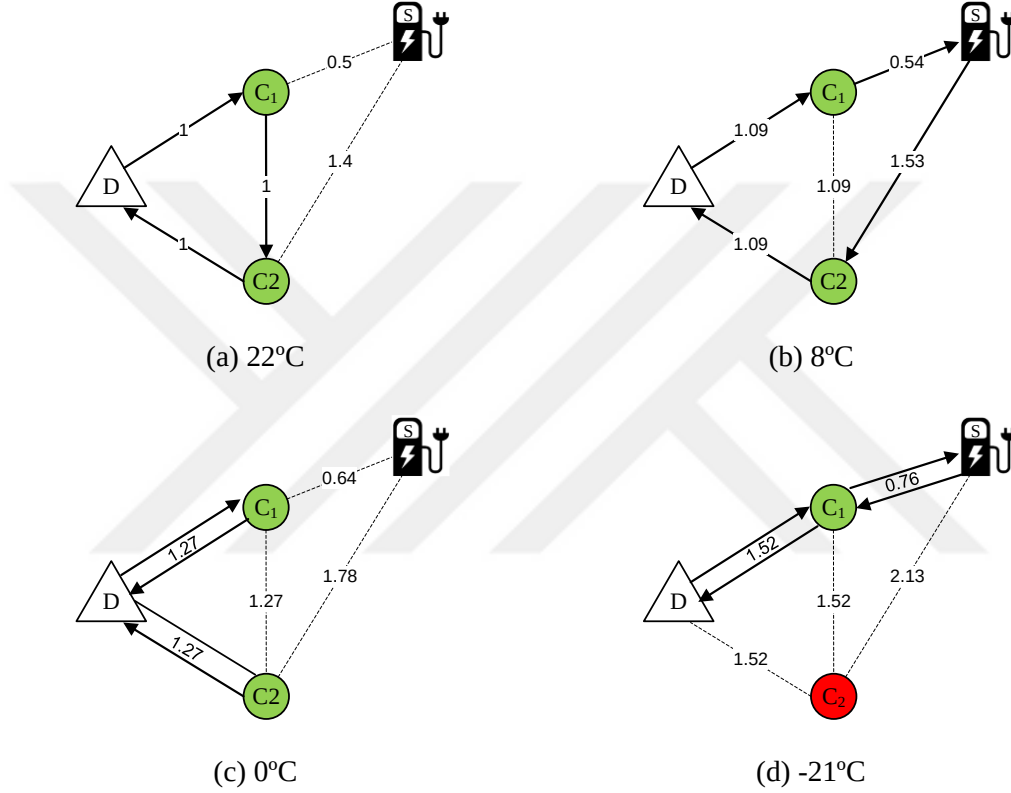


Figure 2.2 Optimal route plans that change according to varying temperatures

2.2.2. Mathematical formulation

Similar to the notation and modelling conventions in Keskin and Çatay (2016) and Bruglieri et al. (2016) we define $V = \{1, \dots, n\}$ as the set of customers and F as the set of recharging stations. Vertices 0 and $n + 1$ denote the depot where each vehicle departs from 0 (departure depot) and returns to $n + 1$ (arrival depot) at the end of its tour. We define $V_0 = V \cup \{0\}$, $V_{n+1} = V \cup \{n + 1\}$ and $V_{0,n+1} = V \cup \{0, n + 1\}$. Then, the problem

can be represented on a complete directed graph $G = (N, A)$ with the set of arcs $A = \{(i, j) | i, j \in N, i \neq j\}$, where $N = V_{0,n+1} \cup F$ is the total set of nodes on the network. The energy consumption depends on the distance traveled and the duration of the trip. Each customer $i \in V$ has a positive demand q_i , service time s_i , and time window $[e_i, l_i]$. All EVs have a cargo capacity of C and a battery capacity of Q . At each recharging station, one unit of energy is transferred in g time units. The direct distance from customer i to customer j is represented by d_{ij} whereas the vehicle travels the additional distance of $\hat{d}_{ijs} = d_{is} + d_{sj} - d_{ij}$ if it is recharged at station s en-route. Notice that the battery can be recharged at most once between two consecutive customers, which is not an unrealistic assumption within the context of city logistics (Keskin and Çatay, 2018).

Similarly, t_{ij} denotes the travel time from customer i to customer j if the journey is direct and $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$ is the additional travel time if it is via station s . Note that $\hat{t}_{ijs}$ does not include the recharging time at station s . The energy consumed for moving the vehicle one-unit distance is represented by h^d whereas h^t denotes the energy consumed by the cabin heating or cooling system per unit time. At cold temperatures, h^t also includes the extra energy consumed per unit time due to battery efficiency drop. The total energy consumption is a linear function of the distance and duration of the journey from customer i to customer j and is calculated as $h_{ij} = h^d d_{ij} + h^t t_{ij}$ when the journey is direct. If the battery is recharged at station s en-route, the additional energy consumption is calculated as $\hat{h}_{ijs} = h_{is} + h_{sj} - h_{ij}$.

Table 2.2 Mathematical notation for EVRPTW with ambient temperature

Sets:	
V	Set of customers
V_0	Set of customers and departure depot
V_{n+1}	Set of customers and arrival depot
$V_{0,n+1}$	Set of customers, departure, and arrival depots
F	Set of recharging stations
N	Set of customers, stations, and depots
K	Set of vehicles
Parameters:	
d_{ij}	Distance between node i and j
$\hat{d}_{ijs}$	Additional distance of visiting station s between customers i and j , $\hat{d}_{ijs} = d_{is} + d_{sj} - d_{ij}$
t_{ij}	Travel time from node i and j
$\hat{t}_{ijs}$	Additional trip time of visiting station s between customers i and j , $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$

q_i	Demand of customer i
r_i	Service time of customer i
$[e_i, l_i]$	Time window of customer i
C	Freight capacity
Q	Battery capacity
g	Recharging rate
h_{ij}	Total energy consumed to traverse arc (i, j)
$\hat{h}_{ijs}$	Additional consumption if the vehicle is recharged in station s while traveling from customer i to customer j , $\hat{h}_{ijs} = h_{is} + h_{sj} - h_{ij}$

Decision variables:

τ_i	Service starting time at customer i
y_i^k	battery SoC of vehicle k upon arrival at (departure from) customer/depot $i \in V_{0,n+1}$
y_{ijs}^k	battery SoC of vehicle k upon arrival at station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$
Y_{ijs}^k	battery SoC of vehicle k at departure from station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$
x_{ij}^k	1 if vehicle k travels from node $i \in V_0$ to node $j \in V_{n+1}$; 0 otherwise
z_{ijs}^k	1 if vehicle k traverses arc (i, j) , $i \in V_0, j \in V_{n+1}$, through station $s \in F$; 0 otherwise

The decision variables y_i^k, y_{ijs}^k , and Y_{ijs}^k , keep track of battery SoC of vehicle k at arrival at customer/depot $i \in V_{0,n+1}$, at arrival at station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$, and at departure from station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$, respectively. Service starting time at any node $i \in N$ is denoted by τ_i . The binary decision variable x_{ij}^k takes value 1 if vehicle k travels from node $i \in V_0$ to node $j \in V_{n+1}$ and 0 otherwise. The binary decision variable z_{ijs}^k takes value 1 if vehicle k traverses arc (i, j) , $i \in V_0, j \in V_{n+1}$, through station $s \in F$. The mathematical notation is summarized in Table 2.2

Reference source not found..

The mixed-integer programming model of the problem can be formulated as follows:

$$\text{Min } \sum_{i \in V_0} \sum_{j \in V_{n+1}} \sum_{k \in K} (h_{ij} x_{ij}^k + \sum_{s \in F} \hat{h}_{ijs} z_{ijs}^k) \quad (2.3)$$

subject to

$$y_0^k = Q \quad \forall k \in K \quad (2.4)$$

$$\sum_{\substack{j \in V_{n+1} \\ j \neq i}} \sum_{k \in K} x_{ij}^k = 1 \quad \forall i \in V \quad (2.5)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij}^k - \sum_{\substack{i \in V_{n+1} \\ i \neq j}} x_{ij}^k = 0 \quad \forall j \in V, k \in K \quad (2.6)$$

$$\sum_{s \in F} z_{ijs}^k \leq x_{ij}^k \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (2.7)$$

$$\tau_i + (t_{ij} + r_i)x_{ij}^k + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs}^k + g(Y_{ijs}^k - y_{ijs}^k)) - l_0(1 - x_{ij}^k) \leq \tau_j \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (2.8)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (2.9)$$

$$\sum_{i \in V} \sum_{\substack{j \in V_{n+1} \\ j \neq i}} q_i x_{ij}^k \leq C \quad \forall k \in K \quad (2.10)$$

$$0 \leq y_j^k \leq y_i^k - h_{ij}x_{ij}^k + Q(1 - x_{ij}^k + \sum_{s \in F} z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (2.11)$$

$$y_j^k \leq \sum_{s \in F} (Y_{ijs}^k - h_{sj}z_{ijs}^k) + Q(1 - \sum_{s \in F} z_{ijs}^k) + Q(1 - x_{ij}^k) \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (2.12)$$

$$0 \leq y_{ijs}^k \leq y_i^k - h_{is}z_{ijs}^k + Q(1 - x_{ij}^k) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (2.13)$$

$$y_{ijs}^k \leq Y_{ijs}^k \leq Qz_{ijs}^k \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (2.14)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (2.15)$$

$$z_{ijs}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (2.16)$$

The objective function (2.3) minimizes the total energy consumption. Constraints (2.4) set the battery SoC of the EVs to full when they depart from the depot. The connectivity of customer visits is enforced by constraints (2.5) whereas the flow conservation at each vertex is ensured by constraints (2.6). Constraints (2.7) make sure that vehicle k serves customer j after customer i if it travels from i to j by recharging its battery en-route. Constraints (2.8) guarantee the time feasibility of arcs emanating from the customers (and the depot). Constraints (2.9) establish the service time windows restriction. Constraints (2.8) and (2.9) also eliminate the formation of sub-tours. Constraints (2.10) impose the cargo capacities of the vehicles. Constraints (2.11)-(2.14) keep track of the battery SoC at each node and make sure that it never falls below zero. Constraints (2.11) establish the battery SoC consistency if the vehicle travels from customer i to customer j without recharging en-route. Constraints (2.12) determine battery SoC at the arrival at customer j if the vehicle visits a recharging station after it has departed from customer i whereas constraints (2.13) check battery SoC at the arrival at a station if the battery is recharged

en-route. Constraints (2.14) set the limits for battery SoC when the vehicle departs from a station. Finally, constraints (2.15)-(2.16) define the binary decision variables.

2.3. Solution methodology

Small-size problems can be solved on a commercial solver using the above mathematical formulation. For large-size instances that are not tractable, we resort to ALNS. ALNS is a metaheuristic method introduced by R pke and Pisinger (2006a, 2006b) and has been employed for solving various VRPs including VRPTW variants (Pisinger and Ropke, 2007; Ribeiro and Laporte, 2012; Demir et al., 2012; Aksen et al., 2014; Grangier et al., 2016; Eme  et al., 2016; Ko  et al., 2016). It has also been successfully applied to EVRPTW and its extensions (Goeke and Schneider, 2015; Hiermann et al., 2016; Keskin and  atay, 2016; Wen et al., 2016; Schiffer and Walther, 2017; Schiffer et al., 2018; Keskin and  atay, 2018; Keskin et al., 2019).

ALNS is a neighborhood search technique that consists of a destroy-and-repair framework where at each iteration a destroy operator is used to remove some nodes from the current solution and a repair operator is applied to insert the removed nodes to improve the incumbent solution. The insertion and removal mechanisms are associated with a numerical score which is updated after each iteration based on their performances. If a mechanism yields to a good solution, its corresponding score and consequently the probability of selecting that mechanism in the subsequent iterations increase.

In this chapter, we employ the ALNS algorithm presented in Keskin and  atay (2016). Since the graph may become incomplete due to the increased energy consumption on arcs in low/high temperatures, the algorithm may struggle to find a feasible solution in certain cases. To overcome this problem, we introduce a new station insertion mechanism utilized both for constructing the initial solution and improving it within the ALNS framework. The insertion algorithms in the ALNS approach of Keskin and  atay (2016) are designed to insert a station in one of the preceding arcs when the insertion of a customer leads to a negative battery SoC at the arrival at that customer. However, adding only one station to the route may not be sufficient in our case since the energy consumption can significantly increase due to the ambient temperature. So, we propose a new station insertion

mechanism in which multiple stations can be inserted to the route simultaneously. We refer to this algorithm as Multi-Station Insertion (MSI) and describe it as follows:

Multi-Station Insertion (MSI): When the insertion of a customer yields an infeasible partial route with respect to the battery SoC, we insert a station on the arc traversed immediately before arriving at the customer with negative SoC and recharge the battery to the maximum level allowed by the battery capacity and time windows restrictions of the succeeding customers. If the SoC is still negative at that customer or if the energy on the battery is not sufficient to reach the inserted station, we attempt inserting another station prior to the customer visited before traveling to the recently inserted station. This procedure is repeated until the partial route becomes energy feasible.

The interested reader is referred to Keskin and Çatay (2016) about the details of the ALNS implementation, destroy and insertion mechanisms, and parameters utilized.

2.4. Computational study

We use the well-known data set of Schneider et al. (2014) to analyze the effect of the ambient temperature on the fleet size, energy consumption, and route plans. The data consists of three problem types where the customers are clustered (c-type), randomly distributed (r-type), and both clustered and randomly distributed (rc-type). It is also classified in two types, which differ by the length of the time windows, scheduling horizon, and vehicle cargo and battery capacities. In subsets r1, c1, and rc1, the time windows are narrow and the scheduling horizon is short whereas the time windows are wide and the scheduling horizon is longer in subsets r2, c2, and rc2. Furthermore, each subset assumes an EV fleet with different cargo and battery capacity. For the sake of simplicity, the consumption rate is assumed to be one unit of energy per unit distance traveled. We use this rate for mild temperature condition and consider the rates given in Table 2.1 for other cases.

Cities around the world may experience extremely low or high temperatures, which substantially affects the energy consumption of the EVs. For instance, the daytime

temperature dropped below -20°C in Saskatoon, Saskatchewan in 10 days and below -15°C in 26 days during 2017. In addition, Montreal, Quebec experienced below -15°C in 8 days and below -10°C in 22 days. On the other hand, the temperature was above 40°C in 75 days in Las Vegas and 97 days in Phoenix. Moreover, Rome observed above 35°C in 32 days during the same year. So, in our experiments we consider temperatures between -21°C and 38°C , which are not unusual to observe for many cities around the world (accuweather, 2018).

To investigate the influence of ambient temperature on energy consumption and routing decisions, we solve 36 small-size and 29 large-size benchmark instances. We use IBM ILOG CPLEX 12.6.3 optimization solver for the small instances and ALNS algorithm to solve larger instances on a workstation equipped with Intel(R) Core(TM) i7-8700 processor with 3.20 GHz speed and 16 GB RAM. We limit the CPU run time with 2 hours. The detailed numerical results are presented in Appendix A.

Table 2.3 Number of infeasible problems in small-size dataset for different temperature conditions

#Cust	#Inst	Mild	Intermediate	Intense	Extreme
5	12	0	0	3	4
10	12	0	2	7	9
15	12	0	0	5	8
Total	36	0	2	15	21

2.4.1. The influence of ambient temperature on routing decisions in small-size instances

Different temperature cases are investigated on the three subsets of 12 small-size instances. Each subset involves 5, 10, and 15 customers and different number of stations varying between two and eight. We solved each instance for four different temperature conditions. So, the total number of problems solved is $3 \times 12 \times 4 = 144$. The detailed results are provided in Appendix A. In Table 2.3, we report the number of infeasible problems. In this table, ‘#Cust’ indicates the number of customers in the data set and ‘#Inst’ is the number of instances. Out of 144 problems, 106 are feasible. Since all instances are feasible in mild temperature, we can say that weather conditions make 35%

of the problems (38 out of 108) infeasible. Among 106 feasible problems, CPLEX solved 101 problems optimally and provided an upper-bound for the remaining five within two hours. As expected, we observe that infeasibility increases as the temperature conditions change from mild to extreme. While only 6% of the problems are infeasible in the intermediate case, 42% and 58% of the problems become infeasible when the temperature conditions are intense and extreme, respectively.

Table 2.4 The influence of ambient temperature on route plans in small-size instances

Ambient Temperature	# of Instances		Average Increase	
	Feasible	Larger Fleet	Δ #Veh	Δ EC
Intermediate	34 / 36	3 / 34	4%	12%
Intense	21 / 36	6 / 21	15%	40%
Extreme	15 / 36	10 / 15	46%	81%

Table 2.4 summarizes how different temperature levels affect the solutions in small-size instances. In this table, column ‘Feasible’ reports the number of feasible solutions for different temperatures whereas ‘Larger Fleet’ column shows the number of instances in which more EVs are needed compared to the fleet size in the mild condition. ‘ Δ #Veh’ and ‘ Δ EC’ columns give the average percentage increase in the fleet size and energy consumption, respectively, again compared to the mild case. The number of infeasible instances increases as the temperature conditions change from mild to extreme, as expected. In the intense temperature case, 15 of the 36 instances become infeasible and six instances among the feasible ones require one extra vehicle each in comparison to the mild case. The most critical case happens when the temperature effect is extreme. In this case, 21 instances turn out to be infeasible and 10 instances among the feasible ones need more vehicles to satisfy the customer demands on time (two out of these 10 instances need two additional vehicles whereas remaining eight instances require one additional vehicle each, compared to the base case). The fleet size grows by 4%, 15% and 46% in the intermediate, intense and extreme temperature cases compared to the mild case. Noting that our primary objective is to minimize the number of vehicles, we also observe significant increase in energy consumption at low temperatures. In the intense temperature case, the average energy consumption increases by 40% compared to mild temperature whereas this increase almost doubles and reaches 81% in the extreme conditions. All these results show the crucial effect of weather conditions on total energy consumption when the logistics operations are performed using an EV fleet.

It is important to note that compared to the mild case even though Table 2.1 reports 9%, 27%, and 52% increases in the energy consumption per unit distance for the intermediate, intense, and extreme temperature cases, respectively, the average energy consumption in the route plans shows an increase of 12%, 40%, and 81%. This substantial difference is the result of visiting more stations en-route and making longer detours.

2.4.2. The influence of ambient temperature on routing decisions in large-size instances

The large-size data consists of 29 instances with 100 customers and 21 stations. We focus on type-1 problems with narrow time windows since wide time windows have minor influence on the recharging decisions (Desaulniers et al., 2016; Keskin and Çatay, 2018). We solved each instance 10 times using ALNS under four different temperature conditions. Since a large set of recharging stations is available, all problems are feasible in intermediate and intense conditions. However, ALNS faced difficulty in finding a feasible solution, particularly for r- and rc-type problems. So, we performed 100 runs in these problem sets. Yet, ALNS still failed to solve six rc-type problems in the extreme case. Even though we cannot prove it, it is highly likely that these problems are infeasible.

Table 2.5 Average results for large-size problems

Type	Mild		Intermediate		Intense		Extreme	
	#Veh	EC	#Veh	EC	#Veh	EC	#Veh	EC
c	10.6	1003.05	10.8	1103.33	11.1	1352.62	11.8	1608.89
r	13.0	1240.03	13.3	1365.47	14.1	1656.17	15.2	2117.24
rc	12.9	1409.54	13.3	1572.00	14.3	1929.25	—	—

The results are summarized in Table 2.5 and detailed results are given in Appendix A. In Table 2.5, ‘#Veh’ shows the average number of vehicles employed whereas ‘EC’ reports the average energy consumption in the problems solved. Results for rc-type problems are not reported because of the aforementioned reason. Furthermore, we observe that the fleet size and energy consumption increase as the temperature drops, as expected. Since the customers are clustered, type-c instances are affected less from the ambient temperature compared to r- and rc-type instances. In the extreme case the average fleet size and

average energy consumption in type-c instances increase by 11% and 60%, respectively, as compared to the mild case, whereas the increases in type-r instances are 17% and 71%, respectively.

Table 2.6 The influence of ambient temperature on route plans in large-size instances

Ambient Temperature	# of Instances		Average Increase	
	Feasible	Larger Fleet	Δ #Veh	Δ EC
Intermediate	29 / 29	9 / 29	3%	11%
Intense	29 / 29	24 / 29	8%	35%
Extreme	23 / 29	22 / 23	15%	68%

Table 2.6 shows the effect of the ambient temperature on route plans in large-size instances. The results are similar to those observed for small-size problems, except the feasibility issue. While 93% of the large-size problems (81 out of 87) were solved feasibly under rougher temperature conditions, this percentage drops to 65% for the small-size data set. This is due to the scarcity of the recharging stations in the small-size data set. In addition, the need for a larger fleet is observed in more problems in the large-size data set: 68% of the problems compared to 27% in the small-size data set. Specifically, in almost all large-size problems (22 out of 23) more EVs are needed in extreme temperature compared to the mild case. On the other hand, the average percentage increase in the fleet size is significantly smaller. For example, for the extreme case this value is 15% for large-size data set compared to 46% for the small-size. The reason behind this is the actual size of the fleet: for the mild case, the number of EVs in the fleet is between one and five in small-size data whereas in large-size data the fleet size varies from 10 to 18. So, two additional EVs in the intense case imply a more significant percentage change in a small-size problem compared to the large-size.

When we consider the average energy consumption in the route plans, we observe an increase of 11%, 35%, and 68% for the intermediate, intense, and extreme cases, respectively, in comparison to the base case of mild temperature. Notice that these percentages are slightly smaller than those reported in Table 2.4. This may be due to the availability of more stations in large-size data, hence, shorter detours for recharging. Nevertheless, these values still reveal a higher consumption rate than those in Table 2.1.

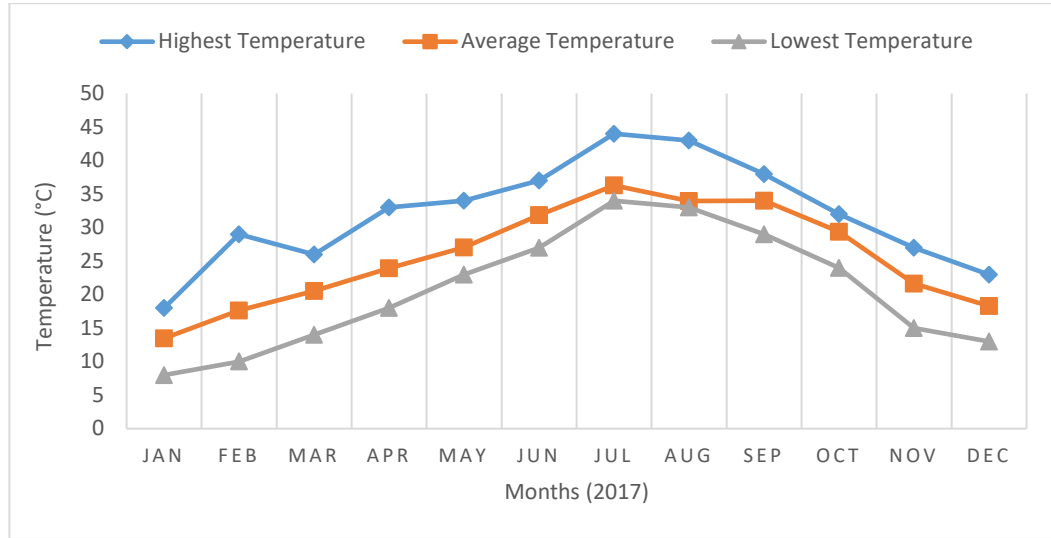


Figure 2.3 Monthly daytime highest/average/lowest temperatures in Adana during 2017

2.5. Case study

In this section, we consider the last-mile distribution planning of Ekol Logistics, a third-party logistics service provider in Turkey. To show the effect of temperature on real world fleet operations, we solve their routing problem in Adana. Adana is a city located in Southern Turkey with a year-round mild-to-hot and mostly humid climate. The highest, average, and lowest daytime temperatures in 2017 are illustrated in Figure 2.3. January was the coldest month in which the lowest, average, and highest daytime temperatures were 8°C, 13°C, and 18°C, respectively; however, July was the warmest, where the highest daytime temperature reached as high as 44°C (Accuweather.com, 2018; Weather.com, 2018). Using cooling equipment is necessary in Adana for several months and it affects the energy consumption, which makes it worthwhile to analyze the optimal route plans for an EV fleet under those conditions.

Table 2.7 Case study data: time windows and demands of the customers

	Customers											
	8	9	10	11	12	13	14	15	16	17	18	19
Early service time	8:30	8:30	10:00	8:30	8:30	8:30	8:30	8:30	13:00	8:30	10:00	9:30
Late service time	17:30	17:30	12:00	17:30	17:30	10:00	17:30	17:30	15:00	12:00	16:00	12:00
Service time (min)	9	5	4	15	4	90	4	7	53	10	12	10
Demand (kg)	50	6	1.5	226.7	2.7	458.6	2.7	17.5	277	71	170.7	65

Ekol uses diesel vehicle fleet of light commercial Fiat Ducato vans. We investigate how the routes would change if they used electric Fiat Ducato vans (eDucato) in their fleet. So, we consider Ducato EVs that have a cyclable battery capacity of 62 kWh and can carry loads up to 718.4 kg. eDucato has an advertised driving range of 200 km, which corresponds to an energy consumption of 0.31 kWh/km (BD Automotive, 2018). Neglecting other factors, we assume that in the mild case the EVs consume $h_{\text{VEH}}(22^{\circ}\text{C}) = 0.31 \text{ kWh/km}$, i.e. we assume that the advertised energy consumption corresponds to consumption at 22°C . Using the relation in Eq. (2.2) in section 2.2.1, the energy consumption for the intermediate (27°C), intense (33°C), and extreme (38°C) cases are obtained as 0.34, 0.4, and 0.47 kWh/km, respectively. To solve the problem with CPLEX, we focused on a smaller subset of 12 selected customers and we assumed that EVs can recharge at depot as well as at public charging stations in that region. These stations have level 2 AC chargers with 22 kVA power as well as specific time windows for providing service (Eşarj, 2018). The depot operates between 5:00 and 17:30 and is represented with index '1'. Indices '2'-'7' refer to the stations and '8'-'19' denote the customers. Table 2.7 shows the time-windows, service times, and demands of the customers. The distance matrix is provided in Appendix C.

Figure 2.4 illustrates the geographical locations of the depot, customers, and stations. The black circle shows the depot, the customers and stations are represented with green pins and red chargers, respectively. The city center is magnified for a clearer view of the region. The optimal route plans for different cases are plotted in Figure 2.5. In this figure, the depot, customers, and stations are displayed with a triangle, circles, and station icons, respectively. Station 2 represents the charger located in the depot. Battery SoCs are given next to the nodes in kWh. The two numbers provided next to the stations show the arrival and departure SoC values; the difference gives the amount of energy charged at the station. The SoC does not change at the customers. We assume that all EVs are fully charged at the depot overnight.

In Figure 2.5(a), we see that, all customers can be served using two vehicles in the mild case when the energy consumption is lowest. EV1 visits Station 2 (in the depot) once to recharge 8.3 kWh of energy. Total energy consumption in this case is 121.6 kWh. In the intermediate case illustrated in Figure 2.5(b), all customers are also served with two vehicles;



Figure 2.4 Geographical area

however, EV2 visits Station 7 and Station 2 to recharge 45.9 and 7.5 kWh, respectively. The total energy consumption is 168.5 kWh, which corresponds to an increase of 39% compared to the mild case. In addition, the sequence of the visits to the customers changes since charging at stations is a time-consuming process and can affect the routing decisions due to customers' time-windows. In the intense case shown in Figure 2.5(c), the effect of ambient temperature on energy consumption is even stronger and it is not possible to make all deliveries using two vehicles. Therefore, the fleet consists of three EVs and EV1 and EV2 visits Stations 7 and 4, respectively, to recharge en-route. The total energy consumption becomes 186.7 kWh, an increase by 54% compared to the mild case. Notice this dramatic increase in the real environment compared to the average increase of 40% in the synthetic benchmark data reported in Table 2.4.

The route plans in the extreme case are not illustrated because no feasible solution exists. When the ambient temperature is 38°C the EV energy consumption increases to 0.47 kWh/km. In this case, the EV cannot serve customer 15 because the trip from depot to that customer requires 45.62 kWh of energy and the EV will run out of battery en-route if it returns to depot directly or travels to the nearest station (Station 7) which requires an additional 19.50 kWh. Hence the problem is infeasible. All route plans are given in Appendix B.

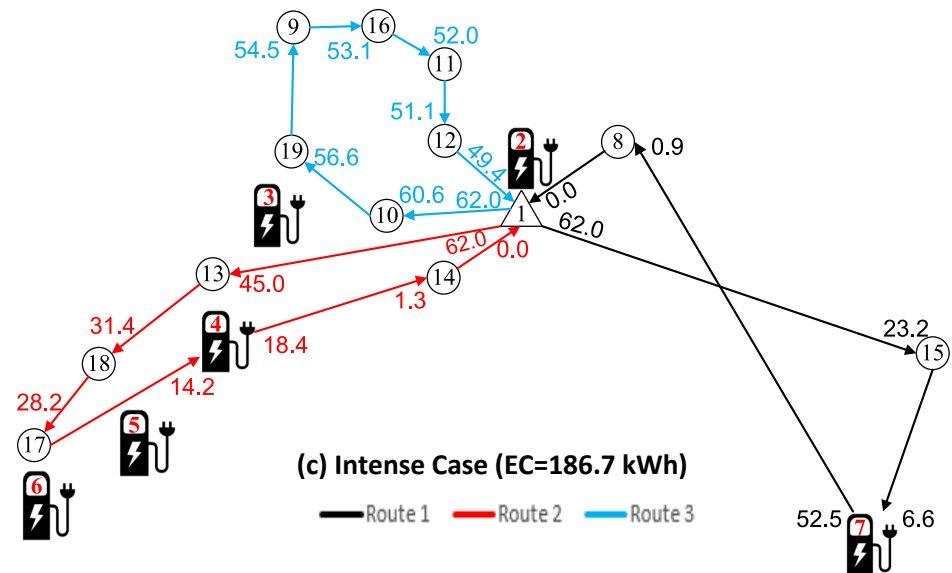
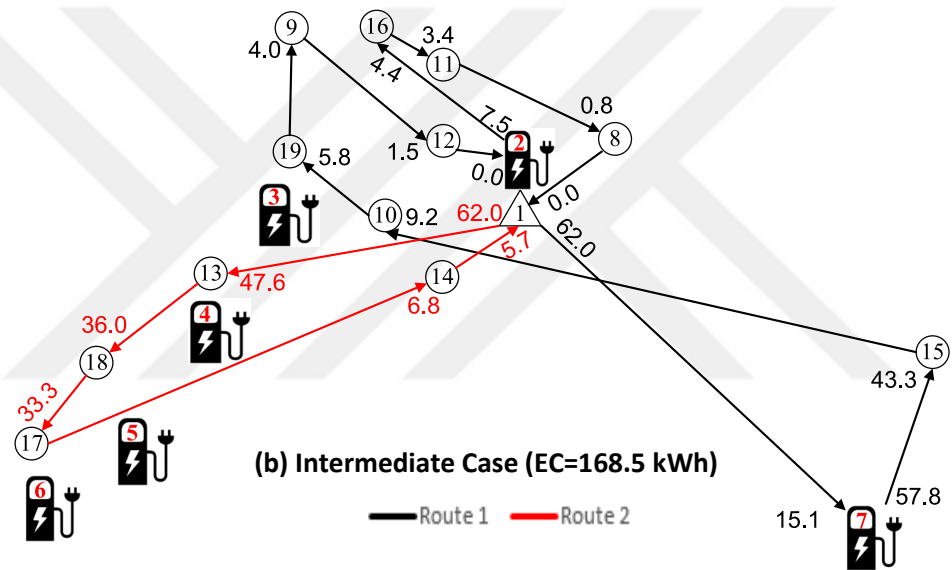
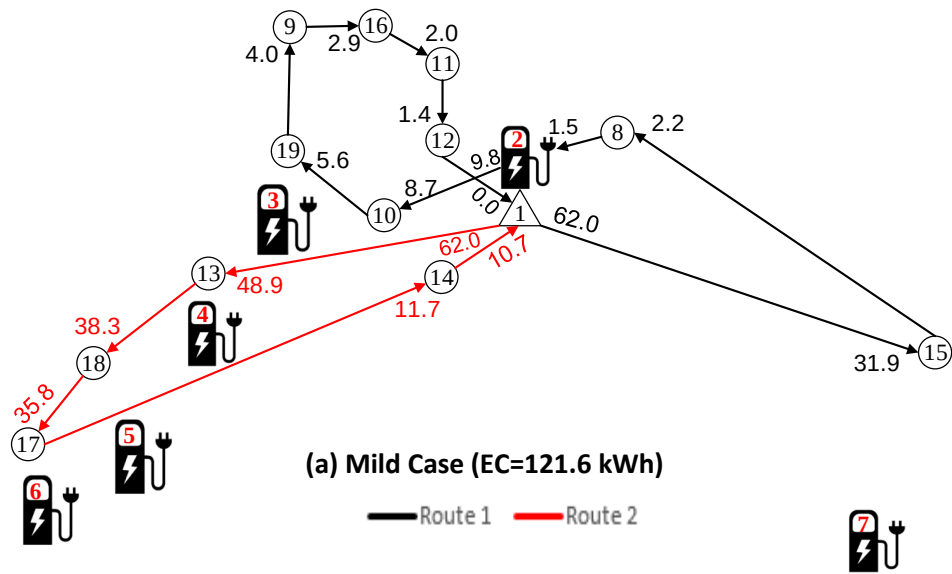


Figure 2.5 Optimal route plans at different ambient temperature conditions

As this case study demonstrates, ambient temperature can have a strong effect on the delivery operations. According to the meteorological records from Adana in the year 2017, the maximum temperature measured was higher than 27, 33 and 38°C for 177, 98 and 3 days, respectively (accuweather, 2018). Also, as can be depicted from Figure 2.3 it is possible to observe temperatures as high as 44°C. Routing decisions made without considering these high temperature values can cause inefficient operations or disruptions.

2.6. Discussion and conclusion

In this chapter, we focused on EVRPTW with partial recharge by considering the effect of ambient temperature on the fleet size and energy consumption. We introduced the mathematical model of the problem and used it to solve small-size problems on CPLEX. We solved large-size problems with an ALNS algorithm. We also performed a case study using real-world data from a logistics service provider in Adana, Turkey where we optimized the delivery routes for a network of 12 customers at four different days with different ambient temperature values. Our results showed that temperature can have a significant effect on the delivery operations since as temperature increases: (i) the total energy consumption of the operations increases, (ii) the route, therefore the sequence of customers served changes, (iii) the number of vehicles required to complete the service might increase, and (iv) it might become impossible to serve all customers using an EV fleet. Routing plans made without considering the temperature effect might not be cost-optimal or might even become infeasible at hotter days. Similar results might also be observed at colder temperatures, since vehicle energy consumption will also increase with a decrease in temperature as shown in Figure 2.1. This issue is particularly important in regions with extreme weather conditions as well as in temperature-sensitive operations such as cold logistics chains. Therefore, planning ahead for expected weather conditions can reduce the efficiency losses and prevent operation failures. In this chapter, we provide a framework that enables to perform such routing plans in last/first mile logistics activities.

The results of this study can have various implications for real world managerial decisions in logistics operations. To address the aforementioned issues, logistics service providers

might consider running a mixed fleet with different powertrain options (EVs, ICEVs, hybrid and plug-in hybrid EVs). Increasing the number of available charging stations and/or having a fleet of EVs with fast-charging capabilities might reduce the impact of temperature. However, frequent and fast charging can have battery life implications and this trade-off has to be considered when planning. Similarly, vehicles with longer driving ranges would reduce the impact, but this would require having bigger batteries and therefore reducing the load capacity; another trade-off that needs to be assessed.

There are some limitations and assumptions in this chapter. We used data collected from Nissan Leaf (a light duty passenger vehicle) users and generalize the temperature dependency relation to be used for a medium-duty EV. In reality, the commercial vehicle might differ in heating, ventilation, air conditioning (HVAC) efficiency, and battery technology, and might therefore have a different actual consumption. In addition, the data was collected from several Nissan Leaf drivers and the reported results reflect the average of their behavior. Individual drivers might differ in terms of driving style and speed, as well as in climate control preferences, therefore might experience different results. Furthermore, in our case study we only consider recorded temperature levels, however, in Adana humidity can reach 70% in summer months (Weather.com, 2018) and the feels-like temperature was recorded as high as 47°C in 2017 (Daily Sabah, 2017). This would cause an increase on the HVAC load and more drastic effects on routing plans might be observed. Nevertheless, our results show the general trends fairly, since HVAC use will increase energy consumption for all EVs, even though its degree may change from vehicle to vehicle. This chapter is the first to investigate these effects within the context of vehicle routing and scheduling.

3. SPEED-UP TECHNIQUES FOR SOLVING THE ELECTRIC VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

3.1. Introduction to Electric Vehicle Routing Problem with Time Windows

Electric vehicles (EVs) are becoming more popular in logistics and transportation operations throughout the world due to various factors including government incentives, fluctuations in oil prices, energy efficiency of electric engines compared to internal combustion engines and low electricity prices. They also help companies reduce their carbon footprint and meet their sustainability objectives. On the other hand, limited driving range and long recharging times of EVs raise additional challenges in route planning. A trip from a customer to another may not be realizable because of long travel distance and/or customer service time windows. In certain cases, it may not even be possible to serve some customers using an EV. While it may be difficult to trace the infeasibility of a given instance, the elimination of infeasible arcs may speed up the solution algorithm. In this chapter, we consider the Electric Vehicle Routing Problem with Time Windows (EVRPTW) and propose new preprocessing techniques to reduce the problem size and enhance the computational performance of the solution methods. Furthermore, we develop an algorithm which can be used to identify if a problem instance is infeasible.

EVRPTW is an extension of the Vehicle Routing Problem with Time Windows (VRPTW) where an EV fleet serves the customers instead of internal combustion engine vehicles. The first study that considers an EV fleet in freight transportation was conducted by Conrad and Figliozzi (2011) where EVs can recharge at selected customer locations. Erdoğan and Miller-Hooks (2012) introduced the Green Vehicle Routing Problem

(GVRP) where the fleet consists of alternative fuel vehicles (AFVs). In this problem, AFVs are allowed to refuel at public alternative fueling stations (AFSs). The refueling time is assumed constant and the tank is full at departure from the station. Wang and Cheu (2013) investigated a similar problem for a fleet of electric taxis by considering a similar full recharge strategy.

Schneider *et al.* (2014) introduced EVRPTW by assuming a full recharge strategy as well. They formulated the mathematical programming model of the problem and developed a hybrid metaheuristic approach that combines Variable Neighborhood Search (VNS) and TS algorithms. They also generated a new data set based on the well-known Solomon (1987) dataset for VRPTW. The full recharge assumption was relaxed by Bruglieri *et al.* (2015) and Keskin and Çatay (2016). The former proposed a VNS Branching method to solve small-size problems whereas the latter developed an Adaptive Large Neighborhood Search (ALNS) algorithm that efficiently solves large-size problems. Desaulniers *et al.* (2016) considered four recharging cases in EVRPTW, namely full and partial strategies allowing single and multiple recharges en-route, and proposed a branch-price-and-cut algorithm.

In the literature, several variants of EVRP and EVRPTW were addressed including the cases of operating a mixed fleet of EVs and internal combustion engine vehicles (Goeke and Schneider, 2015; Sassi *et al.*, 2015, Macrina *et al.*, 2018, Hiermann *et al.*, 2019), heterogeneous fleet of EVs (Hiermann *et al.*, 2016), fast charging (Felipe *et al.*, 2014; Çatay and Keskin, 2017; Keskin and Çatay, 2018), non-linear charging function (Montoya *et al.*, 2017; Froger *et al.*, 2019), battery swapping (Yang and Sun, 2015; Hof *et al.*, 2017; Paz *et al.*, 2018), location routing (Worley *et al.*, 2012; Hof *et al.*, 2017; Schiffer and Walther, 2017), two-echelons (Jie *et al.*, 2019), flexible time windows (Taş, 2020), and two-stages (Basso *et al.*, 2019). Some recent studies addressed the availability of recharging stations and queueing for recharging service (Froger *et al.*, 2017; Keskin *et al.*, 2019). A comprehensive review of the EV technology and survey of the EVRP variants may be found in Pelletier *et al.* (2016), Pelletier *et al.* (2017), and Erdelić and Carić (2019).

In this chapter, we revisit the EVRPTW by allowing partial recharges and present some reduction techniques. Our contributions to the literature are threefold: (i) we develop a

preprocessing procedure to reduce the graph, hence the number of decision variables in the problem; (ii) we propose an algorithm to identify whether a problem instance is feasible or not; (iii) we perform extensive computational tests to investigate the performance of the proposed approaches. The remainder of this chapter is organized as follows: the problem description and mathematical model are presented in Section 3.2. Section 3.3 introduces the network reduction procedure by describing the proposed preprocessing techniques and valid inequalities. Section 3.4 presents an algorithm that effectively checks the feasibility of a problem instance. Section 3.5 investigates the impact of the proposed preprocessing techniques on the computational performance. Final remarks and future research directions conclude this chapter.

3.2. Problem statement

EVRPTW deals with a set of customers with known demands, service time windows, and service durations. The fleet is homogeneous and consists of EVs that are allowed to recharge their batteries en-route at charging stations. Although the energy transferred by a charger is a non-linear function of the recharge duration, recharging in the first phase is a linear function of time and it is practical in the real world to operate within this phase to prolong the battery life (Pelletier *et al.*, 2017). Therefore, without loss of generality, we assume a linear recharging function. Similar to the second chapter we allow one recharge between two consecutive customers, which is the practical case in last-mile logistics (Bruglieri *et al.*, 2016; Keskin and Çatay, 2018; Bruglieri *et al.*, 2018). We assume that all EVs depart with full battery as they can be recharged overnight in the depot. The objective function is to minimize the total energy consumption by using minimum number of vehicles.

We use the notation and modelling conventions in line with the recent literature for ease of understanding (see Keskin and Çatay, 2016; Bruglieri *et al.*, 2016). $V = \{1, \dots, n\}$ and F denote the set of customers and recharging stations, respectively. All EVs depart from node 0 (departure depot) and at the end of their tour return to node $n + 1$ (arrival depot). Let $V_0 = V \cup \{0\}$, $V_{n+1} = V \cup \{n + 1\}$ and $V_{0,n+1} = V \cup \{0, n + 1\}$. The problem can be defined on a complete directed graph $G = (N, A)$ where $A = \{(i, j) | i, j \in N, i \neq j\}$

represents the set of arcs and $N = V_{0,n+1} \cup F$ is the total set of nodes. Each customer $i \in V$ is associated with service time s_i , time window $[e_i, l_i]$, and a positive demand q_i . The battery and cargo capacities of the vehicles are denoted by Q and C , respectively. g is the recharge rate per unit time and h is discharge rate per unit distance. d_{ij} and t_{ij} represent the distance and travel time, respectively, from customer i to customer j if the journey is direct. If the vehicle recharges at station s during its trip from customer i to customer j , then $\hat{d}_{ijs} = d_{is} + d_{sj} - d_{ij}$ and $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$ denote the additional detour distance and travel time. Note that $\hat{t}_{ijs}$ does not include the recharging time at station s . The total energy consumption from customer i to customer j is calculated as $h_{ij} = h \times d_{ij}$ when the journey is direct whereas the additional energy consumption is $\hat{h}_{ijs} = h_{is} + h_{sj} - h_{ij}$ if the journey is via station s .

We keep track of battery SoC at arrival at customer/depot $i \in V_{0,n+1}$, at arrival at station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$, and at departure from station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$ using decision variables y_i , y_{ijs} , and Y_{ijs} , respectively. τ_i denotes service starting time at node $i \in N$. x_{ij} is a binary decision variable where it takes value 1 if node $j \in V_{n+1}$ visited after node $i \in V_0$ and 0 otherwise. The binary decision variable z_{ijs} takes value 1 if arc (i, j) is traversed, $i \in V_0, j \in V_{n+1}$, by visiting station $s \in F$ en-route. Table 3.1 summarize the mathematical notation.

Table 3.1 Mathematical notation for EVRPTW

Sets:	
V	Set of customers
V_0	Set of customers and departure depot
V_{n+1}	Set of customers and arrival depot
$V_{0,n+1}$	Set of customers, departure, and arrival depots
F	Set of recharging stations
N	Set of customers, stations, and depots
Parameters:	
d_{ij}	Distance between node i and j
$\hat{d}_{ijs}$	Additional distance of visiting station s between customers i and j , $\hat{d}_{ijs} = d_{is} + d_{sj} - d_{ij}$
t_{ij}	Travel time from node i and j
$\hat{t}_{ijs}$	Additional trip time of visiting station s between customers i and j , $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$
q_i	Demand of customer i
r_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

g Recharging rate

h_{ij} Total energy consumed to traverse arc (i, j) , $h_{ij} = h \times d_{ij}$

$\hat{h}_{ijs}$ Additional consumption if station s is visited between customers i and j , $\hat{h}_{ijs} = h_{is} + h_{sj} - h_{ij}$

Decision variables:

τ_i Service starting time at customer i

u_i Cargo level at departure from customer i

y_i Battery SoC of a vehicle upon arrival at (departure from) customer/depot $i \in V_{0,n+1}$

y_{ijs} Battery SoC of a vehicle upon arrival at station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$

Y_{ijs} Battery SoC of a vehicle at departure from station $s \in F$ on route (i, s, j) , $i \in V_0, j \in V_{n+1}$

x_{ij} 1 if a vehicle travels from node $i \in V_0$ to node $j \in V_{n+1}$; 0 otherwise

z_{ijs} 1 if a vehicle traverses arc (i, j) , $i \in V_0, j \in V_{n+1}$, through station $s \in F$; 0 otherwise

The 0-1 mixed-integer programming formulation of the problem is as follows:

$$\text{Min} \sum_{i \in V_0} \sum_{j \in V_{n+1}} (h_{ij}x_{ij} + \sum_{s \in F} \hat{h}_{ijs}z_{ijs}) + P \sum_{j \in V_{n+1}} x_{0j} \quad (3.1)$$

subject to

$$y_0 = Q \quad (3.2)$$

$$\sum_{\substack{j \in V_{n+1} \\ j \neq i}} x_{ij} = 1 \quad \forall i \in V \quad (3.3)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V_{n+1} \\ i \neq j}} x_{ji} = 0 \quad \forall j \in V \quad (3.4)$$

$$\sum_{s \in F} z_{ijs} \leq x_{ij} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.5)$$

$$\begin{aligned} \tau_i + (t_{ij} + r_i)x_{ij} + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs} + g(Y_{ijs} - y_{ijs})) \\ - l_0(1 - x_{ij}) \leq \tau_j \end{aligned} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.6)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (3.7)$$

$$q_i \leq u_i \leq u_j - q_j x_{ij} + C(1 - x_{ij}) \leq C \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.8)$$

$$0 \leq y_j \leq y_i - h_{ij}x_{ij} + Q(1 - x_{ij} + \sum_{s \in F} z_{ijs}) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.9)$$

$$y_j \leq \sum_{s \in F} (Y_{ijs} - h_{sj} z_{ijs}) + Q(1 - \sum_{s \in F} z_{ijs}) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.10)$$

$$+ Q(1 - x_{ij})$$

$$0 \leq y_{ijs} \leq y_i - h_{is} z_{ijs} + Q(1 - x_{ij}) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (3.11)$$

$$y_{ijs} \leq Y_{ijs} \leq Q z_{ijs} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (3.12)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (3.13)$$

$$z_{ijs} \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (3.14)$$

The objective function (3.1) minimizes the total energy consumed by a fleet consisting of minimum number of EVs. The latter is guaranteed with the last term which minimizes the number of routes by associating each with a large positive penalty P . Constraints (3.2(2.4) assure that the EVs depart the depot with their batteries fully recharged. Constraints (3.3) and (3.4(2.6) are the connectivity and flow conservation constraints. Constraints (3.5) guarantee that if the EV travels from customer i to customer j by recharging its battery en-route, then it serves customer j following customer i . Constraints (3.6) keep track of service times and ensure their consistency. Constraints (3.7) make sure that each customer is served within its predetermined time windows. Constraints (3.6) and (3.7) serve as subtour elimination constraints as well. Cargo capacity of the vehicles is imposed by constraints (3.8). Constraints (3.9)-(3.12) determine the battery SoC at each node and ensure that it is never negative. Constraints (3.9) check the battery SoC consistency if an EV travels from customer i to customer j without recharging en-route. Constraints (3.10) keep track of battery SoC at the arrival at customer j if the vehicle visits a recharging station after it has departed from customer i . Battery SoC at the arrival at a recharging station is established by Constraints (3.11). Constraints (3.12) impose lower and upper bounds on the battery SoC at departure from recharging stations. Finally, the domain of binary decision variables are defined by constraints (3.13) and (3.14).

3.3. Network reduction

Schneider *et al.* (2014) presented some basic preprocessing steps to remove the infeasible arcs. The removal of infeasible arcs may significantly reduce the graph and hence, accelerate the solution time. So, in this section, we propose a set of conditions that helps

identify infeasible arcs and remove them from the problem in an attempt to reduce the computational effort and enhance solution quality.

3.3.1. Connectivity of the depot to a customer

A vehicle departing from depot 0 can visit customer i if it can reach station $\bar{s}$ (the closest station to the customer i) after visiting customer i without violating the battery capacity and time-window restrictions. There are two possible cases: (a) the vehicle travels to customer i directly; (b) the vehicle travels to customer i after having its battery recharged en-route.

In case (a), we define θ_{0i} and $\theta_{i\bar{s}}$ as the earliest times that the vehicle can start serving customer i , and the vehicle can arrive at station $\bar{s}$, respectively. If at least one of the conditions (3.15)-(3.17) is violated, then the vehicle cannot travel from the depot to customer i directly.

$$h_{0i} + h_{i\bar{s}} \leq Q \quad i \in V \quad (3.15)$$

$$\theta_{0i} = e_0 + t_{0i} \leq l_i \quad i \in V, \bar{s} \in F \quad (3.16)$$

$$\theta_{i\bar{s}} = \max\{\theta_{0i}, e_i\} + r_i + t_{i\bar{s}} \leq l_{\bar{s}} \quad i \in V, \bar{s} \in F \quad (3.17)$$

Conditions (3.15) and (3.16) impose energy consumption and time-window restrictions for a vehicle to reach customer i after having departed from the depot. Condition (3.17) checks the time window feasibility of station $\bar{s}$.

Let F_{0i} be the set of feasible stations that can be visited when the vehicle travels from the depot to customer i . In case (b), we define θ_{0s} , θ_{si} , and $\theta'_{i\bar{s}}$ as the earliest times that the vehicle can start recharging at station $s \in F_{0i}$, the vehicle can start serving customer i , and the vehicle can arrive at $\bar{s}$, respectively. μ_s denotes the minimum amount of energy that the vehicle should recharge in station s in order to continue its route to reach station $\bar{s}$ after visiting customer i and is calculated as $\mu_s = h_{si} + h_{i\bar{s}} - (Q - h_{0s})$. Consider the following conditions:

$$h_{0s} \leq Q \quad s \in F \quad (3.18)$$

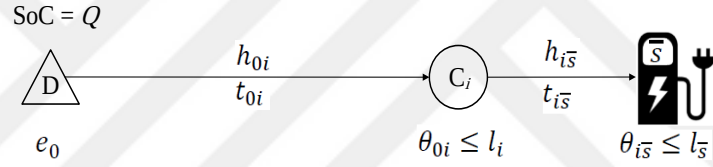
$$\theta_{0s} = e_0 + t_{0s} \leq l_s \quad s \in F \quad (3.19)$$

$$\theta_{si} = \max\{\theta_{0s}, e_s\} + g(\mu_s) + t_{si} \leq l_i \quad i \in V, s \in F \quad (3.20)$$

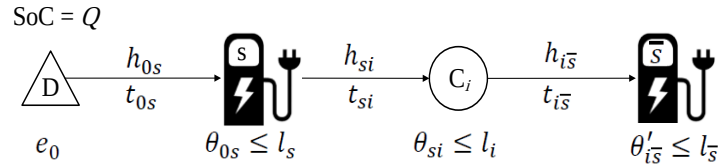
$$h_{si} + h_{i\bar{s}} \leq Q \quad i \in V, s \in F, \bar{s} \in F \quad (3.21)$$

$$\theta'_{i\bar{s}} = \max\{\theta_{si}, e_i\} + r_i + t_{i\bar{s}} \leq l_{\bar{s}} \quad i \in V, s \in F, \bar{s} \in F \quad (3.22)$$

Conditions (3.18) and (3.19) impose energy consumption and time-window restrictions for a vehicle to reach station s after having departed from the depot. Condition (3.20) checks the time window feasibility of customer i . Conditions (3.21) and (3.22) make sure that after visiting customer i the vehicle can arrive at station $\bar{s}$ without running out of energy and before its late service time $l_{\bar{s}}$, respectively. Station s is included in F_{0i} if it satisfies (3.18)-(3.22). If $F_{0i} = \emptyset$, then a vehicle departing from the depot cannot reach customer i after having been recharged en-route.



(a) The vehicle travels directly from the depot to customer i



(b) The vehicle is recharged while traveling from the depot to customer i

Figure 3.1 Conditions for the connectivity of the depot to customer i

Arc $(0, i)$ can be removed from the set of potential arcs in the network if the vehicle cannot travel to customer i directly or via a station from the depot. Both cases are illustrated in Figure 3.1.

3.3.2. Connectivity of a customer to another customer

Customer j is accessible from customer i if the vehicle can reach station $\underline{s}$ (closest station to customer j) after having departed from station $\bar{s}$ (closest station to customer i) and

visited customers i and j on its path without violating battery capacity and time-window restrictions. Again, there are two possible cases: (a) the vehicle travels to customer j directly; (b) the vehicle travels to customer j after having its battery recharged en-route. Case (a) is feasible if all the following conditions hold:

$$\theta_{\bar{s}i} = e_{\bar{s}} + t_{\bar{s}i} \leq l_i \quad i \in V, \bar{s} \in F \quad (3.23)$$

$$\theta_{ij} = \max \{\theta_{\bar{s}i}, e_i\} + r_i + t_{ij} \leq l_j \quad i, j \in V, \bar{s} \in F \quad (3.24)$$

$$\theta_{j\underline{s}} = \max \{\theta_{ij}, e_j\} + r_j + t_{j\underline{s}} \leq l_{\underline{s}} \quad i, j \in V, s_j^* \in F \quad (3.25)$$

$$h_{\bar{s}i} + h_{ij} + h_{j\underline{s}} \leq Q \quad i, j \in V, \bar{s}, \underline{s} \in F \quad (3.26)$$

Conditions (3.23), (3.24), and (3.25) make sure that the service start time at customer i , customer j , and station $\underline{s}$ are less than their latest service times, respectively. Condition (3.26) checks whether the vehicle battery capacity is sufficient to make this trip. If condition (3.26) does not hold, the vehicle cannot make this path without recharging between customers i and j .

For case (b), let F_{ij} be the set of feasible stations that can be visited between customers i and j . Then, consider the following conditions:

$$h_{\bar{s}i} + h_{is} \leq Q \quad i \in V, \bar{s} \in F, s \in F \quad (3.27)$$

$$h_{sj} + h_{j\underline{s}} \leq Q \quad j \in V, \underline{s} \in F, s \in F \quad (3.28)$$

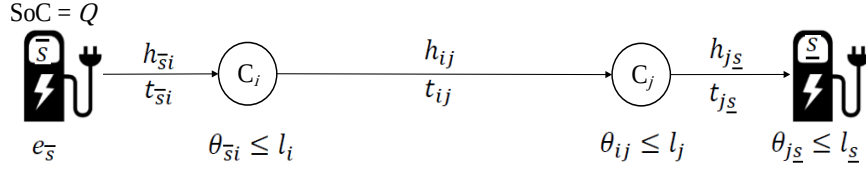
$$\theta_{is} = \max \{\theta_{\bar{s}i}, e_i\} + r_i + t_{is} \leq l_s \quad i \in V, \bar{s} \in F, s \in F \quad (3.29)$$

$$\theta_{sj} = \max \{\theta_{is}^*, e_s\} + g(\mu'_s) + t_{sj} \leq l_j \quad i, j \in V, s \in F \quad (3.30)$$

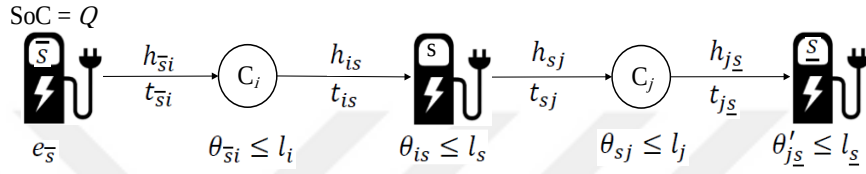
$$\theta'_{j\underline{s}} = \max \{\theta_{sj}, e_j\} + r_j + t_{j\underline{s}} \leq l_{\underline{s}} \quad j \in V, \underline{s} \in F, s \in F \quad (3.31)$$

where $\mu'_s = h_{sj} + h_{j\underline{s}} - (Q - h_{\bar{s}i} - h_{is})$ is the minimum energy that the vehicle needs to travel path $i \rightarrow s \rightarrow j$. Conditions (3.27) and (3.28) check the driving range to serve customer i and customer j , respectively. Conditions (3.29), (3.30), and (3.31) assure time-window feasibility of station s , customer j , and station $\underline{s}$, respectively. Station s is included in F_{ij} if (3.23) and (3.27)-(3.31) are satisfied. If $F_{ij} = \emptyset$, then the vehicle cannot travel from customer i to customer j via any station. Arc (i, j) can be removed from the set of potential arcs in the network if customer i is not connected to customer j directly

or via a station. Figure 3.2 shows the conditions that customer i is connected to customer j .



(a) The vehicle is recharged while traveling from customer i to customer j



(b) The vehicle travels directly from customer i to customer j

Figure 3.2 Conditions for the connectivity of customer i to customer j

3.3.3. Connectivity of a customer to the depot

Depot $n + 1$ is accessible from customer j if a fully charged vehicle departing from station $\underline{s}$ (closest station to customer j) can serve customer j and then reach the depot without violating the battery capacity and time-window restrictions. Again, the trip from customer j to the depot can be direct (a) or via a station (b). For case (a), consider the following conditions:

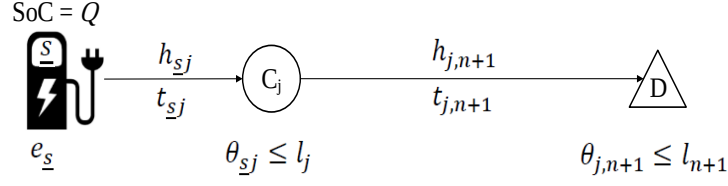
$$h_{\underline{s}j} + h_{j,n+1} \leq Q \quad j \in V, \underline{s} \in F \quad (3.32)$$

$$\theta_{\underline{s}j} = e_{\underline{s}} + t_{\underline{s}j} \leq l_j \quad j \in V, \underline{s} \in F \quad (3.33)$$

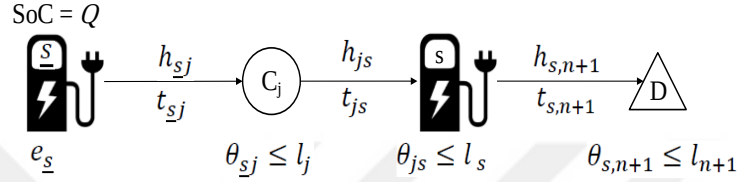
$$\theta_{j,n+1} = \max\{\theta_{\underline{s}j}, e_j\} + r_j + t_{j,n+1} \leq l_{n+1} \quad j \in V, \underline{s} \in F \quad (3.34)$$

Condition (3.32) checks the driving range feasibility whereas conditions (3.33) and (3.34) control the time-window feasibility of customer j and the depot at the end of the trip, respectively. $\theta_{\underline{s}j}$ denotes the earliest time that customer j can be served if the vehicle

travels from station $\underline{s}$ to customer j while $\theta_{j,n+1}$ represents the earliest time that the vehicle can return to depot after having served customer j .



(a) The vehicle is recharged while traveling from customer j to depot $n + 1$



(b) The vehicle travels directly from customer j to depot $n + 1$

Figure 3.3 Conditions for the connectivity of customer j to depot $n+1$

For case (b), let $F_{j,n+1}$ be the set of feasible stations that the vehicle can visit when traveling from customer j to depot $n + 1$. Consider the following conditions:

$$h_{\underline{s}j} + h_{js} \leq Q \quad j \in V, \underline{s} \in F, s \in F \quad (3.35)$$

$$\theta_{js} = \max\{\theta_{\underline{s}j}, e_j\} + s_j + t_{js} \leq l_s \quad j \in V, \underline{s} \in F, s \in F \quad (3.36)$$

$$h_{s,n+1} \leq Q \quad s \in F \quad (3.37)$$

$$\theta_{s,n+1} \leq l_{n+1} \quad s \in F \quad (3.38)$$

where $\mu_s'' = h_{s,n+1} - (Q - h_{\underline{s}j} - h_{js})$ represents the minimum amount of energy that the vehicle needs to recharge in station s in order to return to the depot. Conditions (3.35) and (3.36) check the driving range and time-window feasibility of the trip from customer j to station s , respectively, whereas conditions (3.37) and (3.38) do the same for the trip from station s to the depot. $\theta_{s,n+1} = \max\{\theta_{js}, e_s\} + g(\mu_s'') + t_{s,n+1}$ shows the earliest time that the vehicle can return to depot in this case. Station s is included in $F_{j,n+1}$ if (3.33) and (3.35)-(3.38) are satisfied. If $F_{j,n+1} = \emptyset$, then the vehicle cannot travel from customer j to depot via any station. Arc $(j, n + 1)$ can be removed from the set of arcs in the network if

customer j is not connected to the depot directly or via a station. Both cases are illustrated in Figure 3.3.

3.3.4. Valid inequalities

We add the following valid inequalities to tighten the model and speed up the solution time:

$$z_{0j1} = 0 \quad \forall j \in V_{n+1} \quad (3.39)$$

$$z_{i,n+1,1} = 0 \quad \forall i \in V_0 \quad (3.40)$$

$$y_i \leq Q \sum_{\substack{j \in V_{n+1} \\ j \neq i}} x_{ij} \quad \forall i \in V_0 \quad (3.41)$$

$$e_0 + t_{0i} \leq \tau_i \leq l_{n+1} - t_{i,n+1} \quad \forall i \in N \quad (3.42)$$

Noting that recharging station 1 is located at the depot, equalities (3.39) and (3.40) prevent a vehicle to be recharged there immediately after it has departed from or before it arrives at the depot. Inequalities (3.41) impose a limitation on battery SoC when the vehicle departs from a customer. Inequalities (3.42) tighten the time window constraints associated with the stations and customers.

Note that we also apply the well-known capacity cut that assign two customers on different routes if their total demand exceeds the cargo capacity and benefit from station dominance rules presented by Bruglieri *et al.* (2016).

3.4. Feasibility check algorithm

The problem is infeasible if at least one customer cannot be accessible from the depot because of the driving range and/or time window restriction. To detect such infeasibilities, we present an algorithm that checks whether the EV can reach every customer j from the depot and return to depot from that customer without running out of battery and violating

the corresponding time windows. Since the battery is full when departing from the depot but has less energy when departing from a customer, the infeasibility of the return trip is more likely. So, we give priority to checking first the path from the customer to the depot. The “path” here refers to the sequence of customers visited. To reduce the computation time, we adopt the k -shortest paths approach since shorter paths are more likely to be feasible due to less energy consumption. The flowchart of the algorithm is illustrated in Figure 3.4.

In the first step, we investigate whether the EV can return to depot after serving customer j either directly or by visiting other customers on its path without running out of energy. Note that since once recharge is permitted between two consecutive customers, the EV may have to visit multiple customers and be able to recharge its battery multiple times in order to reach the depot. We check if there is a feasible path from customer j to depot by solving k^{th} shortest path. If no such path exists then the problem is infeasible; otherwise, we select the shortest path from the set of shortest paths from customer j to depot. The maximum possible number for k is $\sum_{i=0}^{(n-2)} i! \binom{n-2}{i}$ where n is the number of customers plus depot.

If a feasible path is found in the first step, in the second step, the algorithm checks whether there exists a path from the depot to customer j (using k^{th} shortest path approach) such that none of the customers served in the return trip is visited again. If no feasible path exists, then the algorithm returns to step one and continues the same search procedure continuing from the $(k + 1)^{st}$ shortest path from customer j to depot and proceeds with the second step when a feasible path is determined. This procedure is repeated for all the customers until feasible forward and return paths are determined. If no such path exists for a customer, then the problem is infeasible. Note that all possible paths are evaluated in the worst case.

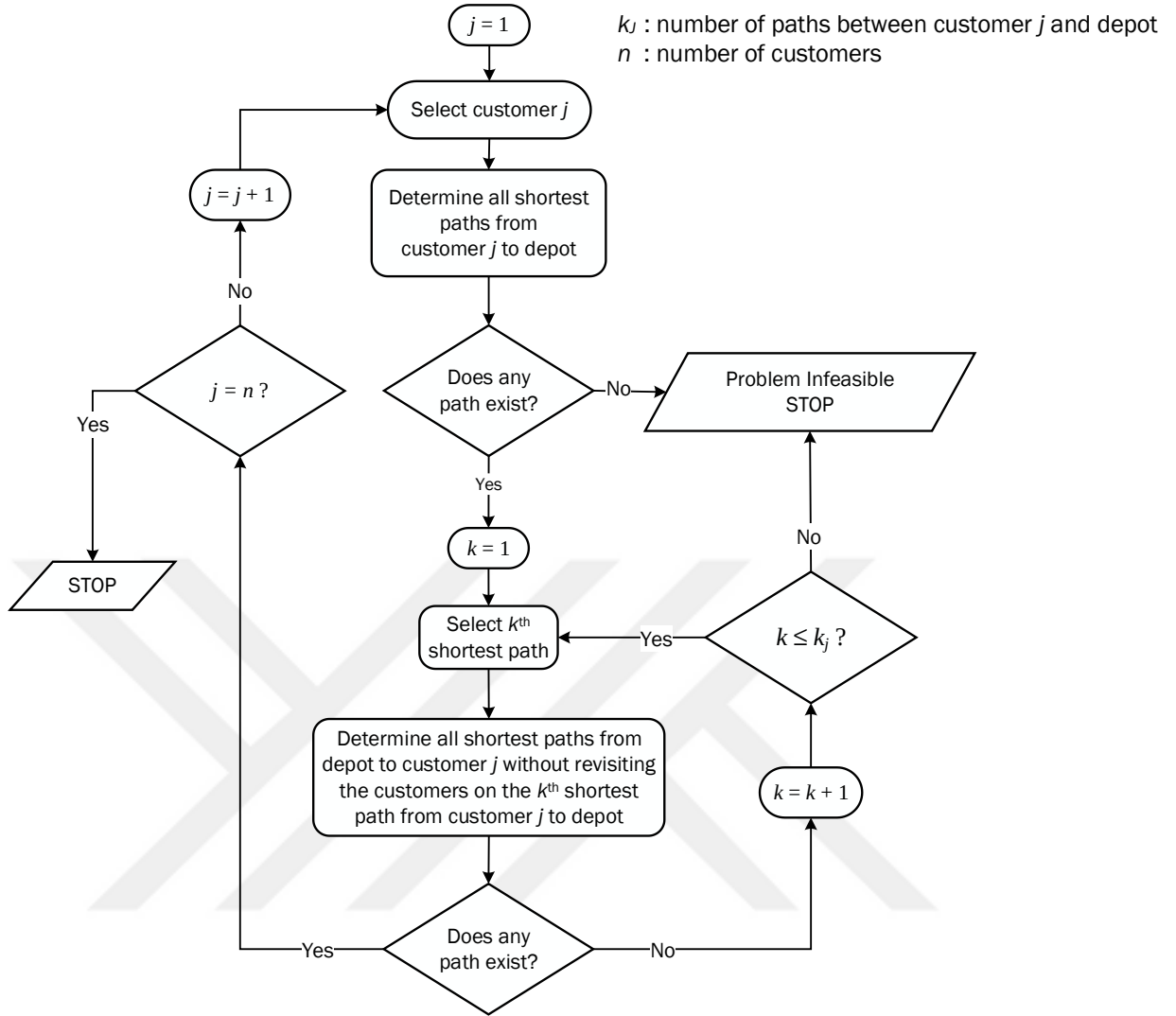


Figure 3.4 Feasibility check algorithm

3.5. Numerical results

To investigate the impact of the proposed preprocessing techniques on computation time and solution quality, we perform computational tests using Schneider *et al.* (2014) data. This data consists of three problem types where the customers are clustered (c-type), randomly distributed (r-type), and both clustered and randomly distributed (rc-type). The data is also classified in two types which differ by the length of the time windows, and the vehicle cargo and battery capacities. Subsets r1, c1, and rc1 involve narrow time windows and shorter scheduling horizon whereas the time windows are wide, and the scheduling horizon is longer in subsets r2, c2, and rc2. The data includes small and large-

size instances. In our investigation we used 12 small-size instances which involve 15 customers and 3 to 8 recharging stations and 29 type-1 large-size instances which consist of 100 customers and 21 stations.

Although all the instances in Schneider *et al.* (2014) dataset are feasible, we usually do not know whether all the customers in a given data can be served using an EV fleet subject to range anxiety. In the second chapter we investigated the influence of ambient temperature on routing decisions and observed that a problem may become infeasible because of increased energy consumption due to the utilization of heater/cooler in cold/hot weather conditions. Furthermore, the graph may benefit more from the proposed preprocessing techniques as its size may reduce with increased consumption rate. So, to better investigate the performance of the preprocessing techniques and feasibility check algorithm, we consider three different temperature conditions as described in chapter 2: mild (22°C), intermediate (8°C or 27°C), and intense (0°C or 33°C). The original data corresponds to the mild temperature case where the consumption rate is set equal to 1.0 unit of energy per unit distance/time traveled and it increases to 1.09 and 1.27 in intermediate and intense cases, respectively.

All the experiments were performed on a workstation equipped with Intel(R) Core (TM) i7-8700 processor with 3.20 GHz speed and 32 GB RAM. Gurobi 9.00 optimization solver was used to solve the small-size instances with 2-hour run-time limit. To solve large-size instances, we employ ALNS.

3.5.1. Analysis of network reduction

Table 3.2 **Error! Reference source not found.** reports the average network densities under different temperature conditions after applying the proposed preprocessing procedure to different data types. ‘#Inst’ refers to the number of instances. In the benchmark problems, the customer network is complete (in mild temperature conditions); so, the network density is 100% without any preprocessing. We observe that in small-size dataset the network density decreases to 76.73% on the average. In other words, almost $\frac{1}{4}$ of the x_{ij} variables can be eliminated from the problem. The reduction is more significant in intermediate and intense cases as expected, where the average densities are

73.92 and 63.60%, respectively. Similar reduction behavior can be observed in large-size instances as well. These results indicate that a solution method may greatly benefit from the proposed reduction techniques.

Table 3.2 Average customer-network densities after preprocessing

Data Type	Small-size Instances				Large-size Instances			
	#Inst	Mild	Intermediate	Intense	#Inst	Mild	Intermediate	Intense
r	4	73.62	70.06	56.88	9	74.39	72.36	67.14
c	4	76.55	74.42	65.48	12	71.50	69.15	63.11
rc	4	80.02	77.27	68.45	8	69.32	66.72	61.44
All	12	76.73	73.92	63.60	29	72.09	69.81	64.31

3.5.2. Performance on small-size problems

The results for mild, intermediate, and intense cases for small-size instances are presented in Table 3.3, Table 3.4, and Table 3.5, respectively. In these tables, columns ‘#Veh’, ‘EC’, ‘ $t(sec)$ ’, and ‘%Gap’ refer to the number of vehicles, energy consumption, run time in seconds and the percentage gap between the upper bound and lower bound as reported by Gurobi in 2-hour time limit, respectively whereas ‘% Δt ’ in the last column reports the percentage acceleration in run time.

Table 3.3 Results for small-size instances in the mild case

Instance	Without Preprocessing				With Preprocessing				
	#Veh	EC	$t(sec)$	%Gap	#Veh	EC	$t(sec)$	%Gap	% Δt
r102c15-s8	5	412.78	1.23	0.00	5	412.78	0.62	0.00	49.37
r105c15-s6	4	336.15	0.86	0.00	4	336.15	0.86	0.00	0.00
r202c15-s6	1	507.32	7013.14	0.00	1	507.32	1435.40	0.00	79.53
r209c15-s5	1	313.24	310.88	0.00	1	313.24	252.14	0.00	18.89
c103c15-s5	3	348.46	7200.00	32.75	3	348.46	7200.00	32.72	0.00
c106c15-s3	3	275.13	108.76	0.00	3	275.13	4.98	0.00	95.42
c202c15-s5	2	383.62	215.96	0.00	2	383.62	194.13	0.00	10.11
c208c15-s4	2	300.55	102.59	0.00	2	300.55	67.35	0.00	34.35
rc103c15-s5	4	397.67	7200.00	43.34	4	397.67	607.98	0.00	91.56
rc108c15-s5	3	370.25	7200.00	98.60	3	370.25	7200.00	97.44	0.00
rc202c15-s5	2	394.39	8.88	0.00	2	394.39	5.09	0.00	42.62
rc204c15-s7	1	382.22	7200.00	96.81	1	382.22	7200.00	96.80	0.00
Average	2.58	368.48	3046.86	22.63	2.6	368.48	2014.05	18.91	35.15

In the mild case (original data), the preprocessing did not improve the solution quality; however, it reduced the average *%Gap* from 22.63% to 18.91% and speeded up the solver by 35% on the average. Particularly in problem ‘rc103c15-s5’, the solver stopped after 2 hours with a 43% gap whereas it proved the optimality of the upper bound in almost 10 minutes when preprocessing was applied.

Table 3.4 Results for small-size instances in the intermediate case

Instance	Without Preprocessing				With Preprocessing				
	#Veh	EC	t(sec)	%Gap	#Veh	EC	t(sec)	%Gap	% Δt
r102c15-s8	5	469.75	1.78	0.00	5	469.75	0.80	0.00	55.26
r105c15-s6	4	396.56	0.95	0.00	4	396.56	0.59	0.00	37.71
r202c15-s6	2	403.24	4215.16	0.00	2	403.24	619.78	0.00	85.30
r209c15-s5	1	363.52	286.93	0.00	1	363.52	61.33	0.00	78.63
c103c15-s5	3	408.13	7200.00	32.74	3	408.13	7200.00	32.68	0.00
c106c15-s3	3	382.33	71.08	0.00	3	382.33	22.24	0.00	68.72
c202c15-s5	2	418.14	393.79	0.00	2	418.14	206.59	0.00	47.54
c208c15-s4	2	327.60	26.25	0.00	2	327.60	14.25	0.00	45.72
rc103c15-s5	4	433.46	5206.26	0.00	4	433.46	635.09	0.00	87.80
rc108c15-s5	3	428.86	7200.00	97.92	3	428.86	7200.00	65.69	0.00
rc202c15-s5	2	475.28	8.61	0.00	2	475.28	5.95	0.00	30.89
rc204c15-s7	1	419.49	7200.00	96.55	1	419.49	7200.00	96.54	0.00
Average	2.67	410.53	2650.90	18.93	2.67	410.53	1930.55	16.24	44.80

Table 3.4 Table 3.4 shows that in the intermediate case, preprocessing reduced the average *%Gap* from 18.93% to 16.24% and accelerated Gurobi by 44.80% on the average. Furthermore, in problem ‘rc108c15-s5’, it allowed Gurobi to reduce the optimality gap from 97.92% to 65.59% in 2-hour time limit.

Table 3.5 Results for small-size instances in the intense case

Instance	Without Preprocessing				With Preprocessing				
	#Veh	EC	t(sec)	%Gap	#Veh	EC	t(sec)	%Gap	% Δt
r102c15-s8	—	INF	1.39	0.00	—	INF	< 0.01	0.00	99.99
r105c15-s6	5	514.52	1.55	0.00	5	514.52	0.67	0.00	56.57
r202c15-s6	—	INF	5691.56	0.00	—	INF	< 0.01	0.00	99.99
r209c15-s5	—	NS	7200.00	—	—	INF	< 0.01	0.00	99.99
c103c15-s5	3	509.10	7200.00	32.63	3	509.10	6081.78	0.00	15.53
c106c15-s3	3	525.15	20.30	0.00	3	525.15	1.62	0.00	92.00
c202c15-s5	2	571.39	104.37	0.00	2	571.39	10.08	0.00	90.34
c208c15-s4	2	500.95	76.58	0.00	2	500.95	13.52	0.00	82.35
rc103c15-s5	4	508.13	1880.45	0.00	4	508.13	217.83	0.00	88.42
rc108c15-s5	—	NS	7200.00	—	—	INF	0.02	0.00	99.99
rc202c15-s5	—	INF	16.86	0.00	—	INF	< 0.01	0.00	99.99
rc204c15-s7	1	509.49	7200.00	95.96	1	509.49	7200.00	95.94	0.00
Average	2.86	519.82	3049.42	12.86	2.86	519.82	1127.13	8.00	77.10

The results for the intense case are presented in Table 3.5. In this case, preprocessing reduced the average %Gap from 12.86% to 8.00% and speeded up the solver by 77.10% on the average. It also allowed Gurobi prove the optimality of the upper bound for problem ‘c103c15-s5’ and infeasibility of ‘r209C15-s5’ and ‘rc108C15-s5’ in less than a second. Note that without preprocessing Gurobi could neither find an upper bound for these two problems nor prove they are infeasible at the end of 2 hours.

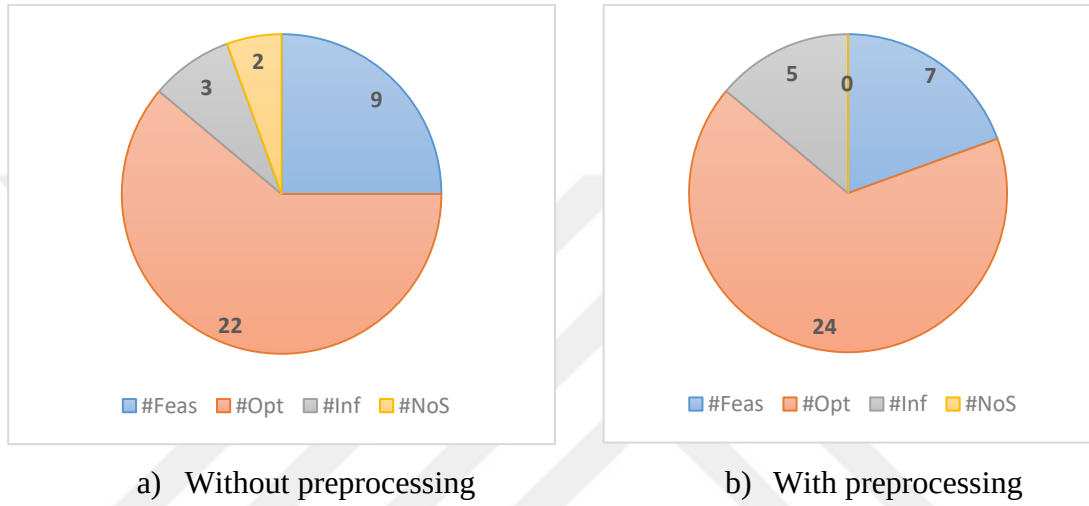


Figure 3.5 Summary of results for small-size instances

The results on small-size instances are summarized in Figure 3.5. In this figure, ‘#Feas’, ‘#Opt’, ‘#Inf’, and ‘NoS’ indicate the number of instances that are feasible but could not be solved optimally in 2 hours, the number instances solved optimally, the number of infeasible instances, and the number of instances with no solution, respectively. The figure shows that Gurobi equipped with preprocessing found the optimal solution in two instances which could not be solved optimally without preprocessing within 2-hour time limit. Furthermore, Gurobi was unable to provide an upper bound in two instances neither could it prove their infeasibility within the given time limit whereas the proposed preprocessing allowed it to prove that they are infeasible.

3.5.3. Performance on large-size problems

To investigate the performance of the proposed preprocessing on large-size instances, we implement it in an ALNS algorithm. ALNS involves a destroy-and-repair framework where the current solution is partially destroyed at each iteration by randomly removing some customers and then repaired by inserting the removed customers into the partial solution in an attempt to improve the incumbent solution. Numerical scores are associated with each insertion and removal operator, which are updated based on their performances after each iteration. In this chapter, we employ the ALNS algorithm presented in chapter 2 and perform 10 runs on each instance.

Table 3.6 Result for large-size instances in mild case

Instance	Without Preprocessing			With Preprocessing			
	#Veh	EC	$t(sec)$	#Veh	EC	$t(sec)$	% Δt
c101	12	1043.38	139.05	12	1043.38	132.77	4.52
c102	11	1019.68	315.39	11	1019.68	294.00	6.78
c103	10	973.92	772.10	10	974.09	719.03	6.87
c104	10	886.84	1638.93	10	886.76	1351.14	17.56
c105	11	1022.75	150.58	11	1046.01	145.52	3.36
c106	11	1022.15	219.95	11	1009.75	213.68	2.85
c107	10	1070.30	276.25	10	1050.18	248.77	9.95
c108	10	1044.76	339.77	10	1045.91	314.63	7.40
c109	10	943.69	472.16	10	969.72	449.16	4.87
r101	18	1642.74	62.01	17	1685.19	57.27	7.65
r102	16	1437.66	142.77	15	1513.97	126.18	11.62
r103	13	1256.89	151.76	13	1263.42	142.80	5.91
r104	11	1096.27	279.72	11	1086.90	250.74	10.36
r105	15	1402.64	75.89	14	1454.37	69.14	8.89
r106	13	1370.55	126.30	13	1313.64	120.70	4.43
r107	11	1154.86	215.79	11	1143.30	199.49	7.56
r108	11	1043.75	302.94	11	1052.60	290.06	4.25
r109	13	1216.39	121.16	13	1212.84	114.72	5.32
r110	12	1114.47	215.51	12	1107.45	201.33	6.58
r111	12	1108.86	227.41	12	1105.93	218.57	3.89
r112	11	1035.29	297.02	11	1034.00	282.92	4.75
rc101	15	1755.13	85.10	15	1691.93	75.34	11.47
rc102	14	1585.91	111.91	14	1526.15	93.03	16.87
rc103	13	1360.77	159.69	12	1380.54	141.00	11.70
rc104	11	1213.39	251.45	11	1196.25	229.46	8.75
rc105	14	1470.17	85.57	14	1460.91	82.28	3.85
rc106	13	1452.45	109.35	13	1421.57	95.66	12.52
rc107	12	1266.05	180.15	12	1265.90	163.85	9.04
rc108	11	1193.58	211.46	11	1163.32	191.39	9.49
Average	12.21	1174.46	266.80	12.07	1163.66	241.88	7.90

The results of large-size instances are shown in Table 3.6, Table 3.7 and Table 3.8 for mild, intermediate, and intense cases, respectively. In these tables, column ' $t(sec)$ ' reports the average computation time of 10 runs in seconds and column ' $\% \Delta t$ ' shows the percentage reduction in the average run time. The last row presents the averages for the whole data set. Note that the average energy consumption reported here is calculated for the instances whose solutions involve same number of vehicles with and without using preprocessing. **#Veh** and **EC** values in bold indicate an improvement in the number of vehicles and energy consumption, respectively.

Table 3.6 Table 3.7 shows the results for the mild case. We observe that the preprocessing can reduce the run time of ALNS by up to 17.56% whereas the average speedup is 7.90%. Furthermore, ALNS with preprocessing provided better solutions in 21 instances out of 29, four regarding fleet size and 17 regarding energy consumption.

Table 3.7 Result for large-size instances in the intermediate case

Instance	Without Preprocessing			With Preprocessing			
	#Veh	EC	t(sec)	#Veh	EC	t(sec)	% Δt
c101	12	1142.19	133.88	12	1142.19	117.05	12.57
c102	11	1128.89	261.35	11	1129.43	237.99	8.94
c103	10	1085.16	722.05	10	1147.76	643.74	10.85
c104	10	988.31	1187.30	10	979.24	947.11	20.23
c105	11	1153.29	141.83	11	1153.29	124.28	12.37
c106	11	1138.75	200.44	11	1122.87	189.63	5.39
c107	11	1116.60	260.65	11	1114.98	225.54	13.47
c108	11	1098.49	384.33	10	1158.48	293.69	23.58
c109	10	1087.70	398.80	10	1047.38	369.69	7.30
r101	18	1809.06	72.14	18	1839.91	63.75	11.63
r102	16	1605.22	152.46	16	1589.85	125.51	17.68
r103	14	1395.47	179.72	14	1374.79	152.76	15.00
r104	12	1191.54	321.32	12	1190.98	271.77	15.42
r105	15	1557.40	90.43	15	1576.50	80.34	11.16
r106	14	1420.32	141.72	13	1433.83	124.80	11.94
r107	12	1255.45	215.52	12	1243.24	193.67	10.14
r108	11	1143.11	379.74	11	1169.01	325.27	14.34
r109	13	1361.21	131.83	13	1349.94	116.51	11.62
r110	12	1225.59	216.36	12	1224.70	190.80	11.81
r111	12	1259.48	223.51	12	1231.42	206.11	7.79
r112	11	1161.81	319.61	11	1149.99	288.78	9.65
rc101	16	1901.34	94.59	16	1908.44	76.81	18.79
rc102	15	1717.75	108.19	15	1688.56	88.01	18.65
rc103	13	1529.18	150.35	13	1471.51	128.54	14.51
rc104	12	1350.57	236.83	12	1341.80	209.06	11.73
rc105	14	1647.14	91.96	14	1644.00	81.63	11.23
rc106	13	1706.69	101.23	13	1673.52	83.32	17.69
rc107	12	1396.43	186.32	12	1405.94	152.84	17.97
rc108	11	1326.87	231.81	11	1330.47	193.29	16.62
Average	12.52	1347.49	252.97	12.45	1342.29	217.32	13.45

Table 3.7 reports the results for the intermediate case. We see that preprocessing accelerates ALNS by 13.45% on average with a maximum speedup of 23.58%. Furthermore, better solution quality is achieved in 17 instances, two with reduced fleet size and 15 with less energy consumption.

Table 3.8 Result for large-size instances in the intense case

Instance	Without Preprocessing			With Preprocessing			
	#Veh	EC	t(sec)	#Veh	EC	t(sec)	% Δt
c101	12	1363.48	115.76	12	1363.04	90.49	21.82
c102	11	1388.57	358.81	11	1406.31	255.04	28.92
c103	11	1302.35	678.06	11	1301.08	475.62	29.85
c104	11	1253.78	1185.04	10	1318.18	806.17	31.97
c105	11	1432.27	179.03	11	1424.47	131.59	26.50
c106	11	1449.76	240.94	11	1403.87	174.91	27.41
c107	11	1356.79	197.78	11	1368.01	152.88	22.70
c108	11	1338.30	288.27	11	1319.37	207.77	27.93
c109	11	1288.30	388.06	11	1296.39	303.45	21.80
r101	19	2265.32	72.70	19	2232.10	62.14	14.52
r102	16	1989.45	149.07	16	1998.27	107.28	28.04
r103	14	1721.53	172.21	14	1697.69	134.13	22.11
r104	12	1444.11	321.02	12	1423.55	255.26	20.49
r105	16	1868.20	104.33	16	1831.89	88.50	15.18
r106	15	1704.90	176.32	14	1748.57	122.24	30.67
r107	13	1537.90	214.62	13	1518.26	176.85	17.60
r108	12	1389.93	415.33	12	1386.60	312.81	24.68
r109	14	1638.86	127.33	14	1631.04	102.03	19.87
r110	13	1471.46	229.11	13	1484.83	173.70	24.19
r111	13	1477.05	222.29	13	1482.63	175.07	21.24
r112	12	1365.34	332.78	12	1364.46	249.61	24.99
rc101	17	2394.81	82.21	17	2348.68	67.82	17.50
rc102	16	2130.26	140.72	16	2109.60	111.21	20.97
rc103	14	1874.29	142.96	14	1847.07	108.72	23.95
rc104	12	1690.18	260.09	12	1668.35	191.87	26.23
rc105	15	2041.41	110.88	15	1995.72	89.36	19.41
rc106	15	1910.64	114.38	15	1953.04	79.23	30.73
rc107	13	1737.06	177.62	13	1735.73	131.49	25.97
rc108	12	1655.34	214.08	12	1637.30	165.88	22.52
Average	13.21	1649.00	255.58	13.14	1638.12	189.76	23.79

The results for the intense case are given in Table 3.8. As the energy consumption per unit distance is higher in this case, the graph can be reduced further and ALNS can benefit more from the preprocessing. As expected, ALNS provides better solutions in 22 instances, two and 20 with respect to the number of vehicles and energy consumption, respectively. Furthermore, it accelerates the computational time up to 31.97% and the average speedup is 23.79% for the whole data set.

Table 3.9 Summary of results for large-size instances

Objective	# Instances Improved		
	Mild	Intermediate	Intense
#Veh	4	2	2
EC	17	15	20

Table 3.9 summarizes the improvements observed in large-size instances. The total number of instances that we considered is $29 \times 3 = 87$. Overall, in 8 instances we see reduction in the fleet size and saving in the energy consumption in 52 instances. It is worth noting that some of these improvements may be due to the stochastic nature of ALNS. Nevertheless, the results show that the proposed preprocessing is effective in enhancing the performance in terms of both solution quality and computation time.

3.5.4. Identifying infeasibility

We compared the performance of the proposed feasibility check algorithm and that of Gurobi with and without using the preprocessing procedures as well as determining the infeasibility of a problem, if exists. Note that while all the large-size instances are feasible five out of 36 small-size instances are shown to be infeasible (using preprocessing). We consider these five instances and summarize in Table 3.10 the average run times and number of instances proven to be infeasible by Gurobi (with and without preprocessing) and the proposed feasibility check algorithm. Without employing the proposed preprocessing Gurobi can identify three as infeasible with an average computation time of 4021 seconds whereas it can prove all five to be infeasible in only 0.003 seconds on the average when it is equipped with the preprocessing. Furthermore, the proposed feasibility check algorithm can also identify these 5 infeasible instances in only 0.4 seconds on the average. These results support the benefit of the proposed preprocessing and show that an instance which is infeasible can be effectively identified using the feasibility check algorithm without using a solver.

Table 3.10 Average results for identifying infeasibilities

	Gurobi without Preprocessing	Gurobi with Preprocessing	Feasibility Check Algorithm
Average time (s)	4021.960	0.003	0.400
#Infeasible identified	3	5	5

3.6. Concluding remarks

In this chapter, we revisited the EVRPTW and proposed new preprocessing techniques to reduce the problem size and accelerate the computational performance of the solution methods. In addition, we developed an algorithm to check the feasibility of an instance. Our experimental tests showed that the proposed preprocessing can reduce the graph by up to 37% on average and speed up the computation time by up to 95% and 32% on small- and large-size instances, respectively. Furthermore, it allowed ALNS to find better solutions in many large-size instances, by reducing the fleet size or saving energy. Moreover, the proposed feasibility check algorithm was able to detect all the instances proven infeasible.

4. Electric Vehicle Routing Problem with Time Windows and Cargo Weight

4.1. Introduction to Electric Vehicle Routing Problem with Time Windows considering cargo load

Electric Vehicle Routing Problem (EVRP) is an extension for VRP, where EVs are used in the fleet instead of fossil fuel vehicles. EVs reduce tailpipe emission and enhance green logistics. It tries to handle distribution tasks of logistics companies by minimizing the total energy consumption cost of serving customers and satisfying their demands. EVRP with Time Window (EVRPTW) is introduced by Schneider et al. (2014) where a full-recharge strategy was adopted. The authors developed the mathematical programming formulation of the problem and proposed a hybrid Variable Neighbourhood Search (VNS) and Tabu Search (TS) algorithm to solve it. Different variants of EVRP and EVRPTW were addressed in several studies including the cases of partial recharge (Bruglieri et al, 2015; Keskin and Çatay, 2016), mixed fleet (Goeke and Schneider, 2015; Hiermann et al, 2016), location routing (Schiffer and Walther, 2017), fast charging (Feliipe et al., 2014; Çatay and Keskin, 2017; Keskin and Çatay, 2018), non-linear charging function (Montoya et al.,2017; Froger et al., 2019), battery swapping (Yang and Sun, 2015; Hof et al., 2017; Paz et al., 2018). Desaulniers et al. (2016) also studied EVRPTW and proposed a branch-price-and-cut algorithm to solve four different recharging strategies. Some recent studies have dealt with the availability of recharging stations and queueing for recharging service (Froger et al., 2017; Kullman et al., 2018; Keskin et al., 2019). A comprehensive review of the EV technology and survey of the EVRP variants may be found in Pelletier et al. (2016), Pelletier et al. (2017), and Erdelić and Carić (2019).

Energy consumption on the road does not only depend on the distance traveled but many other factors including the vehicle's weight, velocity, auxiliary equipment (internal factors) as well as ambient temperature and road gradient (external factors). These factors have been often neglected in the VRP literature either because they make the problem too complex to solve or the driving range is not an issue as the vehicles can easily refuel at a nearby gas station. However, they may play a critical role in the operational efficiency of the EVs since they can increase their energy consumption significantly which is discussed in chapter 2. Among them, the weight of the transported cargo may play a crucial role in route planning. The logistics operations of hypermarkets, hardware stores and other companies that deal with heavy loads are examples for which a load-dependent model produces more efficient transportation plans in comparison with basic routing models (Zachariadis et al., 2015), which constitutes the main motivation of this study.

Load-dependent Vehicle Routing Problem (LDVRP) was introduced in Kara et al. (2007). They used the weighted distance objective and relate it with the energy requirements of vehicles. They proposed mathematical formulations for collection and distribution cases. Xiao et al. (2012) attacked the same problem by emphasizing the relation of the weighted distance with the fuel consumption of the vehicles within the context of Fuel Capacitated VRP. Zachariadis et al. (2015) extended LDVRP by considering simultaneous pick-ups and deliveries and proposed a local-search algorithm to solve large-scale instances.

In this chapter, we address the load-dependent variant of EVRPTW with partial recharges by taking into account the energy consumption associated with the cargo carried on the vehicle. We adopt a hierarchical objective function where the primary objective is to minimize the fleet size whereas the secondary objective is to minimize total energy consumption. We solve small-size instances using a commercial solver, and for the large-size instances, we develop a Large Neighbourhood Search (LNS) algorithm. The remainder of this chapter is organized as follows: section 4.2 introduces the problem and formulates its mathematical programming model. Section 4.3 describes the proposed LNS method. Section 4.4 presents the experimental study and discusses the results. Finally, concluding remarks are provided in section 4.5.

4.2. Problem definition and formulations

We tackle EVRPTW where a homogeneous fleet of EVs serve a set of customers with known demands, time windows, and service times. As opposed to previous studies in the literature which assume that the energy on the battery is consumed proportional to the distance traveled, we take into account the additional energy consumption related to freight load. Carrying more load by an EV causes more energy consumption. Furthermore, we allow partial recharging and its duration depends on the amount of energy transferred. Since it is a common practice in the real world to operate within the first phase of recharging where the energy transferred is a linear function of the recharge duration in order to prolong the battery life (Pelletier et al., 2017), we also assume a linear charging function. In addition, we assume that the EV can be recharged at most once between two consecutive customers, which is practical in last-mile logistics. We consider a pick-up problem where the load of the EV increases along its tour as it visits the customers. Each EV departs from the depot with the full battery since it can be recharged overnight.

4.2.1. Energy consumption function

The energy consumption of an EV that travels from one node to another depends on various factors such as its mass, shape, road gradient, acceleration, etc. By using tractive power requirements placed on the vehicle at the wheels, the power demand of a vehicle can be obtained using function (4.2) (Demir et al., 2012):

$$F = Ma + Mg\sin\theta + 0.5C_d\rho AV^2 + MgC_r\cos\theta \quad (4.1)$$

$$P_{tract}(kW) = Fv/1000 \quad (4.2)$$

where F shows the tractive effort as calculated in (4.1), M is the total weight of the vehicle that consist of its curb weight and the cargo load (kg), a is the acceleration (m/sec^2), g is the gravitational constant, θ is road gradient, C_d is the coefficient of aerodynamic drag, ρ is the air density in (kg/m^3), A is the frontal area, v is the speed (m/s), and C_r the

coefficient of rolling resistance. The tractive power requirement can be converted to second-by-second battery power output (kW) as follows:

$$P = P_{tract}/\mu_{tf} + P_{acc} \quad (4.3)$$

where the vehicle's drive train efficiency is shown by μ_{tf} which includes the energy losses between electric motor and battery as well as the energy losses in transforming energy to the wheels. P_{acc} is the power demand associated with the accessory equipment such as air conditioning, audio system, cabin lights, which is neglected in this study. Then, the energy consumption in (kWh/km) can be calculated as follows:

$$E = P/v \quad (4.4)$$

4.2.2. Mathematical models

We presented and evaluated three models to check their performances.

Model I. In line with the mathematical notation and modelling convention in the literature (Schneider et al., 2014; Keskin and Çatay 2016) we define $V = \{1, \dots, n\}$ as the set of customers and F as the set of recharging stations. Vertices 0 and $n + 1$ denote the depot where each vehicle departs from 0 (departure depot) and returns to $n + 1$ (arrival depot) at the end of its tour. We define $V_0 = V \cup \{0\}$, $V_{n+1} = V \cup \{n + 1\}$ and $V_{0,n+1} = V \cup \{0, n + 1\}$. Then, the problem can be represented on a complete directed graph $G = (N, A)$ with the set of arcs $A = \{(i, j) | i, j \in N, i \neq j\}$, where $N = V_{0,n+1} \cup F$ is the total set of nodes on the network.

The energy consumption depends on the distance traveled and the total weight of the EV, which is affected by the cargo load carried on the EV. Each customer $i \in V$ has a positive demand q_i , service time s_i , and time window $[e_i, l_i]$. All EVs have a cargo capacity of C and a battery capacity of Q . At each recharging station, one unit of energy is transferred in g time units. The direct distance from node i to j is represented by d_{ij} .

Travel time from customer i to customer j is denoted by t_{ij} if the journey is direct and $\hat{t}_{ijs} = t_{is} + t_{sj} - t_{ij}$ is the additional travel time if it is via station s . Note that $\hat{t}_{ijs}$ does not include the recharging time at station s . The amount of extra energy needed in order to move one unit of cargo is represented by w . The total energy consumption starting from customer i to customer j is calculated as $(h + wu_i)d_{ij}$, where u_i is the weight of the load on the vehicle upon departure from customer i .

The decision variables y_i^k, y_{ijs}^k , and Y_{ijs}^k , keep track of battery SoC of vehicle k at arrival at customer/depot i , at arrival at station s on route (i, s, j) , and at departure from station s on route (i, s, j) , respectively. τ_i denotes the time when the loading starts at customer i . The binary decision variable x_{ij}^k takes value 1 if vehicle k travels from node i to node j , and 0 otherwise whereas the binary decision variable z_{ijs}^k takes value 1 if vehicle k traverses arc (i, j) , through station s .

$$\text{Min } \sum_{k \in K} (y_0^k - y_{n+1}^k) + \sum_{i \in V_0} \sum_{j \in V_{n+1}} \sum_{k \in K} \sum_{s \in F} (Y_{ijs}^k - y_{ijs}^k) \quad (4.5)$$

subject to

$$y_0^k = Q \quad \forall k \in K \quad (4.6)$$

$$\sum_{j \in V_{n+1}} \sum_{k \in K} x_{ij}^k = 1 \quad \forall i \in V \quad (4.7)$$

$$\sum_{i \in V_0} x_{ij}^k - \sum_{i \in V_{n+1}} x_{ij}^k = 0 \quad \forall j \in V, k \in K \quad (4.8)$$

$$\sum_{s \in F} z_{ijs}^k \leq x_{ij}^k \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (4.9)$$

$$\tau_i + (t_{ij} + r_i)x_{ij}^k + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs}^k + g(Y_{ijs}^k - y_{ijs}^k)) - l_0(1 - x_{ij}^k) \leq \tau_j \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (4.10)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (4.11)$$

$$0 \leq y_j^k \leq y_i^k - (h + wu_i)d_{ij} + M(1 - x_{ij}^k + \sum_{s \in F} z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (4.12)$$

$$y_j^k \leq Y_{ijs}^k - (h + wu_i)d_{sj} + M(1 - z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (4.13)$$

$$0 \leq y_{ijs}^k \leq y_i^k - (h + wu_i)d_{is} + M(1 - z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (4.14)$$

$$y_{ijs}^k \leq Y_{ijs}^k \leq Qz_{ijs}^k \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (4.15)$$

$$y_j^k \leq Q \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij}^k \quad \forall j \in V_{n+1}, k \in K \quad (4.16)$$

$$u_j \geq u_i + q_j \sum_{k \in K} x_{ij}^k - C(1 - \sum_{k \in K} x_{ij}^k) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.17)$$

$$0 \leq u_i \leq C \quad \forall i \in V_{0,n+1} \quad (4.18)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (4.19)$$

$$z_{ijs}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (4.20)$$

The objective function (4.5) minimizes the total energy consumption. Constraints (4.6) set the initial battery SoC of EVs at departure to full. The connectivity of customer visits is imposed by constraints (4.7) whereas the flow conservation at each vertex is ensured by constraints (4.8). Constraints (4.9) make sure that vehicle k serves customer j after customer i if it travels from i to j by recharging its battery en-route. Constraints (4.10) guarantee the time feasibility of arcs emanating from the customers (the depot). Constraints (4.11) establish the service time windows restriction. Constraints (4.10) and (4.11) also eliminate the formation of sub-tours. Constraints (4.12)-(4.15) keep track of the battery SoC at each node and make sure that it never falls below zero where $M = Q + (h + w \cdot \sum_{i \in V} q_i) \cdot \max(d_{ij})$. Constraints (4.12) establish the battery SoC consistency if the vehicle travels from customer i to customer j without recharging en-route. Constraints (4.13) determine battery SoC at the arrival at customer j if the vehicle visits a recharging station after it has departed from customer i whereas constraints (4.14) check battery SoC at the arrival at a station if the battery is recharged en-route. Constraints (4.15) set the limits for battery SoC when the vehicle departs from a station. Constraints (4.16) allow positive battery SoC at the arrival of an EV at customer j only if that EV serves customer j . Constraints (4.17) keep track of the load of the vehicle throughout its journey. Constraints (4.18) ensure the non-negativity of the load on the vehicle and guarantee that the cargo capacity is not exceeded. Finally, constraints (4.19) and (4.20) define the binary decision variables.

Model II. In order to decrease the size of the problem we eliminate the vehicle index and instead we used the dummy copies of the arrival depot in order to keep track of the battery SoC. We define AD as the set of arrival depots and accordingly we define $V_{AD} = V \cup AD$ and $V_{0,AD} = V \cup \{0\} \cup AD$. Then, the problem can be represented on a complete directed

graph $G = (N, A)$ with the set of arcs $A = \{(i, j) | i, j \in N, i \neq j\}$, where $N = V_{0,AD} \cup F$ is the total set of nodes on the network.

$$\text{Min } (y_0 - y_{AD}) + \sum_{i \in V_0} \sum_{j \in V_{AD}} \sum_{s \in F} (Y_{ijs} - y_{ijs}) \quad (4.21)$$

subject to

$$y_0 = Q \quad (4.22)$$

$$\sum_{\substack{j \in V_{AD} \\ j \neq i}} x_{ij} = 1 \quad \forall i \in V \quad (4.23)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} = 1 \quad \forall j \in V_{AD} \quad (4.24)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V_{AD} \\ i \neq j}} x_{ji} = 0 \quad \forall j \in V \quad (4.25)$$

$$\sum_{s \in F} z_{ijs} \leq x_{ij} \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (4.26)$$

$$\tau_i + (t_{ij} + r_i)x_{ij} + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs} + g(Y_{ijs} - y_{ijs})) - l_0(1 - x_{ij}) \leq \tau_j \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (4.27)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (4.28)$$

$$0 \leq y_j \leq y_i - (h + wu_i)d_{ij} + M(1 - x_{ij} + \sum_{s \in F} z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (4.29)$$

$$y_j \leq Y_{ijs} - (h + wu_i)d_{sj} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (4.30)$$

$$0 \leq y_{ijs} \leq y_i - (h + wu_i)d_{is} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (4.31)$$

$$y_{ijs} \leq Y_{ijs} \leq Qz_{ijs} \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (4.32)$$

$$y_j \leq Q \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} \quad \forall j \in V_{AD} \quad (4.33)$$

$$u_j \geq u_i + q_j x_{ij} - C(1 - x_{ij}) \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (4.34)$$

$$0 \leq u_i \leq C \quad \forall i \in V_{0,AD} \quad (4.35)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V_0, j \in V_{AD}, i \neq j \quad (4.36)$$

$$z_{ijs} \in \{0,1\} \quad \forall i \in V_0, j \in V_{AD}, s \in F, i \neq j \quad (4.37)$$

The objective function and constraints are similar to the model I; however, the vehicle index is removed and instead a set of arrival depots are added to the model where each copy of arrival depot is corresponding to a vehicle.

Model III. The vehicle index is eliminated, but we did not use the dummy copies of the arrival depot. A mixed-integer nonlinear programming model is used where the objective function is nonlinear, and it minimizes the energy consumption where it is calculated by the multiplication of energy consumption rate per unit distance times traveled distance. In the load-dependent case the energy consumption rate per unit distance is related to the amount of cargo carried on that arc which is a decision variable. This causes nonlinearity in the objective function. The constraints are similar to the previous models.

$$\text{Min } \sum_{i \in V_0} \sum_{j \in V_{n+1}} ((h_{ij} + wu_i d_{ij})x_{ij} + \sum_{s \in F} (\hat{h}_{ijs} + wu_i \hat{d}_{ijs})z_{ijs}) \quad (4.38)$$

subject to

$$y_0 = Q \quad (4.39)$$

$$\sum_{\substack{j \in V_{n+1} \\ j \neq i}} x_{ij} = 1 \quad \forall i \in V \quad (4.40)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V_{n+1} \\ i \neq j}} x_{ji} = 0 \quad \forall j \in V \quad (4.41)$$

$$\sum_{s \in F} z_{ijs} \leq x_{ij} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.42)$$

$$\begin{aligned} \tau_i + (t_{ij} + r_i)x_{ij} + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs} + g(Y_{ijs} - y_{ijs})) \\ - l_0(1 - x_{ij}) \leq \tau_j \end{aligned} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.43)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (4.44)$$

$$0 \leq y_j \leq y_i - (h + wu_i)d_{ij} + M(1 - x_{ij} + \sum_{s \in F} z_{ijs}) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.45)$$

$$y_j \leq Y_{ijs} - (h + wu_i)d_{sj} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (4.46)$$

$$0 \leq y_{ijs} \leq y_i - (h + wu_i)d_{is} + M(1 - z_{ijs}) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (4.47)$$

$$y_{ijs} \leq Y_{ijs} \leq Qz_{ijs} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (4.48)$$

$$y_j \leq Q \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij} \quad \forall j \in V_{n+1} \quad (4.49)$$

$$u_j \geq u_i + q_j x_{ij} - C(1 - x_{ij}) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.50)$$

$$0 \leq u_i \leq C \quad \forall i \in V_{0,n+1} \quad (4.51)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (4.52)$$

$$z_{ijs} \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, i \neq j \quad (4.53)$$

In order to check whether the objective function of model III is convex or concave, we considered a small instance where there are only a depot and one customer. The objective function can be written as Eq. (4.54):

$$\text{Min } (h_{0,1} + wu_0 d_{0,1})x_{0,1} + (h_{1,0} + wu_1 d_{1,0})x_{1,0} \quad (4.54)$$

note that $d_{0,1} = d_{1,0}$ and $h_{0,1} = h_{1,0}$ since the network is symmetric. For the sake of simplicity, we will use d and h instead of them. The Hessian matrix comprises geometric information about the surface $z = f(x, y)$. The Hessian matrix of $z = f(x, y)$ is defined as

$$H_f(x, y) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$$

at any point where all of the second partial derivatives of function f exist. Let's matrix H shows the Hessian matrix of the function Eq. (4.54) as:

$$H = \begin{matrix} & \begin{matrix} x_{0,1} & x_{1,0} & u_0 & u_1 \end{matrix} \\ \begin{matrix} x_{0,1} \\ x_{1,0} \\ u_0 \\ u_1 \end{matrix} & \begin{bmatrix} 0 & 0 & wd & 0 \\ 0 & 0 & 0 & wd \\ wd & 0 & 0 & 0 \\ 0 & wd & 0 & 0 \end{bmatrix} \end{matrix}$$

We need to calculate the Eigenvalues of the Hessian matrix to obtain geometric information about the surface of the function. If all the eigenvalues are positive the function will be convex. If all of them be negative, the function will be concave. Otherwise, if some of them are positive and some negative signs, the function will be neither convex nor concave and it will have a saddle point. In order to calculate the

eigenvalues, we should compute $\det(H - I\lambda) = 0$ where the λ values show the eigenvalues.

$$\det\left(\begin{bmatrix} 0 & 0 & wd & 0 \\ 0 & 0 & 0 & wd \\ wd & 0 & 0 & 0 \\ 0 & wd & 0 & 0 \end{bmatrix} - \begin{bmatrix} -\lambda & 0 & 0 & 0 \\ 0 & -\lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & -\lambda \end{bmatrix}\right) = \det\left(\begin{bmatrix} -\lambda & 0 & wd & 0 \\ 0 & -\lambda & 0 & wd \\ wd & 0 & -\lambda & 0 \\ 0 & wd & 0 & -\lambda \end{bmatrix}\right) \\ = (\lambda^2 - (wd)^2)^2 = 0 \quad \rightarrow \quad \lambda = \pm wd.$$

Since w and d are positive thus λ values can be positive or negative. Therefore, the eigenvalues are not totally positive or negative and the function is neither convex nor concave. Consequently, the solution obtained from this model is a local minimum for the original model.

4.3. Large Neighborhood Search algorithm

We attempt to solve small-size instances using a commercial solver. To solve the large-size instances, we develop an LNS method. LNS was introduced by Shaw (1998) and aims at improving an initial solution by using several destroy and repair mechanisms iteratively. In each iteration, some customers are removed from the solution and reinserted into the routes to create a new feasible solution. This procedure is repeated for a predetermined number of iterations. LNS and Adaptive LNS (ALNS) have been successfully applied to many VRP variants including EVRPs and EVRPTWs (Goeke and Schneider, 2015; Keskin and Çatay, 2016; Wen et al., 2016; Hiermann et al., 2016; Schiffer and Walther, 2017; Schiffer et al., 2018; Keskin and Çatay, 2018; Keskin et al., 2019).

We create the initial solution using the insertion heuristic of Keskin and Çatay (2016) where the cost of inserting a customer into a route is calculated as $(h + wu_i)d_{ik} + (h + wu_k)d_{kj} - (h + wu_i)d_{ij}$. This insertion cost is calculated for all unvisited customers and the minimum cost insertion is performed by ensuring that the related constraints are not violated. If an EV runs out energy, a station may be inserted to make its tour energy feasible. We use First-Feasible Station Insertion algorithm which will be

described in Section 4.3.2. If no customer can be feasibly inserted in the route, a new route is initialized, and the procedure is repeated until all customers are served.

Our LNS consists of customer destroy and repair mechanisms. In each iteration, a customer removal algorithm is applied to a feasible solution to remove a subset of customers from the routes. If any station is no longer needed in the partial solution, they are removed as well. Next, we apply a customer insertion algorithm that inserts all the customers removed to repair the solution in an attempt to obtain a new improved solution. Stations may be inserted to maintain the energy feasibility along the route. This procedure continues until the stopping criterion is satisfied, which is a limit on the number of iterations in our implementation. Note that the set of stations that can be visited between any two customers is reduced by using the dominance rules presented in Bruglieri et al. (2016).

4.3.1. Destroy operators

The current feasible solution is destroyed by removing γ customers. We use Worst-Consumption, Random Worst-Consumption, Shaw, Random Worst-Time, Random, Random Route Removal and Greedy Route Removal procedures of Keskin and Çatay, (2016) by modifying them for the load-dependent problem. The destroy operators are selected randomly.

- Worst-Consumption algorithm selects the customers with high energy consumption imposed on the route by visiting that customer, which is calculated as $(h + wu_i)d_{ik} + (h + wu_k)d_{kj} - (h + wu_i)d_{ij}$ that considers distance and cargo load effect in energy consumption.
- Random Worst-Consumption sorts the customers with respect to the associated energy consumptions, considers a subset of $\sigma \times \gamma$ customers with the highest costs to select γ customers randomly and remove them.
- Shaw Removal removes similar customers with respect to their energy consumption, earliest service time, being in the same route, and their demand. It randomly selects customer i and calculates the relatedness measure as $R_{ij} = \phi_1 h_i d_{ij} + \phi_2 |e_i - e_j| +$

$\phi_3 l_{ij} + \phi_4 |D_i - D_j|$ to find similar customers j . $\phi_1 - \phi_4$ are the Shaw parameters, $l_{ij} = -1$ if i and j are in the same route, 1 otherwise. Small R_{ij} shows high similarity. So, using the non-decreasing order of the relatedness value with customer i , γ customers are removed from the solution.

- Random Worst-Time algorithm is a version of Shaw Removal where ϕ_1, ϕ_3, ϕ_4 are set equal to 0. The customers are sorted in the non-decreasing order of their relatedness values and γ customers are randomly removed from the subset of $\sigma \times \gamma$ customers with the lowest relatedness values.
- Random Removal mechanism randomly removes γ customers from the solution.
- Random Route Removal algorithm randomly removes ω routes from the solution.
- Greedy Route Removal mechanism sorts the routes in the non-decreasing order of the number of customers visited and removes ω routes which serve the least number of customers.

Note that the Route Removal algorithms attempt to reduce the fleet size.

4.3.2. Repair operators

We adapt Random Greedy, Regret-2, Random Time-Based, Random Greedy with Noise Function, and Regret-2 with Noise Function repair algorithms of Keskin and Çatay (2016) and Demir et al. (2012) for our load-dependent case. In addition, we propose Exhaustive Greedy, Exhaustive Time-Based, Exhaustive Time-Based with Noise Function, and Random Time-Based with Noise Function mechanisms. The repair operators are selected randomly.

- Random Greedy Insertion selects a customer and inserts it in the best position which leads to least increase in energy consumption.
- Regret-2 Insertion tries to avoid the higher costs in the subsequent iteration. It calculates the difference between the cost of the best insertion and the second-best insertion for all customers and selects the customer with the highest difference.
- Random Time-Based Insertion calculates insertion costs similar to the Exhaustive Time-Based algorithm, however, at first an unassigned customer is selected randomly, and the algorithm inserts it in its best position.

- Random Greedy Insertion with Noise Function is an extension of the Random Greedy Insertion mechanism with a degree of freedom. We use the same noise function presented in Demir et al. (2012). The cost of insertion using the freedom degree is calculated as $NewCost = ActualCost + \bar{d}\mu\epsilon$, where $\bar{d}$ represents the maximum distance in the network, the noise parameter used for diversification is shown by μ , and ϵ is a random number between $[-1, 1]$.
- Exhaustive Greedy Insertion considers all possible insertion positions for all not-inserted customers and selects the customer-position matching which leads to least increase of energy consumption.
- Exhaustive Time-Based Insertion calculates the difference between the route duration after and before inserting a customer as the insertion cost. For all customers, the insertion costs in all possible positions are calculated and the customer with least the insertion cost is selected.

Note that Regret-2 with Noise Function, Exhaustive Time-Based with Noise Function, Random Time-Based with Noise Function are extensions of Regret-2, Exhaustive Time-Based and Random Time-Based insertion mechanisms, respectively, using a similar noise function.

As we mentioned earlier, the unnecessary stations are removed from the partial solution obtained using the destroy operator. During the repair procedure, the insertion of a customer may not be feasible with respect to battery SoC. In that case, we first attempt to increase the recharge quantity if a station is visited prior to arriving at that customer. If the energy recharged at the station cannot be increased or no station is visited en-route we apply a station insertion operator to make the insertion feasible. We modified Best Station Insertion from the literature (Keskin and Çatay, 2016) and Multiple Station Insertion operators introduced in section 2.3 and applied them for the load-dependent problem. Also, we develop First Feasible Station Insertion operator for this problem. Note that at most one station can be inserted between two consecutive customers in a route.

- First-Feasible Station Insertion considers the first customer (or depot) where the vehicle arrives at with negative SoC and checks the insertion of a station in the preceding arcs backward. The first station which makes the problem feasible is inserted.

- Best-Station Insertion algorithm checks all possible stations in all possible arcs before the first customer (or depot) with negative SoC and inserts the best station in its best position.
- Multiple-Station Insertion algorithm inserts multiple stations into a route when the insertion of a single station cannot make the route feasible. A station is inserted on the arc traversed immediately before arriving at the customer (or depot) with a negative SoC where the vehicle is recharged up to the maximum level allowed by the battery capacity and time windows restrictions of the succeeding customers. If the SoC is still negative at that customer or if the vehicle runs out of energy before reaching the inserted station, we attempt to insert another station prior to the last customer visited before traveling to the recently inserted station. This procedure is repeated until the route becomes energy feasible.

One of the First-Feasible Station Insertion and Best-Station Insertion algorithms is selected randomly. If it does not a feasible route, we resort to Multiple-Station Insertion algorithm. Note that, we remove all stations in the solution after every β iterations and use Best-Station Insertion algorithm to insert stations to obtain an improved feasible solution.

4.3.3. Repair-opt operator

We introduce a repair operator that attempts to insert the removed customers in the partial solution optimally along with the recharging stations, if needed. We use Gurobi solver to solve the mathematical model to reinsert removed customers and stations in their best places in the given partial routes in order to obtain a better solution. We call this repair operator in the algorithm after performing η iterations because it is expensive in terms of computational time and is more promising after the algorithm converges to a good solution. Furthermore, as the LNS algorithm progresses it provides tight upper bounds on the number of vehicles, which accelerates Gurobi solver's run time. The performance of the proposed model is investigated in section 4.4. The parameters for the LNS are illustrated in Appendix D.

4.4. Experimental study

We performed our computational tests using the dataset of Schneider et al. (2014) and Desaulniers et al. (2016) for the small-size and large-size instances, respectively. The small-size dataset consists of 36 instances involving 5, 10, and 15 customers, and the large-size dataset includes 56 instances generated based on the VRPTW instances of Solomon (1987). The instances are classified according to the geographic distribution of the customers: clustered (c-type), random (r-type), and half clustered half random (rc-type). Furthermore, in type-1 problems (i.e., subsets r1, c1, rc1) the planning horizon is shorter, and customers' time windows are narrower compared to type-2 problems (i.e., r2, c2, rc2). In our study, we only consider type-1 problems from the large-size dataset as they better exhibit the influence of recharging decisions on route planning (Desaulniers et al., 2016; Keskin and Çatay, 2018). In order to deal with realistic vehicle cargo capacity and customer demands, we assumed an electric truck based on the specifications provided in Demir et al. (2012). Since the capacity of this vehicle is 3650 kg, we converted the demand quantities to reasonable weights by multiplying each by $(3650 / \text{original capacity})$ in order to observe the effect of cargo weight on energy consumption. We assumed a drive train efficiency of 0.9 as EVs are more efficient than internal combustion engine vehicles. Furthermore, since the EVs in the original data are assumed to consume one unit of energy per unit distance/time traveled, we used Eq. (4.4) to calculate the actual energy consumption of an empty vehicle (i.e., 6350 kg) per unit distance and scaled it to $h = 1$. We used the same approach to determine the energy consumption w associated with unit load carried. We consider a flat network where road gradients are zero and we neglected vehicle acceleration.

The small-size instances were solved using GUROBI 9.0 with a 2-hour time limit. LNS was employed to solve both small- and large-size instances. LNS was coded in Python 3.7.1 and all runs were performed on an Intel Core(TM) i7-8700 processor with 3.20 GHz speed and 32 GB RAM. We performed five runs for each instance. The number of LNS iterations is set to 1000 for the small-size instances and 6000 for the large-size instances.

4.4.1. Analysis on the performance of the proposed models

In this section, we analyzed the performance of the models introduced in section 4.2.2. Table 4.1 presents the results obtained from models I, II and III for the load-dependent case. “#Veh”, “EC”, and “t” refer to the fleet size, energy consumption, and run time (in seconds), respectively. Rows “Avg.” present the average values for the number of vehicles, energy consumption (for the instances with the same number of vehicles), and the run time for the instances with 5, 10, and 15 customers. The total averages for all the instances are shown in the row “Total Avg.”.

Table 4.1 Comparison of results obtained using different models for the load-dependent case

Instance	Model I			Model II			Model III		
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	#Veh	EC	t (sec)
r104c5-s3	2	141.54	<1	2	141.54	<1	2	141.54	<1
r105c5-s3	2	159.23	<1	2	159.23	<1	2	159.23	<1
r202c5-s3	1	144.12	<1	1	144.12	<1	1	144.12	<1
r203c5-s4	1	181.32	<1	1	181.32	<1	1	181.32	<1
c101c5-s3	2	266.02	<1	2	266.02	<1	2	266.02	<1
c103c5-s2	1	186.83	<1	1	186.83	<1	1	186.83	<1
c206c5-s4	1	250.63	<1	1	250.63	<1	1	250.63	<1
c208c5-s3	1	168.91	<1	1	168.91	<1	1	168.91	<1
rc105c5-s4	2	256.62	<1	2	256.62	<1	2	256.62	<1
rc108c5-s4	2	264.00	<1	2	264.00	<1	2	264.00	<1
rc204c5-s4	1	188.59	<1	1	188.59	<1	1	188.59	<1
rc208c5-s3	1	170.82	<1	1	170.82	<1	1	170.82	<1
Avg.	1.42	198.22	0.33	1.42	198.22	0.25	1.42	198.22	0.04
r102c10-s4	3	336.00	37	3	336.00	4	3	336.00	<1
r103c10-s3	2	220.50	1507	2	220.50	213	2	220.50	<1
r201c10-s4	1	262.40	6	1	262.40	4	1	262.40	<1
r203c10-s5	1	227.26	7200	1	227.26	3297	1	227.26	<1
c101c10-s5	3	410.10	117	3	410.10	11	3	410.10	<1
c104c10-s4	2	308.81	7200	2	305.64	7200	2	305.64	1
c202c10-s5	1	319.04	8	1	319.04	<1	1	319.04	<1
c205c10-s3	2	233.74	148	2	233.74	18	2	233.74	<1
rc102c10-s4	5	475.00	435	5	475.00	54	5	475.00	<1
rc108c10-s4	3	364.62	4472	3	364.62	46	3	364.62	<1
rc201c10-s4	1	424.49	<1	1	424.49	<1	1	424.49	<1
rc205c10-s4	2	334.54	432	2	334.54	103	2	334.54	<1
Avg.	2.17	326.38	1796.85	2.17	326.11	912.57	2.17	326.11	0.92
r102c15-s8	5	430.83	7200	5	430.83	7200	5	430.83	<1
r105c15-s6	4	350.09	7200	4	350.09	7200	4	350.09	<1
r202c15-s6	2	365.06	7200	2	365.06	7200	1	590.31	7200
r209c15-s5	1	361.83	7200	1	347.31	7200	1	347.31	145
c103c15-s5	4	393.15	7200	3	403.18	7200	3	401.97	27
c106c15-s3	3	370.84	7200	3	352.29	2478	3	352.29	4
c202c15-s5	2	408.12	7200	2	393.19	7200	2	393.19	2
c208c15-s4	2	310.12	7200	2	310.12	7200	2	310.12	<1
rc103c15-s5	5	415.62	7200	4	415.79	7200	4	415.79	4
rc108c15-s5	5	453.49	7200	4	430.69	7200	3	417.92	72
rc202c15-s5	2	403.03	7200	2	403.03	7200	2	403.03	1
rc204c15-s7	1	444.03	7200	1	402.41	7200	1	402.15	7200
Avg.	3.00	384.86	7200.00	2.75	373.66	6806.50	2.58	373.63	1221.41
Total Avg.	2.19	303.15	2999.06	2.11	299.33	2573.11	2.06	299.32	407.46

The results show that the model II overperforms model I in two instances with respect to the number of vehicles and in five instances regarding energy consumption. On the other hand, the computational time for model II is 14.2% less than the model I on the average. Model III obtains solutions with a smaller number of vehicles in two instances and improves two instances with respect to the energy consumption compared to the model II. The computational time for model III is about 84.2 % less than model II on the average.

Table 4.2 Results for small-size instances obtained using GUROBI and LNS for Load Independent and Load-dependent cases

Instance	GUROBI							LNS		
	Load Independent			Load Dependent				Load Dependent		
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	% Δ EC	#Veh	EC	t (sec)
r104c5-s3	2	136.69	<1	2	141.54	<1	3.55	2	141.54	1
r105c5-s3	2	156.08	<1	2	159.23	<1	2.01	2	159.23	1
r202c5-s3	1	128.88	<1	1	144.12	<1	11.82	1	144.12	2
r203c5-s4	1	179.06	<1	1	181.32	<1	1.27	1	181.32	2
c101c5-s3	2	257.75	<1	2	266.02	<1	3.21	2	266.02	1
c103c5-s2	1	175.37	<1	1	186.83	<1	6.54	1	186.83	1
c206c5-s4	1	242.56	<1	1	250.63	<1	3.33	1	250.63	2
c208c5-s3	1	164.34	<1	1	168.91	<1	2.78	1	168.91	2
rc105c5-s4	2	233.77	<1	2	256.62	<1	9.78	2	256.62	1
rc108c5-s4	2	253.93	<1	2	264.00	<1	3.97	2	264.00	1
rc204c5-s4	1	185.16	<1	1	188.59	<1	1.85	1	188.59	3
rc208c5-s3	1	167.98	<1	1	170.82	<1	1.69	1	170.82	2
Avg.	1.42	190.13	0.02	1.42	198.22	0.04	4.32	1.42	198.22	1.57
r102c10-s4	3	249.19	<1	3	336.00	<1	34.84	3	336.00	3
r103c10-s3	2	206.30	1	2	220.50	<1	6.88	2	220.50	14
r201c10-s4	1	241.51	<1	1	262.40	<1	8.65	1	262.40	2
r203c10-s5	1	222.64	<1	1	227.26	<1	2.08	1	227.26	8
c101c10-s5	3	388.25	<1	3	410.10	<1	5.63	3	410.10	2
c104c10-s4	2	273.93	<1	2	305.64	1	11.58	2	305.64	85
c202c10-s5	1	304.06	<1	1	319.04	<1	4.93	1	319.04	1
c205c10-s3	2	228.28	<1	2	233.74	<1	2.39	2	233.74	3
rc102c10-s4	4	424.00	<1	5	475.00	<1	—	5	475.00	2
rc108c10-s4	3	347.90	<1	3	364.62	<1	4.81	3	364.62	4
rc201c10-s4	1	412.86	<1	1	424.49	<1	2.82	1	424.49	3
rc205c10-s4	2	325.98	<1	2	334.54	<1	2.63	2	334.54	4
Avg.	2.08	290.99	0.23	2.17	312.58	0.92	7.93	2.17	312.58	10.94
r102c15-s8	5	412.78	1	5	430.83	<1	4.37	5	430.83	5
r105c15-s6	4	336.15	<1	4	350.09	<1	4.15	4	350.09	4
r202c15-s6	1	507.32	1152	1	590.31	7200	16.36	2	365.06	272
r209c15-s5	1	313.24	10	1	347.31	145	10.88	1	347.31	36
c103c15-s5	3	348.46	1	3	401.97	27	15.36	3	401.97	419
c106c15-s3	3	275.13	<1	3	352.29	4	28.04	3	352.29	12
c202c15-s5	2	383.62	2	2	393.19	2	2.50	2	393.19	33
c208c15-s4	2	300.55	<1	2	310.12	<1	3.18	2	310.12	15
rc103c15-s5	4	397.67	2	4	415.79	4	4.56	4	415.79	57
rc108c15-s5	3	370.25	6	3	417.92	72	12.88	3	417.92	20
rc202c15-s5	2	394.39	<1	2	403.03	1	2.19	2	403.03	11
rc204c15-s7	1	382.22	7200	1	402.15	7200	5.21	1	402.15	371
Avg.	2.58	355.86	697.90	2.58	384.06	1221.41	9.14	2.67	384.06	104.63
Total Avg.	2.03	278.99	232.72	2.38	298.29	407.46	7.13	2.42	298.29	39.05

Although model III does not guarantee to obtain the global opima, the results show that

it can find a better solution with respect to the number of vehicles and energy consumption with less computational run times compared to the other models, that is why we can use it in the repair-opt operator of the LNS algorithm.

4.4.2. Analysis on the effect of load on the route planning

The results for small-size instances are provided in Table 4.2 which also shows the performance of LNS with respect to Gurobi. Column “GUROBI” shows the results using GUROBI and “LNS” provides the results obtained by the proposed LNS algorithm. Column “Load Independent” reports the results for the case that does not take into account the increased energy consumption associated with the cargo carried whereas column “Load Dependent” shows the results for the case that considers the load on the vehicle.

Table 4.3 Result for large-size instances obtained using LNS for Load-Independent and Load-Dependent cases

Instance	Load Independent			Load Dependent		
	#Veh	EC	t (sec)	#Veh	EC	t (sec)
r101	17	1679.69	782	18	1741.27	712
r102	16	1607.23	1761	17	1630.97	1135
r103	14	1277.51	2200	14	1404.57	1885
r104	12	1162.21	2776	13	1247.14	2545
r105	15	1348.54	1608	16	1482.75	1324
r106	14	1404.93	1885	15	1469.73	1752
r107	12	1243.62	2057	13	1332.27	1774
r108	12	1091.86	3085	12	1267.00	2571
r109	14	1261.83	1854	14	1415.50	1734
r110	13	1146.24	2656	13	1251.46	2129
r111	12	1193.67	2567	13	1262.93	2029
r112	12	1105.43	3616	12	1218.47	2763
Avg.	13.58	1176.57	2237.23	14.08	1311.40	1862.85
c101	12	1043.38	682	12	1186.46	718
c102	11	1100.17	2252	12	1173.36	1969
c103	11	1246.22	3019	11	1395.65	3005
c104	11	1142.82	5640	11	1373.41	4524
c105	11	1037.26	1617	12	1161.57	1345
c106	11	1040.79	1896	12	1166.90	1919
c107	11	1021.93	2142	12	1179.50	1634
c108	12	1096.64	2570	12	1206.47	2466
c109	11	1001.64	3667	11	1318.41	2648
Avg.	11.22	1106.14	2609.59	11.67	1296.08	2247.49
rc101	15	1652.33	1733	17	1921.03	1119
rc102	16	1607.69	1517	16	1815.88	1555
rc103	14	1501.91	1884	14	1722.56	1773
rc104	12	1380.05	2496	13	1515.47	2930
rc105	15	1539.88	2015	15	1734.78	1654
rc106	14	1484.98	1930	15	1650.57	1557
rc107	12	1264.11	2314	13	1467.85	2060
rc108	12	1398.05	2815	13	1533.09	2240
Avg.	13.75	1549.83	2087.81	14.50	1757.74	1861.02
Total Avg.	12.85	1277.51	2311.54	13.42	1455.07	1990.45

The comparison of these two columns exhibits the influence of the load on routing decisions. “ t ” and “ $\% \Delta EC$ ” refers to the average run time (in seconds) and the percentage increase in the energy consumption comparing load-independent and load-dependent case, respectively. The results show that #Veh increases by one in one instance (rc102c10-s4). Furthermore, we observe that EC values obtained in the load-independent case are far from the actual energy consumption found by taking into account the cargo load. Finally, we see that LNS finds optimal solutions (or equal to the upper bound obtained from Gurobi) in 35 out of 36 instances. In one instance (r202c15-s6) it finds a solution with a higher number of vehicles which is equal to the upper bound gained from models I and II.

We solved the large-size instances for both load-independent and load-dependent cases using LNS. The results are provided in Table 4.3 **Error! Reference source not found..** We observe that the number of vehicles increases by two in one instance and by one in fifteen instances (shown in bold) in the load-dependent case compared to the load-independent. Furthermore, in the remaining thirteen instances, EC increases by 13.90% on the average. These results show the importance of considering cargo weight in route optimization.

4.4.3. Analysis on the repair-opt operator

In this section, we investigate the performance of the repair-opt operator on the performance of the proposed LNS algorithm for the load-dependent problem. Note that we performed the LNS algorithm for 15000 and 25000 iterations for solving the small and large-size instances, respectively, when it does not include the repair-opt mechanism. We performed our tests on the 5-, 10-, 15-, and 100-customer instances. The results are summarized in Table 4.4. “#Inst”, “#Veh”, “EC”, “ t (sec)” denote the number of instances solved, the average number of vehicles, average energy consumption and average computational time, respectively. “#Bet.Veh” and “#Bet.EC” refer the number of instances where LNS with repair-opt operator obtains a better solution with regard to the number of vehicles and energy consumption, respectively. The results show that the repair-opt operator allowed LNS to find better solutions in 8 small-size instances out of

36; in two instances it reduces the fleet size whereas in six instances it decreased the total energy consumption. In the large-size instances, the LNS equipped with the repair-opt operator achieves reduced fleet size in three instances and provides savings in energy consumption in sixteen instances. The detailed results are reported in Appendix E. These results show the contribution of the repair-opt operator in the overall performance of the LNS algorithm.

Table 4.4 Average results for solving small and large-size instances without and with considering repair-opt operator in LNS algorithm

Instance	#Inst	Without Repair-opt			With Repair-opt				
		#Veh	EC	t (sec)	#Veh	EC	t (sec)	#Bet.Veh	#Bet.EC
5-Customer	12	1.42	198.33	10.83	1.42	198.33	1.57	0	0
10-Customer	12	2.33	318.30	22.08	2.17	315.30	10.94	2	4
15-Customer	12	2.67	390.75	43.50	2.67	382.44	104.63	0	2
100-Customer-R	12	14.25	1386.98	2325.83	14.08	1371.58	1862.85	2	6
100-Customer-C	9	11.78	1254.27	2695.85	11.67	1230.42	2247.49	1	8
100-Customer-RC	8	14.50	1664.60	2186.83	14.50	1670.15	1861.02	0	2

4.5. Conclusions and future directions

In this chapter, we addressed EVRPTW with partial recharge by taking into account the energy consumption associated with the cargo carried on the vehicle. We formulated three different 0-1 mixed-integer programming models of the problem and compared their performances on small-size instances. For solving the large-size instances, we developed an LNS method by proposing new destroy and repair operators as well as a new repair mechanism based on an exact method. Our computational tests showed how the fleet size and/or energy consumption increase in comparison to the case where the load factor is neglected and revealed the importance of considering the cargo weight of the vehicles for more accurate route planning. Our experiments also showed the contribution of the proposed exact repair method on the performance of the LNS algorithm.

5. Electric Vehicle Routing Problem with Time Windows considering Cargo Weight, Road Gradient and Regenerative Braking

5.1. Introduction to Electric Vehicle Routing Problem with Time Windows and Effect of Road Gradient on Energy Consumption

Studying different factors which affect energy consumption is often neglected in the VRP literature since it may bring more complexity to the problem or it may not an important issue as the ICEs does not need to refuel frequently and the refueling process is pretty fast. Nevertheless, because of the limited battery capacity in EVs which restricts their driving ranges, and recharging process, which is time-consuming, considering the factors which affect energy consumption in the EVRP literature becomes vital. Energy consumption on the road does not only depend on the distance traveled but many other factors including the vehicle's weight, velocity, auxiliary equipment (internal factors) as well as ambient temperature and road gradient (external factors). Among them, the road gradient and the weight of the transported cargo may significantly influence the routing decisions. In the reality the road network of the cities where the logistic companies operate are not flat. Traveling on an arc with a positive road gradient requires more energy comparing to an arc in a flat network. As the gradient of an arc slightly rises, the energy consumption per unit distance increases which can extremely increase the energy consumption on that arc. Furthermore, in the operations where the EVs deal with heavy loads, the effect of road gradient on the energy consumption intensifies since an EV moving uphill with heavy load requires more energy in order to finish its journey. On the other hand, if an EV traverses on an arc with a negative road gradient where the driver needs to push the brake pedal in order to travel with a constant speed, energy can be saved on the battery because of the regenerative braking technology (Clegg, 1996, Xu et al.,

2011, Zhang et al., 2008, Wu et al., 2015). The companies dealing with heavy loads which are operating in the cities that are not flat get affected more and are examples for which a model considering cargo weight and road gradient network produces more effective transportation strategies in comparison with basic routing mode, which is the main motivation of this study.

In this chapter, we address an EVRPTW with partial recharges by taking into account the energy consumption associated with the road gradient, regenerative braking, and the cargo carried on the vehicle. We assume that the road gradient between two nodes is the average gradient in that road. We adopt a hierarchical objective function where the primary objective is to minimize the fleet size whereas the secondary objective is to minimize total energy consumption. We develop a Large Neighbourhood Search (LNS) algorithm to solve the problem. The remainder of this chapter is organized as follows: section 5.2 introduces the problem and formulates its mathematical programming model. Section 5.3 presents the experimental study and discusses the results. Finally, concluding remarks are provided in section 5.4.

5.2. Problem description

A homogeneous fleet of EVs serves a set of customers with known demands, time windows, and service times in the context of EVRPTW. Contrasted with former studies in the literature that assume a constant energy consumption proportional to the distance traveled, we consider the additional energy consumption related to road gradient, regenerative braking, and freight load. Carrying more load by an EV on the roads with positive road gradient causes more energy consumption while if the vehicle travels on the arcs with negative road gradient where the driver needs to push the brake pedal to keep the vehicle speed constant, the energy released from braking can be saved in the battery. When an EV regenerate energy in negative road gradient arcs, not also it does not consume energy but also it saves the energy in the battery. Moreover, we allow partial recharging, and charging duration depends on the amount of energy transferred. It is a common practice in the real world to operate within the first phase of recharging where the energy transferred is a linear function of the recharge duration in order to prolong the

battery life (Pelletier et al., 2017) thus we also assume a linear charging function. Furthermore, we assume that the EV can be recharged at most once between two consecutive customers, which is practical in last-mile logistics. We consider a pick-up problem where the load of the EV increases along its tour as it visits the customers. Each EV departs from the depot with the full battery since it can be recharged overnight.

5.2.1. Energy consumption considering road gradient and regenerative braking

Considering the tractive power function in chapter 4 (Eq. 4.2), once the road gradient is positive the tractive power will be positive. However, if the road gradient is negative and $Ma + 0.5C_d\rho AV^2 + MgC_r\cos\theta \geq |Mg\sin\theta|$ the tractive power will be positive but if the road is steeper and $Ma + 0.5C_d\rho AV^2 + MgC_r\cos\theta < |Mg\sin\theta|$ the tractive power will be negative and thus the vehicle can save energy using the regenerative braking technology.

When the tractive power is positive, it can be converted to second-by-second engine power output (kW) as Eq. (4.3). On the other hand, if the tractive power is negative the second-by-second battery power output (kW) will be:

$$P = \mu_r \cdot P_{tract} + P_{acc} \quad (5.1)$$

where μ_r represents the efficiency of the regenerative braking process since the engine cannot save the whole regenerated energy in the battery. Then, the energy consumption in (kWh/km) can be calculated using Eq. (4.4). Finally, the energy consumption on an arc considering its gradient and the vehicle's cargo weight will be as follows:

$$h_{ij}(u_i) = d_{ij}(P_{tract}(u_i)/\mu_{tf})/v \quad P_{tract}(u_i) \geq 0 \text{ (kW)} \quad (5.2)$$

$$h_{ij}(u_i) = d_{ij}(\mu_r \cdot P_{tract}(u_i))/v \quad P_{tract}(u_i) < 0 \text{ (kW)} \quad (5.3)$$

where $P_{tract}(u_i)$ is the tractive power for the arc (i, j) considering its gradient and the cargo load carrying by the vehicle on that arc. $P_{tract}(u_i)$ is the function of load and it depends on the sequence of the routes. The distance between i and j is represented by d_{ij}

and consequently the total energy consumption on arc (i, j) is denoted by $h_{ij}(u_i)$ which is a function of the cargo load.

5.2.2. Problem formulation

All the sets, parameters, and decision variables are similar to the model I in section 4.2.2 except the energy consumption. The energy consumption on arc (i, j) is calculated by $h_{ij}(u_i)$ which includes the effect of road gradient and cargo load.

$$\text{Min } \sum_{k \in K} (y_0^k - y_{n+1}^k) + \sum_{i \in V_0} \sum_{j \in V_{n+1}} \sum_{k \in K} \sum_{s \in F} (Y_{ijs}^k - y_{ijs}^k) \quad (5.4)$$

subject to

$$y_0^k = Q \quad \forall k \in K \quad (5.5)$$

$$\sum_{j \in V_{n+1}} \sum_{k \in K} x_{ij}^k = 1 \quad \forall i \in V \quad (5.6)$$

$$\sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij}^k - \sum_{\substack{i \in V_{n+1} \\ i \neq j}} x_{ij}^k = 0 \quad \forall j \in V, k \in K \quad (5.7)$$

$$\sum_{s \in F} z_{ijs}^k \leq x_{ij}^k \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (5.8)$$

$$\tau_i + (t_{ij} + r_i)x_{ij}^k + \sum_{s \in F} (\hat{t}_{ijs}z_{ijs}^k + g(Y_{ijs}^k - y_{ijs}^k)) - l_0(1 - x_{ij}^k) \leq \tau_j \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (5.9)$$

$$e_j \leq \tau_j \leq l_j \quad \forall j \in N \quad (5.10)$$

$$0 \leq y_j^k \leq y_i^k - h_{ij}(u_i) + M(1 - x_{ij}^k + \sum_{s \in F} z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (5.11)$$

$$y_j^k \leq Y_{ijs}^k - h_{sj}(u_i) + M(1 - z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (5.12)$$

$$0 \leq y_{ijs}^k \leq y_i^k - h_{is}(u_i) + M(1 - z_{ijs}^k) \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (5.13)$$

$$y_{ijs}^k \leq Y_{ijs}^k \leq Qz_{ijs}^k \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (5.14)$$

$$y_j^k \leq Q \sum_{\substack{i \in V_0 \\ i \neq j}} x_{ij}^k \quad \forall j \in V_{n+1}, k \in K \quad (5.15)$$

$$u_j \geq u_i + q_j \sum_{k \in K} x_{ij}^k - C(1 - \sum_{k \in K} x_{ij}^k) \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \quad (5.16)$$

$$0 \leq u_i \leq C \quad \forall i \in V_{0,n+1} \quad (5.17)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, k \in K, i \neq j \quad (5.18)$$

$$z_{ijs}^k \in \{0,1\} \quad \forall i \in V_0, j \in V_{n+1}, s \in F, k \in K, i \neq j \quad (5.19)$$

The objective function (5.4) minimizes the total energy consumption. Constraints (5.5) set the initial battery SoC of EVs at departure to full. The connectivity of customer visits is imposed by constraints (5.6) whereas the flow conservation at each vertex is ensured by constraints (5.7). Constraints (5.8) make sure that vehicle k serves customer j after customer i if it travels from i to j by recharging its battery en-route. Constraints (5.9) guarantee the time feasibility of arcs emanating from the customers (the depot). Constraints (5.10) establish the service time windows restriction. Constraints (5.9) and (5.10) also eliminate the formation of sub-tours. Constraints (5.11)-(5.14) keep track of the battery SoC at each node and make sure that it never falls below zero where $M = Q + \max(h_{ij}(u_i))$. Constraints (5.11) establish the battery SoC consistency if the vehicle travels from customer i to customer j by recharging en-route. Constraints (5.12) determine battery SoC at the arrival at customer j if the vehicle visits a recharging station after it has departed from customer i whereas constraints (5.13) check battery SoC at the arrival at a station if the battery is recharged en-route. Constraints (5.14) set the limits for battery SoC when the vehicle departs from a station. Constraints (5.15) allow positive battery SoC at the arrival of an EV at customer j only if that EV serves customer j . Constraints (5.16) keep track of the load of the vehicle throughout its journey. Constraints (5.17) ensure the non-negativity of the load on the vehicle and guarantee that the cargo capacity is not exceeded. Finally, constraints (5.18) and (5.19) define the binary decision variables.

Constraints (5.11)-(5.13) are nonlinear since $h_{ij}(u_i)$ is a function of the load as well as efficiency factors that should be decided. If the tractive power is positive, Eq. (5.2) will be used and if it is negative Eq. (5.3) will be used.

5.3. Experimental design and computational study

We modified the proposed LNS in section 4.3 to solve small and large-size instances. It contains all the operators except the repair-opt operator since the constraints are nonlinear because of considering load and road gradient and it is not possible to solve it using a

commercial solver. The initial solution method, destroy, and repair operators are the same, but we modified the energy consumption on each arc which is calculated by equations Eq. (5.2) and (5.3).

We generated data based on Schneider et al. (2014) and Desaulniers et al. (2016) datasets to perform the computational tests for small-size and large-size instances, respectively. We assumed a drive train efficiency of 0.9 as EVs are more efficient than internal combustion engine vehicles. We used the same vehicle specifications as section 4. The regenerative braking efficiency is assumed to be 0.8 which means that 80% of the regenerated energy in the braking process can be saved in the battery. Furthermore, since the EVs in the original data are assumed to consume one unit of energy per unit distance/time traveled, we used Eq. (4.4) to calculate the actual energy consumption of an empty vehicle (i.e., 6350 kg) per unit distance on a flat network and scaled it to $h = 1$. We calculate the energy consumption considering road gradient and cargo load weight and scaled it correspondingly. We assumed that the vehicles move at a constant speed and we neglected vehicle acceleration.

LNS was employed to solve both small- and large-size instances. LNS was coded in Python 3.7.1 and all runs were performed on an Intel Core(TM) i7-8700 processor with 3.20 GHz speed and 32 GB RAM. We performed five runs for each instance. The number of LNS iterations is set to 15000 for the small-size instances and 25000 for the large-size.

5.3.1. Data Generation

With the purpose of generating road gradients, we assign an altitude to each node in the network. In order to have consistent data, we applied the well known K-Means clustering algorithm to cluster the data based on their distance. We assign a high/low altitude to the cluster centers and based on the distance of the nodes in that cluster from the cluster center, we decrease/increase their altitude. We clustered 5-, 10-, 15-, and 100-customer instances into 2, 3, 4, and 10 clusters. We assumed three gradient levels based on the road gradient standards. “Level”, “nearly level”, and “very gentle slope” are the cases that we considered. In the level case, the network is flat, and the average absolute road gradient is zero. In the “nearly level” and “very gentle slope” cases the averages of absolute road

gradients are $[0.3 - 1.1)$ and $[1.1 - 3)$, respectively (Geography fieldwork, 2020).

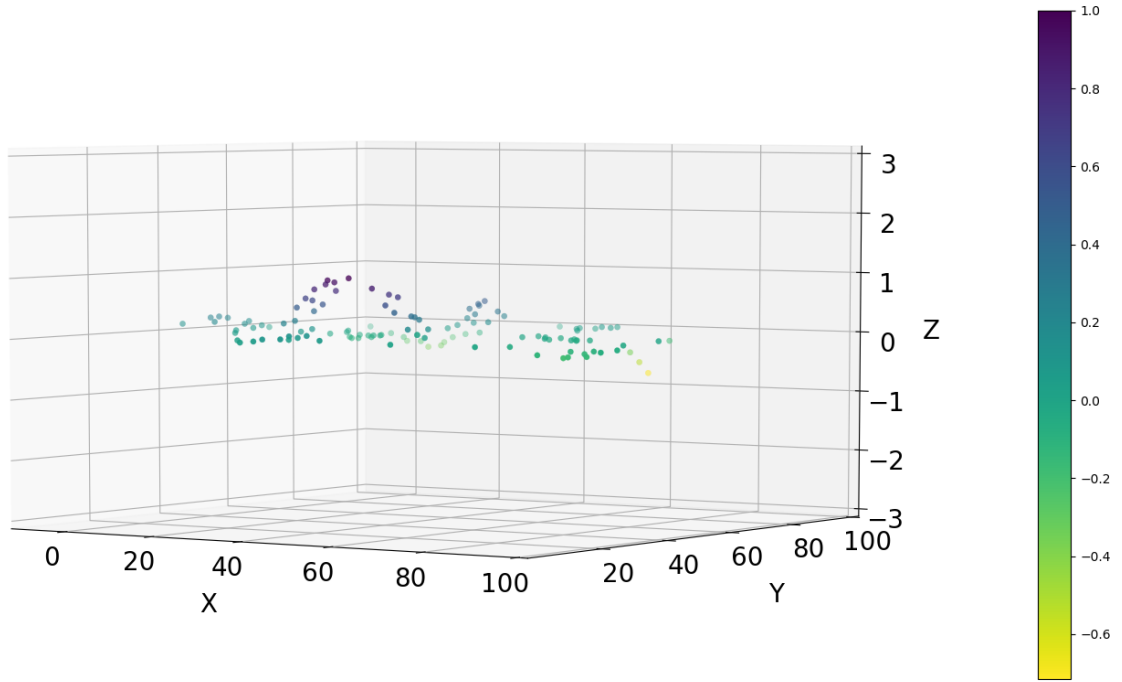


Figure 5.1. Scatter network for “c101” with 100 customers

Figure 5.1 shows the scatter network of “c101” with 100 customers, 21 stations, and one depot. The color bar denotes the altitude of the nodes which is between $[-0.71, 1.00]$. We clustered this network into 10 clusters and assigned an altitude to each node based on their distance from the cluster center which they are assigned.

The network related to “c101” is illustrated in Figure 5.2. In order to make the slope of the roads more observable, we draw the z axis between $[-3, 3]$ and assigned a color bar which is related to the nodes’ altitude.

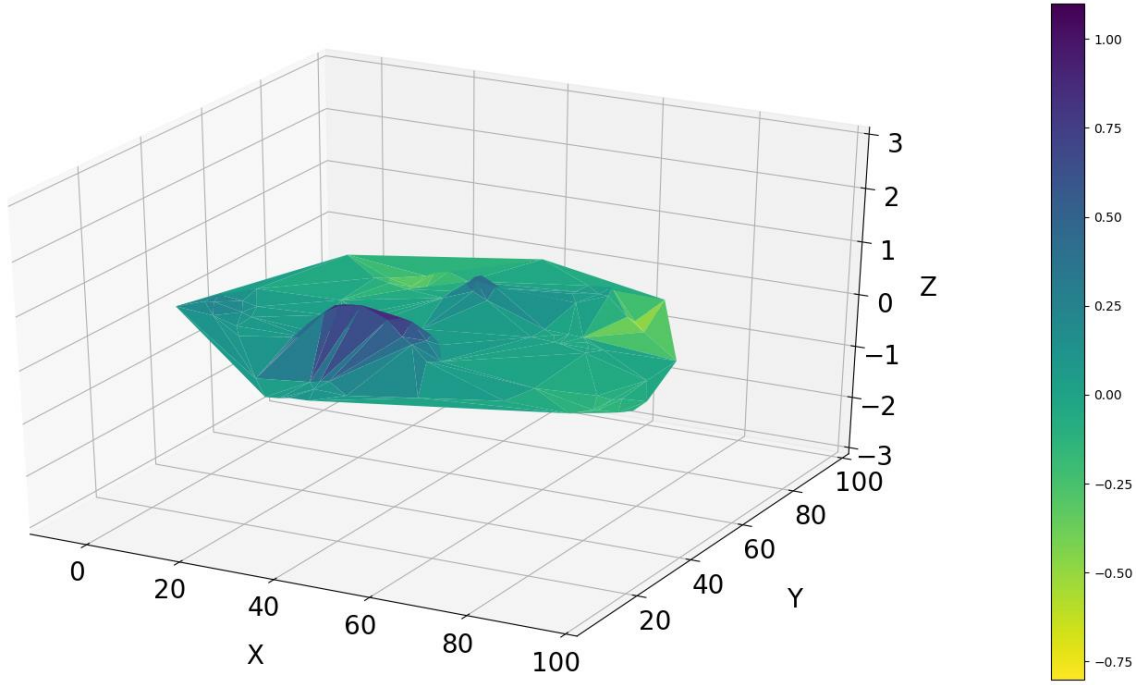


Figure 5.2. Network for "c101" with 100 customers

5.3.2. Analysis on the effect of road gradient and regenerative braking on the route planning

The results for small-size instances are provided in Table 5.1. Column “Level” reports the results for the case that the network is flat. The results in this column are corresponding to the results of model III in section 4.2.2 where the network is flat and only the cargo load affects the energy consumption whereas columns “Nearly Level” and “Very Gentle Slope” show the results for the cases that the road gradients are considered. The comparison of these columns exhibits the influence of the road gradient on routing decisions. “#Veh”, “EC”, and “t” refer to the fleet size, energy consumption, and average run time (in seconds), respectively. “% Δ EC” denotes the percentage change in the energy consumption comparing the energy consumption of “Nearly Level” and “Very Gentle Slope” cases with the “Level” case (only the cases where the number of vehicles is equal). Rows “Avg.” present the average values for the number of vehicles, energy consumption (for the instances with the same number of vehicles), and the run time for the instances with 5, 10, and 15 customers. The total averages for all the instances are shown in the row “Total Avg.”.

Table 5.1 Result for small-size instances using LNS for Level, Nearly Level, Very Gentle Slope cases

Instance	Level		Nearly Level				Very Gentle Slope			
	#Veh	EC	#Veh	EC	t (sec)	% ΔEC	#Veh	EC	t (sec)	% ΔEC
r104c5-s3	2	141.54	2	142.97	63	1.01		–		–
r105c5-s3	2	159.23	2	184.89	18	16.12		–		–
r202c5-s3	1	144.12	1	143.75	80	-0.25	1	200.45	69	39.09
r203c5-s4	1	181.32	1	216.72	55	19.52		–		–
c101c5-s3	2	266.02	2	292.15	22	9.82		–		–
c103c5-s2	1	186.83	1	196.20	28	5.02	2	197.16	33	–
c206c5-s4	1	250.63	1	340.45	40	35.84		–		–
c208c5-s3	1	168.91	1	186.15	103	10.20		–		–
rc105c5-s4	2	256.62	2	250.51	16	-2.38	3	283.84	44	–
rc108c5-s4	2	264.00	2	258.27	48	-2.17		–		–
rc204c5-s4	1	188.59	1	187.74	93	-0.45	1	196.94	110	4.43
rc208c5-s3	1	170.82	1	188.34	90	10.26		–		–
Avg.	1.42	198.22	1.42	215.68	54.51	8.54				21.76
r102c10-s4	3	336.00	3	306.91	74	-8.66		–		–
r103c10-s3	2	220.50	2	223.01	132	1.14	2	226.88	77	2.89
r201c10-s4	1	262.40	1	311.85	22	18.85		–		–
r203c10-s5	1	227.26	2	259.03	513	–		–		–
c101c10-s5	3	410.10	3	437.26	42	6.62	3	523.44	36	27.64
c104c10-s4	2	305.64	2	325.07	338	6.36	3	395.61	376	–
c202c10-s5	1	319.04	1	335.64	68	5.20	2	260.54	209	–
c205c10-s3	2	233.74	2	270.04	142	15.53	4	491.11	141	–
rc102c10-s4	5	475.00	5	447.19	40	-5.86		–		–
rc108c10-s4	3	364.62	3	374.59	122	2.73		–		–
rc201c10-s4	1	424.49	1	454.27	33	7.02	2	343.60	215	–
rc205c10-s4	2	334.54	2	380.39	191	13.70		–		–
Avg.	2.17	335.10	2.25	351.47	143.06	5.69				15.26
r102c15-s8	5	430.83	5	418.78	107	-2.80		–		–
r105c15-s6	4	350.09	4	403.60	129	15.28		–		–
r202c15-s6	1	590.31	2	417.40	462	–		–		–
r209c15-s5	1	347.31	1	456.32	63	31.39		–		–
c103c15-s5	3	401.97	3	418.69	413	4.16		–		–
c106c15-s3	3	352.29	3	313.51	237	-11.01	3	514.68	74	46.10
c202c15-s5	2	393.19	2	449.40	268	14.30	3	553.46	263	–
c208c15-s4	2	310.12	2	416.33	729	34.25		–		–
rc103c15-s5	4	415.79	4	447.13	140	7.54		–		–
rc108c15-s5	3	417.92	3	434.81	150	4.04		–		–
rc202c15-s5	2	403.03	2	416.91	516	3.44		–		–
rc204c15-s7	1	402.15	1	517.56	1097	28.70	2	397.73	1199	–
Avg.	2.58	384.06	2.67	426.64	359.25	11.75				46.10
Total Avg.	2.06	305.79	2.11	331.26	185.60	8.66				27.70

The results show that the number of vehicles increases by one in two instances in the “Nearly Level” case (shown in bold) compared to the “Level” case. Furthermore, we observe that although energy consumption values obtained in the “Nearly Level” case are often higher than the energy consumption in “Level” case, in eight instances the energy consumption decreases by considering the effect of road gradient since vehicles can save energy using the regenerative braking process. The maximum energy saving happens in

“c106c15-s3” by an 11% reduction in energy consumption compared to the “Level” case. Finally, we see in the “Very Gentle Slope” case that LNS can solve thirteen instances and the rest of them can be infeasible since the road gradient effect increases but we cannot prove their infeasibilities. In the seven instances, the number of vehicles increased by one and in one instance the number of vehicles increased by two compared to the “Level” case. The energy consumption for the instances with the same number of vehicles increased up to 46.10 % in “c106c15-s3”.

We solved the large-size instances for the mentioned cases using LNS. The results are provided in Table 5.2. We observe that in the “Nearly Level” case, the number of vehicles increases by one in the six instances, and in one instance it increases by two (shown in bold) compared to the “Level” case. Furthermore, in “c102” where the number of vehicles does not change, the energy consumption increases by 20.14%. In the Very Gentle Slope case, the LNS is unable to find a feasible initial solution as the roads became steeper and the energy consumption increased in the uphill arcs significantly. These results show the importance of considering the road gradient in route optimization.

Table 5.2 Result for large-size instances using LNS for Level, Nearly Level cases

Instance	Level		Nearly Level		
	#Veh	EC	#Veh	EC	<i>t (sec)</i>
r101	18	1741.27	19	1980.58	2000
r102	17	1630.97	18	1734.65	3535
r103	14	1404.57	15	1607.73	6616
c101	12	1186.46	13	1435.55	4544
c102	12	1173.36	12	1409.67	12102
c103	11	1395.65	13	1544.49	20327
rc101	17	1921.03	18	2056.67	4207
rc102	16	1815.88	18	2130.69	4906
rc103	14	1722.56	15	1690.53	7316

5.4. Conclusions and future research

In this chapter, we investigated the effect of road gradient, regenerative braking process, and cargo weight carried on the vehicle for the EVRPTW with a partial recharge strategy. Furthermore, we generated data based on the benchmark datasets in the literature by

assigning altitude to each node in the network. Clustering techniques are used to elevate the nodes in order to have a consistent dataset. We presented an LNS algorithm to solve the small and large-size instances. Results show that considering road gradient along with cargo load can significantly change the routing decisions and it can make the problem infeasible in the networks with steep road slope.



6. Concluding Remarks and Future Research Directions

In this thesis, we investigate the effect of ambient temperature, cargo weight, road gradient, and regenerative braking process on the fleet composition, energy consumption, and routing decisions in last-mile delivery operations. We propose new preprocessing techniques to reduce the problem size and enhance the computational performance of the solution methods.

In chapter 2 we studied EVRPTW with partial recharge by considering the effect of ambient temperature on the fleet size and energy consumption. We introduced the mathematical model of the problem and used it to solve small-size problems on CPLEX. We solved large-size problems with an ALNS algorithm. We also performed a case study using real-world data from a logistics service provider in Adana, Turkey. Our results showed that temperature can have a significant effect on the delivery operations since as temperature increases/decreases. It can increase the total energy consumption of the operations as well as the fleet size, and the routing might become impossible to serve all customers using an EV fleet. Routing plans made without considering the temperature effect might not be cost-optimal or might even become infeasible at hotter/chilly days. We performed our case study simulations considering one temperature level for each case; however, the temperature actually changes during the day. Further research on this topic may address the time-dependent variant of the problem that takes into account the hourly change of temperature may improve the accuracy of the results.

In chapter 3, we revisited the EVRPTW and proposed new preprocessing techniques to reduce the problem size and accelerate the computational performance of the solution methods. Furthermore, we developed an algorithm to check the feasibility of an instance. Our experimental tests showed that the proposed preprocessing can reduce the graph by up to 37% on average and speed up the computation time by up to 95% and 32% on small-

and large-size instances, respectively. Furthermore, it allowed ALNS to find better solutions in many large-size instances, by reducing the fleet size or saving energy. Moreover, the proposed feasibility check algorithm was able to detect all the instances proven infeasible. Further research on this topic may focus on other EVRP variants where a solution method can benefit from the proposed preprocessing techniques and feasibility check algorithm. The proposed approaches can be easily incorporated in different settings such as nonlinear charging times, different charging technologies, partial-full or single-multiple recharging strategies en-route, load-dependent EVRP. In addition, the influence of the proposed preprocessing can also be investigated on the performances of exact methods such as branch-and-cut and branch-and-price algorithms where the reduction rules can eliminate some branching options at each node and reduce the size of the branching tree.

In chapter 4 we addressed EVRPTW with partial recharge by taking into account the energy consumption associated with the cargo carried on the vehicle. We formulated three alternative mathematical models and tested their performances by solving the small-size instances using Gurobi solver. For the large-size instances, we proposed an LNS method. Our computational tests showed how the fleet size and/or energy consumption increase in comparison to the case where the load factor is neglected and revealed the importance of considering the weight of the vehicles for more accurate route planning.

In chapter 5 we investigated the effect of road gradient, regenerative braking process, and cargo weight carried on the vehicle for the EVRPTW with partial recharge strategy. Furthermore, we generated new data based on the benchmark datasets in the literature by assigning altitude to each node in the network. Clustering techniques are used to elevate the nodes in order to have a consistent dataset. We presented an LNS algorithm to solve the small- and large-size instances. Our results showed that considering road gradient along with cargo load can significantly change the routing decisions and it can make the problem infeasible in the networks with steep road slope.

We assumed charging stations were always available whenever the vehicles required recharging; however, this may not be the case in real life and there might be queues in the stations. In addition, as mentioned before, it is possible for logistics service providers to own a heterogeneous fleet to address the implications of different factors on energy

consumption. The optimal combination of different powertrain options will depend on the needs of the operation, as well as on the regional climate conditions and road network. Optimal fleet and route design for different regions arises as a challenging problem to be studied. Furthermore, the emission benefits of having an EV fleet compared to mixed fleet should be assessed. EVs are one of the most promising technologies to provide emission benefits in logistics operations; however, the effect of operational needs, climate and other regional conditions on these benefits need further investigation. Time-dependent EVRPTW where the traffic imposes different speed levels in different time periods is another future research direction in this field.



Appendix A. Results for benchmark instances

In this section, we provide the detailed results of our computational study on 5-, 10-, and 15-customer instances in Table A.1 as well as 100-customer instances in Table A.2. In Table A.1, the instance name in the first column expresses the problem type as well as the number of customers and stations it includes. For example, ‘r104c5-s3’ indicates that the customers are randomly distributed (r), their time windows are narrow (104), and the data involves 5 customers (c5) and 3 stations (s3). In Table A.2, the instance name indicates only the geographical distribution of the customers since all instances have the same number of customers and stations. Under each temperature case, columns ‘#Veh’, ‘EC’, ‘t (sec)’ refer to the number of vehicles in the fleet, total energy consumption of the route plan, and the run time in seconds. Note that in Table A.1, if the run time is 7200 seconds the optimality of the solution is not guaranteed.

Table A.1. Results for 5-, 10-, and 15-customer benchmark instances in different ambient temperatures

Inst.	Mild			Intermediate			Intense			Extreme		
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	#Veh	EC	t (sec)	#Veh	EC	t (sec)
r104c5-s3	2	137	<1	2	163	<1	2	203	<1	2	265	<1
r105c5-s3	2	156	<1	2	212	<1		INF	<1		INF	<1
r202c5-s3	1	129	<1	1	155	<1	1	205	<1	1	245	<1
r203c5-s4	1	179	<1	1	195	<1	1	264	<1	2	351	<1
c101c5-s3	2	258	<1	2	281	<1	3	344	<1		INF	<1
c103c5-s2	1	175	<1	1	191	<1		INF	<1		INF	<1
c206c5-s4	1	243	<1	1	272	<1		INF	<1		INF	<1
c208c5-s3	1	164	<1	1	179	<1	1	209	<1	1	305	<1
rc105c5-s4	2	233	<1	2	268	<1	2	314	<1	3	403	<1
rc108c5-s4	2	254	<1	2	277	<1	2	329	<1	4	591	<1
rc204c5-s4	1	185	<1	1	206	<1	1	252	<1	2	319	<1
rc208c5-s3	1	168	<1	1	189	<1	1	227	<1	1	308	<1
r102c10-s4	3	249	<1	4	287	<1	4	346	<1	5	495	<1
r103c10-s3	2	206	8	2	228	5		INF	<1		INF	<1
r201c10-s4	1	242	<1	1	288	<1	2	323	<1	2	435	<1
r203c10-s5	1	223	<1	1	280	1		INF	<1		INF	<1
c101c10-s5	3	388	<1	3	423	<1		INF	<1		INF	<1
c104c10-s4	2	274	<1	2	314	1	2	440	16		INF	<1
c202c10-s5	1	304	<1	1	331	<1	2	320	<1	2	383	<1
c205c10-s3	2	228	<1		INF	<1		INF	<1		INF	<1
rc102c10-s4	4	424	<1		INF	<1		INF	<1		INF	<1
rc108c10-s4	3	348	<1	3	384	1		INF	<1		INF	<1
rc201c10-s4	1	413	<1	2	388	<1	2	462	<1		INF	<1
rc205c10-s4	2	326	<1	2	390	<1		INF	<1		INF	<1
r102c15-s8	5	413	3	5	470	4		INF	<1		INF	<1
r105c15-s6	4	336	2	4	397	1	5	515	62		INF	<1
r202c15-s6	1	507	594	2	403	21		INF	<1		INF	<1
r209c15-s5	1	313	11	1	364	27		INF	<1		INF	<1
c103c15-s5	3	348	33	3	408	297	3	509	839	4	630	92
c106c15-s3	3	275	2	3	382	49	3	525	19		INF	<1
c202c15-s5	2	384	11	2	418	21	2	571	163	3	710	20
c208c15-s4	2	301	1	2	328	1	2	501	20		INF	<1
rc103c15-s5	4	398	117	4	433	156	4	508	31	5	750	1745
rc108c15-s5	3	370	2002	3	429	7200		INF	<1		INF	<1
rc202c15-s5	2	394	1	2	475	2		INF	<1		INF	<1
rc204c15-s7	1	382	7200	1	419	7200	1	504	7200	1	621	7200

Table A.2. Results for 100-customer benchmark instances in different ambient temperatures

Inst.	Mild			Intermediate			Intense			Extreme		
	#Veh	EC	t(sec)	#Veh	EC	t(sec)	#Veh	EC	t(sec)	#Veh	EC	t(sec)
c101	12	1043.38	140	12	1142.19	130	12	1363.48	113	12	1652.78	112
c102	11	1019.68	342	11	1128.89	270	11	1388.57	290	12	1604.49	335
c103	10	973.92	928	10	1085.16	741	11	1302.35	530	11	1628.51	415
c104	10	886.84	2017	10	988.31	1328	11	1253.78	1122	11	1560.71	987
c105	11	1022.75	169	11	1153.29	144	11	1432.27	175	12	1612.39	185
c106	11	1022.15	244	11	1138.75	233	11	1449.76	276	12	1605.75	193
c107	10	1070.30	229	11	1116.60	259	11	1356.79	239	12	1611.13	279
c108	10	1044.76	244	11	1098.49	302	11	1338.30	302	12	1601.25	322
c109	10	943.69	471	10	1078.24	1088	11	1288.30	420	12	1602.99	567
r101	18	1642.74	67	18	1809.06	77	19	2265.32	74	21	2811.71	76
r102	16	1437.66	148	16	1605.22	161	16	1989.45	165	18	2454.76	123
r103	13	1256.89	145	14	1395.47	207	14	1721.53	189	15	2120.62	165
r104	11	1096.27	239	12	1191.54	312	12	1444.11	316	13	1873.86	282
r105	15	1402.64	79	15	1557.40	84	16	1868.20	106	17	2379.43	108
r106	13	1370.55	131	14	1420.32	169	15	1704.90	182	16	2297.56	131
r107	11	1154.86	262	12	1255.45	221	13	1537.90	226	14	1930.08	240
r108	11	1043.75	342	11	1143.11	461	12	1389.93	443	13	1862.81	401
r109	13	1216.39	118	13	1361.21	126	14	1638.86	136	15	2101.29	130
r110	12	1114.47	213	12	1225.59	211	13	1471.46	208	13	1838.66	257
r111	12	1108.86	257	12	1259.48	213	13	1477.05	219	14	1921.64	193
r112	11	1035.29	286	11	1161.81	246	12	1365.34	285	13	1814.48	271
rc101	15	1755.13	84	16	1901.34	100	17	2394.81	92	—	—	—
rc102	14	1585.91	117	15	1717.75	109	16	2130.26	103	—	—	—
rc103	13	1360.77	154	13	1529.18	131	14	1874.29	136	—	—	—
rc104	11	1213.39	189	12	1350.57	84	12	1690.18	318	—	—	—
rc105	14	1470.17	85	14	1647.14	84	15	2041.41	112	—	—	—
rc106	13	1431.33	129	13	1706.69	98	15	1910.64	158	—	—	—
rc107	12	1266.05	198	12	1396.43	184	13	1737.06	219	14	2236.70	161
rc108	11	1193.58	190	11	1326.87	208	12	1655.34	265	13	2139.63	177

Appendix B. Optimal route plans for different temperatures in Adana

Tables B.1 reports the optimal route plans showing the arrival time at each node on the route, the/departure time, and the energy consumption along the route for different temperatures.

Table B.1. Optimal route plans at different ambient temperatures

Mild Case (19°C)												EC (kWh)	
Route 1	1	15	8	2	10	19	9	16	11	12	1	70.3	
Arr. time	-	8:30	10:13	10:24	10:51	11:05	11:20	13:00	13:56	14:13	14:22		
Dep. time	5:00	8:37	10:22	10:47	10:55	11:15	11:25	13:53	14:11	14:17	-		
Route 2	1	13	18	17	14	1						51.3	
Arr. time	-	8:30	10:34	10:55	12:23	12:30							
Dep. time	5:00	10:00	10:46	11:05	12:27	-							
Intermediate Case (27°C)													
Route 1	1	7	15	10	19	9	12	2	16	11	8	1	112.2
Arr. time	-	7:18	9:57	11:45	11:59	12:15	12:27	12:36	13:06	14:01	14:24	14:35	
Dep. time	5:00	9:15	10:04	11:49	12:09	12:20	12:31	12:56	13:59	14:16	14:33	-	
Route 2	1	13	18	17	14	1							56.3
Arr. time	-	8:30	10:34	10:55	12:23	12:30							
Dep. time	5:00	10:00	10:46	11:05	12:27	-							
Intense Case (33°C)													
Route 1	1	15	7	8	1								107.9
Arr. time	-	8:30	9:19	13:33	13:44								
Dep. time	5:00	8:37	11:24	13:42	-								
Route 2	1	13	18	17	4	14	1						66.2
Arr. time	-	8:30	10:34	10:55	11:40	12:34	12:42						
Dep. time	5:00	10:00	10:46	11:05	11:51	12:38	-						
Route 3	1	10	19	9	16	11	12	1					12.6
Arr. time	-	10:00	10:14	10:30	13:00	13:56	14:13	14:22					
Dep. time	5:00	10:04	10:24	10:35	13:53	14:11	14:17	-					
Extreme Case (39°C)													
INFEASIBLE													

Appendix C. Distance data of the distribution network in Adana

Table C.1 presents distances between each pair of nodes in kilometers. The first row and first column indicate the node indexes where ‘1’, ‘2’-‘7’, and ‘8’-‘19’ represent depot, stations, and customers, respectively.

Table C.1. Distance matrix (km)

Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0.00	0.00	28.70	52.20	78.69	81.53	138.00	2.25	11.88	3.50	6.67	4.65	42.39	3.41	97.08	9.06	80.33	76.48	12.20
2	0.00	0.00	28.70	52.20	78.69	81.53	138.00	2.25	11.88	3.50	6.67	4.65	42.39	3.41	97.08	9.06	80.33	76.48	12.20
3	24.90	24.90	0.00	29.20	54.10	57.50	159.50	26.90	22.20	22.80	25.90	24.10	19.40	22.50	121.72	25.30	56.70	53.49	18.70
4	45.20	45.20	20.40	0.00	32.10	35.50	179.90	47.30	42.60	43.20	46.30	44.50	5.00	42.90	142.12	45.70	34.40	31.70	39.10
5	76.90	76.90	52.10	31.70	0.00	5.50	211.60	79.00	74.30	74.90	78.00	76.20	36.70	74.60	173.82	77.40	4.30	5.80	70.80
6	81.20	81.20	56.40	36.00	4.30	0.00	215.90	83.30	78.60	79.20	82.30	80.50	40.85	78.90	178.12	81.70	1.50	8.60	75.10
7	131.26	131.26	159.96	183.46	209.71	212.55	0.00	129.00	139.00	134.18	135.27	133.00	173.41	134.43	42.60	137.50	211.35	207.50	142.43
8	2.26	2.26	30.96	54.46	80.71	83.55	136.32	0.00	13.56	5.18	8.35	6.33	44.41	5.43	94.82	10.74	82.35	78.50	14.23
9	11.80	11.80	25.82	49.22	76.18	79.02	145.35	13.27	0.00	10.01	5.44	7.35	39.88	12.80	105.90	3.53	77.82	73.97	4.82
10	3.44	3.44	26.50	49.90	76.45	79.28	139.00	4.91	10.46	0.00	5.97	3.95	40.15	2.92	99.24	7.65	78.08	74.23	9.96
11	6.41	6.41	29.31	52.71	80.87	83.71	139.95	7.87	5.78	5.64	0.00	1.95	44.57	7.71	100.50	2.97	82.51	78.66	8.31
12	4.46	4.46	30.20	53.60	80.14	82.98	138.00	5.92	7.31	3.70	2.27	0.00	43.84	5.77	98.55	4.50	81.78	77.93	9.43
13	42.64	42.64	18.10	11.40	36.30	39.14	177.60	44.89	39.90	40.61	44.00	42.20	0.00	40.60	139.72	43.40	37.94	34.09	36.31
14	3.18	3.18	26.30	49.80	76.27	79.11	137.00	5.43	11.77	1.61	7.11	5.09	39.97	0.00	100.26	8.96	77.91	74.05	9.78
15	98.23	98.23	126.87	150.27	176.68	179.52	41.50	95.96	106.11	100.37	102.67	100.40	140.38	101.39	0.00	103.81	178.32	174.47	107.12
16	9.29	9.29	28.10	52.03	78.99	81.83	142.84	10.76	2.81	7.68	2.93	4.84	42.69	9.27	103.39	0.00	80.63	76.78	7.63
17	80.20	80.20	55.40	35.00	3.30	1.20	214.90	82.30	77.60	78.20	81.30	79.50	39.35	77.90	177.12	80.70	0.00	9.10	74.10
18	76.40	76.40	51.50	31.20	5.90	9.15	211.00	78.40	73.70	74.30	77.40	75.60	34.68	74.00	173.22	76.80	7.95	0.00	70.20
19	13.97	13.97	21.00	44.40	72.56	75.40	145.00	15.44	5.26	10.69	8.78	9.97	36.26	11.03	106.34	8.79	74.20	70.35	0.00

Appendix D. Parameters

The parameters used in the LNS algorithm are displayed in Table D.1.

Table D.1. Parameter values

Par.	Description	Value
γ	Number of customers removed	Random between [20%, 55%] of all customers
Γ	Number of customers removed when exact method is selected as insertion operator	Random between [15%, 45%] of all customers
σ	Parameter used in Random Worst-Consumption and Random Worst-Time algorithm	1.5
ω	Number of routes removed	Random between [10%, 40%] of all routes
ϕ_1	First Shaw parameter	0.5
ϕ_2	Second Shaw parameter	0.25
ϕ_3	Third Shaw parameter	0.15
ϕ_4	Fourth Shaw parameter	0.25
μ	Noise parameter	0.1
ϵ	Random number for noise function	Random between [-1, 1]
α	First-Feasible Station Insertion selection probability	0.7
δ_1	Worst-Consumption selection probability	0.30
δ_2	Random Worst-Consumption selection probability	0.23
δ_3	Shaw selection probability	0.08
δ_4	Random Worst-Time selection probability	0.08
δ_5	Random selection probability	0.08
δ_6	Random Route Removal selection probability	0.15
δ_7	Greedy Route Removal selection probability	0.08
λ_1	Exhaustive Greedy Insertion selection probability	0.066
λ_2	Random Greedy Insertion selection probability	0.133
λ_3	Regret-2 Insertion selection probability	0.266
λ_4	Exhaustive Time-Based Insertion selection probability	0.133
λ_5	Random Time-Based Insertion selection probability	0.066
λ_6	Random Greedy Insertion selection probability	0.066
λ_7	Regret-2 with Noise Function insertion selection probability	0.066
λ_8	Exhaustive Time-Based with Noise Function insertion selection probability	0.066
λ_9	Random Time-Based with Noise Function insertion selection probability	0.066
β	Number of iterations to remove and reinsert stations	50
η	The number of iterations until calling the exact method for small and large-size instances	800, 5700

Appendix E. Detailed result analyzing the performance of repair-opt operator

The results for the small-size instances obtained from LNS algorithm without and with considering repair-opt operator for the load-dependent case are reported in Table E.1. “% ΔEC ” refers to the percentage change in energy consumption between the solution obtained from LNS without and with considering the repair-opt operator.

Table E.1. Results for the small-size instances obtained from LNS algorithms without and with considering repair-opt operator for the load-dependent case

Instance	LNS			Matheuristic			
	#Veh	EC	t (sec)	#Veh	EC	t (sec)	% ΔEC
r104c5-s3	2	142	11	2	142	1	0.00
r105c5-s3	2	159	8	2	159	1	0.00
r202c5-s3	1	144	16	1	144	2	0.00
r203c5-s4	1	181	9	1	181	2	0.00
c101c5-s3	2	266	9	2	266	1	0.00
c103c5-s2	1	187	10	1	187	1	0.00
c206c5-s4	1	251	10	1	251	2	0.00
c208c5-s3	1	169	9	1	169	2	0.00
rc105c5-s4	2	257	8	2	257	1	0.00
rc108c5-s4	2	264	10	2	264	1	0.00
rc204c5-s4	1	189	16	1	189	3	0.00
rc208c5-s3	1	171	14	1	171	2	0.00
r102c10-s4	4	272	22	3	336	3	—
r103c10-s3	2	220	29	2	220	14	0.00
r201c10-s4	1	270	14	1	262	2	-2.82
r203c10-s5	1	227	4	1	227	8	0.00
c101c10-s5	3	428	20	3	410	2	-4.21
c104c10-s4	2	308	48	2	306	85	-0.77
c202c10-s5	1	321	9	1	319	1	-0.62
c205c10-s3	2	234	26	2	234	3	0.00
rc102c10-s4	5	475	17	5	475	2	0.00
rc108c10-s4	3	365	23	3	365	4	0.00
rc201c10-s4	2	327	28	1	424	3	—
rc205c10-s4	2	335	25	2	335	4	0.00
r102c15-s8	5	431	28	5	431	5	-0.04
r105c15-s6	4	350	32	4	350	4	0.00
r202c15-s6	2	365	30	2	365	272	0.00
r209c15-s5	1	360	25	1	347	36	-3.52
c103c15-s5	3	402	73	3	402	419	-0.01
c106c15-s3	3	352	52	3	352	12	0.00
c202c15-s5	2	393	44	2	393	33	0.00
c208c15-s4	2	310	47	2	310	15	0.00
rc103c15-s5	4	416	45	4	416	57	0.00
rc108c15-s5	3	418	40	3	418	20	0.00
rc202c15-s5	2	403	48	2	403	11	0.01
rc204c15-s7	1	489	58	1	402	371	-17.76

The results show that in the small-size instances, 8 instances LNS with considering the

repair-opt operator finds better solutions compared with the LNS algorithm without considering repair-opt operator. In two instances, it improves the number of vehicles and in six instances it improves the solution with respect to energy consumption (improvements are shown in bold).

The results for the large-size instances obtained from the Matheuristic and LNS algorithms for the load-dependent case are reported in Table E.2. The LNS with considering repair-opt operator overperforms LNS without considering repair-opt operator in three instances with respect to the number of vehicles and in sixteen instances with respect to the energy consumption (improvements are shown in bold).

Table E.2. Results for the large-size instances obtained from LNS algorithms without and with considering repair-opt operator for the load-dependent case

	LNS			Matheuristic			
	#Veh	Best	time	#Veh	Best	time	% ΔEC
r101	19	1833	1357	18	1741	712	–
r102	17	1746	1430	17	1631	1135	-6.57
r103	14	1375	1898	14	1405	1885	2.13
r104	13	1253	3017	13	1247	2545	-0.49
r105	16	1481	1629	16	1483	1324	0.12
r106	15	1469	1928	15	1470	1752	0.06
r107	13	1357	2148	13	1332	1774	-1.83
r108	12	1190	3596	11	1252	2571	–
r109	14	1457	1818	14	1416	1734	-2.87
r110	13	1256	2967	13	1251	2129	-0.36
r111	13	1271	2601	13	1263	2029	-0.66
r112	12	1204	3521	12	1218	2763	1.19
c101	12	1198	1454	12	1186	718	-0.93
c102	12	1210	2008	12	1173	1969	-3.05
c103	11	1433	2809	11	1396	3005	-2.61
c104	11	1456	5247	11	1373	4524	-5.68
c105	12	1167	1934	12	1162	1345	-0.50
c106	12	1171	2348	12	1167	1919	-0.39
c107	12	1183	2490	12	1180	1634	-0.32
c108	12	1215	3060	12	1206	2466	-0.70
c109	12	1256	2913	11	1318	2648	–
rc101	17	1947	1485	17	1921	1119	-1.34
rc102	16	1812	1627	16	1816	1555	0.23
rc103	14	1653	1975	14	1723	1773	4.23
rc104	13	1490	3282	13	1515	2930	1.71
rc105	15	1734	1845	15	1735	1654	0.02
rc106	15	1630	2008	15	1651	1557	1.25
rc107	13	1523	2486	13	1468	2060	-3.65
rc108	13	1527	2786	13	1533	2240	0.37

BIBLIOGRAPHY

accuweather.com < <https://www.accuweather.com/en> > (last accessed: 2019-04-15)

Afroditi, A., Boile, M., Theofanis, S., 2014. Electric vehicle routing problem with industry constraints : trends and insights for future research. *Transp. Res. Procedia* 3, 452–459. <https://doi.org/10.1016/j.trpro.2014.10.026>

Akcelik, R. and Besley, M., 2003, December. Operating cost, fuel consumption, and emission models in aaSIDRA and aaMOTION. In 25th conference of australian institutes of transport research (CAITR 2003) (pp. 1-15). University of South Australia Adelaide, Australia.

Aksen, D. , Kaya, O. , Salman, S. , Tüncel, Ö. , 2014. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *Eur. J. Oper. Res.* 239 (2), 413–426 .

Barth, M. and Boriboonsomsin, K., 2009. Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, 14(6), pp.400-410.

Barth, M. and Boriboonsomsin, K., 2008. Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record*, 2058(1), pp.163-171.

Barth, M., Younglove, T. and Scora, G., 2005. Development of a heavy-duty diesel modal emissions and fuel consumption model.

Basso, R., Kulcsár, B., Egardt, B., Lindroth, P., Sanchez-Diaz, I., 2019. Energy consumption estimation integrated into the electric vehicle routing problem. *Transp. Res. Part D Transp. Environ.* 69, 141–167. <https://doi.org/10.1016/j.trd.2019.01.006>

BD Automotive, light commercial EVs, features
<<http://www.bdoto.com/productDetail.aspx?cat ID=13>> (last accessed: 2020-08-12)

Bektaş, T. and Laporte, G., 2011. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8), pp.1232-1250.

- Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., 2015. A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electron. Notes Discret. Math.* 47, 221–228. <https://doi.org/10.1016/j.endm.2014.11.029>
- Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., 2016. A new mathematical programming model for the green vehicle routing problem. *Electron. Notes Discret. Math.* 55, 89–92. <https://doi.org/10.1016/j.endm.2016.10.023>
- Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., 2018. A path-based solution approach for the green vehicle routing problem. *Comput. Oper. Res.* 103, 109–122. <https://doi.org/10.1016/j.cor.2018.10.019>
- Çatay, B., Keskin, M., 2017. The impact of quick charging stations on the route planning of electric vehicles. *IEEE Symposium on Computers and Communications (ISCC)*, 2017, IEEE (2017), 152–157. <https://doi.org/10.1109/ISCC.2017.8024521>
- Clegg, S.J., 1996. A review of regenerative braking systems. Institute of Transport Studies, University of Leeds. Working Paper 471. http://eprints.whiterose.ac.uk/2118/1/ITS105_WP471_uploadable.pdf (last accessed: 2020-08-12).
- Conrad, R.G., Figliozzi, M.A., 2011. The recharging vehicle routing problem. In: *Proceedings Industrial Engineering Research Conference*. Reno, NV, May 2011.
- Demir, E., Bektaş, T. and Laporte, G., 2011. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5), pp.347-357.
- Demir, E., Bektaş, T. and Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2), pp.346-359.
- Demir, E., Bektaş, T. and Laporte, G., 2014. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3), pp.464-478.
- Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. *Oper. Res.* 64, 1388–1405. <https://doi.org/10.1287/opre.2016.1535>
- Erdelić, T., and Carić, T., 2019. A survey on the electric vehicle routing problem: variants

- and solution approaches. *Journal of Advanced Transportation*.
<https://doi.org/10.1155/2019/5075671>
- Daily Sabah, 2017. <<https://www.dailysabah.com/turkey/2017/07/03/turkey-reeling-from-african-heat-wave>> (last accessed: 2019-04-15)
- Duran, A., Ragatz, A., Prohaska, R., Kelly, K., Walkowicz, k., 2014, Characterization of in-use medium duty electric vehicle driving and charging behavior. 2014 IEEE International Electric Vehicle Conference (IEVC), Florence, 1-8
- DW, Move is on to ban diesel cars from cities <<https://www.dw.com/en/move-is-on-to-ban-diesel-cars-from-cities/a-42747043>> (last accessed: 2020-08-12)
- Edenhofer, O., Pichs-Madruga, R., Sokona, Y., Kadner, S., Minx, J.C., Brunner, S., Agrawala, S., Baiocchi, G., Bashmakov, I.A., Blanco, G. and Broome, J., 2014. Technical summary. *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK (2014), 33–107.
- Emeç, U., Çatay, B., Bozkaya, B., 2016. An adaptive large neighborhood search for an e-grocery delivery routing problem. *Comput. Oper. Res.* 69, 109–125
<https://doi.org/10.1016/j.cor.2015.11.008>
- Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. *Transp. Res. Part E Logist. Transp. Rev.* 48, 100–114. <https://doi.org/10.1016/j.tre.2011.08.001>
- Eşarj, eşarj electric chargers map <<https://esarj.com/harita?s=TR-MER-001>> (accessed: 2020-08-12)
- European Commission, 2013. *Urban Mobility Package: Together Towards Competitive and Resource Efficient Urban Mobility*.
- European Commission, 2011. *White Paper - Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system*.
- 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. *Transp. Res. Part E Logist. Transp. Rev.* 71, 111–128.
<https://doi.org/10.1016/j.tre.2014.09.003>

- Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2017. A matheuristics for the electric vehicle routing problem with capacitated charging stations. CIRRELT.
- Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Comput. Oper. Res.* 104, 256–294. <https://doi.org/10.1016/j.cor.2018.12.013>
- Geography fieldwork, Barcelona field study center, Standard slope descriptors <<https://geographyfieldwork.com/SlopeSteepnessIndex.htm> > (last accessed: 2020-08-12)
- Giordano, A., Fischbeck, P., Matthews, H.S., 2017. Environmental and economic comparison of diesel and battery electric delivery vans to inform city logistics fleet replacement strategies. *Transp. Res. Part D Transp. Environ.* 64, 216–229. <https://doi.org/10.1016/j.trd.2017.10.003>
- Goeke, D., Schneider, M., 2015. Routing a mixed fleet of electric and conventional vehicles. *Eur. J. Oper. Res.* 245, 81–99. <https://doi.org/10.1016/j.ejor.2015.01.049>
- 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. *Eur. J. Oper. Res.* 254 (1), 80–91. <https://doi.org/10.1016/j.ejor.2016.03.040>
- 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. *Eur. J. Oper. Res.* 252, 995–1018. <https://doi.org/10.1016/j.ejor.2016.01.038>
- Hiermann, G., Hartl, R. F., Puchinger, J., Vidal, T., 2019. Routing a mix of conventional, plug-in hybrid, and electric vehicles. *Eur. J. Oper. Res.* 272(1), 235–248. <https://doi.org/10.1016/j.ejor.2018.06.025>
- 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. *Transp. Res. Part B Methodol.* 97, 102–112. <https://doi.org/10.1016/j.trb.2016.11.009>
- Hsu, C.I., Hung, S.F. and Li, H.C., 2007. Vehicle routing problem with time-windows for

- perishable food delivery. *Journal of food engineering*, 80(2), pp.465-475.
- International Energy Agency, 2017. CO2 Emissions from Fuel Combustion: Overview. IEA Stat.
14.<<https://www.iea.org/publications/freepublications/publication/CO2EmissionsfromFuelCombustionHighlights2017.pdf> > (last accessed: 2019-04-15)
- International Energy Agency, 2016. Energy and Air Pollution. *World Energy Outlook - Spec. Rep.*
266.<<https://www.iea.org/publications/freepublications/publication/WorldEnergyOutlookSpecialReport2016EnergyandAirPollution.pdf> > (last accessed: 2019-04-15)
- Jabali, O., Van Woensel, T. and De Kok, A.G., 2012. Analysis of travel times and CO2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6), pp.1060-1074.
- Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *Eur. J. Oper. Res.* 272, 879–904.
<https://doi.org/10.1016/j.ejor.2018.07.002>
- Kara, I., Kara, B.Y., Yetis, M.K., 2007. Energy minimizing vehicle routing problem. In *Combinatorial optimization and applications*. Springer Berlin Heidelberg, 62-71.
- Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transp. Res. Part C* 65, 111–127.
<https://doi.org/10.1016/j.trc.2016.01.013>
- Keskin, M., Çatay, B., 2018. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Comput. Oper. Res.* 100, 172–188.
<https://doi.org/10.1016/j.cor.2018.06.019>
- Keskin, M., Laporte, G., Çatay, B., 2019. Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Comput. Oper. Res.* 107, 77–94.
<https://doi.org/10.1016/j.cor.2019.02.014>
- Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2016. The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm. *Eur. J. Oper. Res.* 248 (1), 33–51. <https://doi.org/10.1016/j.ejor.2015.06.082>

- Koç, Ç., Karaoglan, I., 2016. The green vehicle routing problem: A heuristic based exact solution approach. *Appl. Soft Comput. J.* 39, 154–164. <https://doi.org/10.1016/j.asoc.2015.10.064>
- Kullman, N., Goodson, J. and Mendoza, J.E., 2018, June. Dynamic Electric Vehicle Routing with Mid-route Recharging and Uncertain Availability, Seventh International Workshop on Freight Transportation and Logistics 2018.
- Leggieri, V., Haouari, M., 2017. A practical solution approach for the green vehicle routing problem. *Transp. Res. Part E Logist. Transp. Rev.* 104, 97–112. <https://doi.org/10.1016/j.tre.2017.06.003>
- 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. *Comput. Oper. Res.* 101, 183–199. <https://doi.org/10.1016/j.cor.2018.07.012>
- Maden, W., Eglese, R. and Black, D., 2010. Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61(3), pp.515-522.
- Mattos Ribeiro, G., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.* 39 (3), 728–735 <https://doi.org/10.1016/j.cor.2011.05.005>
- Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2016. A multi-space sampling heuristic for the green vehicle routing problem. *Transp. Res. Part C Emerg. Technol.* 70, 113–128. <https://doi.org/10.1016/j.trc.2015.09.009>
- Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. *Transp. Res. Part B Methodol.* 103, 87–110. <https://doi.org/10.1016/j.trb.2017.02.004>
- Neubauer, J., Wood, E., 2014. Thru-life impacts of driver aggression, climate, cabin thermal management, and battery thermal management on battery electric vehicle utility. *J. Power Sources* 259, 262–275. <https://doi.org/10.1016/j.jpowsour.2014.02.083>
- Palmer, A., 2007. The development of an integrated routing and carbon dioxide emissions model for goods vehicles. , pp.1–151.

- Paz, J.C., Granada-Echeverri, M., Escobar, J.W., 2018. The multi-depot electric vehicle location routing problem with time windows. *Int. J. Ind. Eng. Comput.* 9, 123–136. <https://doi.org/10.5267/j.ijiec.2017.4.001>
- Pelletier, S. , Jabali, O. , Laporte, G. , 2016. 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transp. Sci.* 50 (1), 3–22.
- Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transp. Res. Part B Methodol.* 103, 158–187. <https://doi.org/10.1016/j.trb.2017.01.020>
- Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Comput. Oper. Res.* 34 (8), 2403–2435. <https://doi.org/10.1016/j.cor.2005.09.012>
- Raadal, H.L., Gagnon, L., Modahl, I.S., Hanssen, O.J., 2011. Life cycle greenhouse gas (GHG) emissions from the generation of wind and hydro power. *Renew. Sustain. Energy Rev.* 15, 3417–3422. <https://doi.org/10.1016/j.rser.2011.05.001>
- Ropke, S., Pisinger, D., 2006. A unified heuristic for a large class of vehicle routing problems with backhauls. *Eur. J. Oper. Res.* 171 (3), 750–775. <https://doi.org/10.1016/j.ejor.2004.09.004>
- Sassi, O., Cherif, W.R., Oulamara, A., 2015. Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. *Int. J. Math. Comput. Phys. Electr. Comput. Eng.* 9, 167–177.
- Schiffer, M., Schneider, M., Laporte, G., 2018. Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. *Eur. J. Oper. Res.* 265, 517–532. <https://doi.org/10.1016/j.ejor.2017.07.067>
- Schiffer, M., Walther, G., 2017. The electric location routing problem with time windows and partial recharging. *Eur. J. Oper. Res.* 260 (3), 995–1013. <https://doi.org/10.1016/j.ejor.2017.01.011>
- Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle routing problem with time windows and recharging stations. *Transp. Sci.* 48, 500–520. <https://doi.org/10.1287/trsc.2013.0490>

- Suzuki, Y., 2011. A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, 16(1), pp.73-77.
- Suzuki, Y., 2016. A dual-objective metaheuristic approach to solve practical pollution routing problem. *International Journal of Production Economics*, 176, pp.143-153.
- Sweda, T.M., Dolinskaya, I.S., Klabjan, D., 2017. Optimal recharging policies for electric vehicles. *Transp. Sci.* 51, 457–479. <https://doi.org/10.1287/trsc.2015.0638>
- Taş, D. 2020. Electric vehicle routing with flexible time windows: a column generation solution approach, *Transportation Letters*, DOI: [10.1080/19427867.2020.1711581](https://doi.org/10.1080/19427867.2020.1711581)
- Wang, H., Cheu, R.L., 2013. Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. *Transp. Res. Rec. J. Transp. Res. Board* 2352, 1–10. <https://doi.org/10.3141/2352-01>
- Weather.com, Past weather in Adana, Turkey
<<https://www.timeanddate.com/weather/turkey/adana/historic?month=12&year=2017>> (last accessed: 2019-04-15)
- Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A., 2016. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Comput. Oper. Res.* 76, 73–83. <https://doi.org/10.1016/j.cor.2016.06.013>
- Worley, O., Klabjan, D., Sweda, T.M. , 2012. Simultaneous vehicle routing and charging station siting for commercial electric vehicles, in: 2012 IEEE International Electric Vehicle Conference, IEVC 2012. <https://doi.org/10.1109/IEVC.2012.6183279>
- Wu, X., Freese, D., Cabrera, A., Kitch, W.A., 2015. Electric vehicles ' energy consumption measurement and estimation. *Transp. Res. Part D* 34, 52–67. <https://doi.org/10.1016/j.trd.2014.10.007>
- Xiao, Y., Zhao, Q., Kaku, I., Xu, Y., 2012. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and operations research*, 39(7), 1419-1431. <https://doi.org/10.1016/j.cor.2011.08.013>
- Xu, G., Li, W., Xu, K. and Song, Z., 2011. An intelligent regenerative braking strategy for electric vehicles. *Energies*, 4(9), pp.1461-1477.

- Yang, J., Sun, H., 2015. Battery swap station location-routing problem with capacitated electric vehicles. *Comput. Oper. Res.* 55, 217–232. <https://doi.org/10.1016/j.cor.2014.07.003>
- Yi, Z., Smart, J., Shirk, M., 2018. Energy impact evaluation for eco-routing and charging of autonomous electric vehicle fleet: Ambient temperature consideration. *Transp. Res. Part C Emerg. Technol.* 89, 344–363. <https://doi.org/10.1016/j.trc.2018.02.018>
- Yuksel, T., Michalek, J.J., 2015. Effects of regional temperature on electric vehicle efficiency, range, and emissions in the united states. *Environ. Sci. Technol.* 49, 3974–3980. <https://doi.org/10.1021/es505621s>
- Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2015. The load-dependent vehicle routing problem and its pick-up and delivery extension. *Transp Res Part B.* 71, 158–181 . <https://doi:10.1016/j.trb.2014.11.004>
- Zhang, J., Lu, X., Xue, J. and Li, B., 2008. Regenerative braking system for series hybrid electric city bus. *World Electric Vehicle Journal*, 2(4), pp.363-369.