Electric Vehicle Routing Problems : models and solution approaches

Alejandro Montoya

l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

université angers

Thèse de Doctorat

Alejandro MONTOYA

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université d'Angers
Label européen
sous le label de l'Université de Nantes Angers Le Mans

École doctorale : Sciences et technologies de l'information, et mathématiques

Discipline : Informatique et applications, section CNU 27
Unité de recherche : Laboratoire Angevin de Recherche en Ingenierie des Systèmes (LARIS)

Soutenue le 9 décembre 2016

Electric Vehicle Routing Problems: models and solution approaches

JURY

Président :	**M. Emmanuel NÉRON**, Professeur, Polytech Tours
Rapporteurs :	**M. Dominique FEILLET**, Professeur, Ecole des Mines de Saint-Etienne
	M. Daniele VIGO, Professeur, Università di Bologna, Bologna, Italy
Examinateurs :	**M. Emmanuel NÉRON**, Professeur, Polytech Tours
	M. Fabien LEHUÉDÉ, Maître de conférences, Ecole des Mines de Nantes
Directrice de thèse :	**M^{me} Christelle GUÉRET**, Professeur, Université d'Angers
Co-directeurs de thèse :	**M. Jorge E. MENDOZA**, Maître de conférences, Polytech Tours
	M. Juan G. VILLEGAS, Professeur, Universidad de Antioquia, Medellin, Colombie

Acknowledgement

This research project has been possible with the help and support of many people. I would like to express my sincere gratitude to all people who were very helpful during my Ph.D. experience. This experience gave me the opportunity to know different countries and cultures, to meet a lot of friendly people, to do new friends, and to improve my academic and research skills.

I first thank to my wife, Diana, for her support, sacrifice, company, and love. She was always there, no matter the place, the weather or the language. I am fortunate to have her by my side. I would also like to thank my mother, Angela, my stepfather Orlando and my grandmother Blanca, for their company. They have supported me all the way in the development of this thesis and they always believed I could follow this dream.

I like to thank my thesis advisors, Christelle Guéret for her patience, support and advice in all this process; Jorge Mendoza I have learned from his many things that go far beyond optimization, vehicle routing problems, and operations research; and Juan Guillermo Villegas for his intellectual generosity and to be with me in great part of my academic formation (my master and Ph.D.).

I would like to thank the jury members Daniele Vigo, Dominique Feillet, Emmanuel Néron, and Fabien Lehuédé for accepting the invitation to be part of the jury.

I want to thank the University EAFIT for the support in this project, especially to Gabriel Arango, Director of Teaching; Alberto Rodriguez, Dean of the Faculty of Engineering; and Sergio Augusto Ramirez, Chief of Production Engineering department for their trust and support. I also want to thank Mario Velez for believing in me since I was doing my career; Jairo Maya for share his time and knowledge; and Gabriel Hincapié and Nora Cadavid for his support and helping. And I would like to thank the Universidad EAFIT scientific computing center (APOLO) for their support in the computational experiments

I thank to people of Université d'Angers, especially to Michel Landron for his support and for introducing me to French culture; Simone Rees for her support and help; and my partners at LARIS, Achraf, Fally, Khadim, Ibrahim, Amin and Khaoula for their friendship.

I would like to thank everyone who supported me during my doctorate in Angers. Pedro and Natalia my "paisas brothers", who offered me their company, and supported me in the first two years. I also would thank Alexis, Evelin, Cristhope, Juan Pablo, Clemence, Ana, Silvan and Victoria for the unforgettable moments.

I want to thank the people who supported me in Colombia. To my in-laws and unconditional friends (Joa, Marc, James, Naty, David and Naty) for being with Diana when I was in France; Pauline for her French classes in Medellin; and Pablo Maya for his network flow course at the University of Antioquia.

I like to thank VeRoLog group for allowing me participating to the summer schools of 2014 and 2015. In those schools, I could improve my research skills and made a lot of friends.

Finally, I gratefully acknowledge the financial support provided by Programa de Movilidad Doctoral hacia Francia (Colfuturo - Emb. de Francia - ASCUN - Colciencias - Min. de Educación), and Programa Enlaza Mundos (Alcaldía de Medellín).

Contents

1 Introduction **11**

2 MSH for the Green VRP **15**
2.1 Introduction . . . 17
2.2 Literature review . . . 18
2.3 Multi-space sampling heuristic . . . 20
2.3.1 General structure . . . 20
2.3.2 Sampling heuristic . . . 21
2.3.3 Split . . . 22
2.3.4 Repair procedure . . . 22
2.3.5 Set partitioning . . . 27
2.4 Computational experiments . . . 27
2.4.1 Results . . . 28
2.5 Conclusion . . . 30
2.6 Nomenclature . . . 31
2.6.1 Notation for problem definition . . . 31
2.6.2 Notation for multi-space sampling heuristic . . . 31
2.6.3 Notation for repair procedure . . . 31

3 A comparative study of charging assumptions in eVRPs **33**
3.1 Introduction . . . 35
3.2 Charging asumption in the literature . . . 35
3.3 Setting up the study: problem and formulations . . . 37
3.3.1 Problem description . . . 37
3.3.2 MILP formulation . . . 38
3.4 Computational experiments . . . 41
3.4.1 Experimental settings & Test instances . . . 41
3.4.2 Experimental environment & Parameter setting . . . 41
3.4.3 Results . . . 41
3.5 Conclusion . . . 42
3.6 Nomenclature . . . 43
3.6.1 Notation for the MILP formulation . . . 43

4 ILS+HC for eVRP-PNL **45**
4.1 Introduction . . . 47
4.2 Hybrid metaheuristic . . . 47
4.2.1 Initial solution . . . 47
4.2.2 Split . . . 49
4.2.3 Variable neighborhood descent . . . 49
4.2.4 Perturb . . . 49
4.2.5 Heuristic concentration . . . 49

4.3 The fixed-route vehicle-charging problem . . . 50
4.3.1 Mixed-integer linear programming formulation . . . 50
4.3.2 Solving the FRVCP . . . 53
4.4 Computational experiments . . . 54
4.4.1 Test instances for the eVRP-NL . . . 56
4.4.2 Parameter settings & experimental environment . . . 56
4.4.3 Solution accuracy: optimal vs. heuristic charging decisions . . . 57
4.5 Conclusion . . . 60
4.6 Nomenclature . . . 61
4.6.1 Notation for problem description . . . 61
4.6.2 Notation for hybrid metaheuristic . . . 61
4.6.3 Notation for the fixed-route vehicle-charging problem . . . 62

5 TRP-CEV **63**
5.1 Introduction . . . 65
5.2 Literature review . . . 65
5.3 Problem description . . . 66
5.3.1 Mixed-integer linear programming formulation . . . 67
5.4 Parallel matheuristic . . . 70
5.4.1 Identifying feasible requests . . . 71
5.4.2 GRASP . . . 71
5.4.3 Set covering . . . 72
5.5 The fixed-route vehicle charging problem with time windows . . . 73
5.5.1 Greedy heuristic . . . 75
5.6 Computational experiments . . . 77
5.6.1 Test instances for the TRP-CEV . . . 77
5.6.2 Parameter settings & experimental environment . . . 77
5.6.3 Performance of PMa . . . 77
5.6.4 Managerial insight . . . 80
5.7 Conclusion . . . 83
5.8 Nomenclature . . . 84
5.8.1 Notation for the problem description . . . 84
5.8.2 Notation for the MILP formulation . . . 85
5.8.3 Notation for the PMa . . . 85
5.8.4 Notation for the FRVCP-TW . . . 86

6 General conclusions & perspectives **87**

A Appendices **89**
A.1 Detailed results for Green VRP instances . . . 89
A.2 Detailed results of the eVRP-NL . . . 92
A.3 Detailed results for the TRP-CEV . . . 95
A.4 Detailed results for the E-FSMFTW . . . 97

List of Tables

2.1 Summary results and comparison of our MSH with other methods on the small instances of Erdoğan & Miller-Hooks (2012). . . . 28
2.2 Summary results and comparison of our MSH with other methods on the large instances of Erdoğan & Miller-Hooks (2012). . . . 29

3.1 Comparison of our charging assumptions with charging assumptions from the literature . . 42

4.1 Comparison of the two versions of the metaheuristic on small instances with proven optima 57
4.2 Comparison of the two versions of the metaheuristic on large instances 58
4.3 Average computing time (in seconds) of different variants of the metaheuristic 59

5.1 Comparison of the PMa on small instances with proven optima 78
5.2 Comparison of the PMa with the routing software used by ENEDIS 78
5.3 Comparison of our PMa with the ALNS by Hiermann et al. (2016) on small instances . . . 79
5.4 Comparison of our PMa with the ALNS by Hiermann et al. (2016) on large instances . . . 79

A.1 Results of MSH on small instances of Erdoğan & Miller-Hooks (2012). . . . 90
A.2 Results of MSH on large instances of Erdoğan & Miller-Hooks (2012). . . . 91
A.3 Results of ILS(H)+HC and ILS(S)+HC on the 20 small instances 92
A.4 Results of ILS(H)+HC and ILS(S)+HC on the 100 large instances 92
A.5 Results of PMa on small instances of TRP-CEV 95
A.6 Results of PMa on large instances of TRP-CEV 96
A.7 Results of PMa on small instances of Hiermann et al. (2016). . . . 98
A.8 Results of PMa on large instances of Hiermann et al. (2016). . . . 99

List of Figures

2.1 Example of a feasible Green VRP solution . . . 17
2.2 Splitting a TSP tour into a Green VRP solution . . . 23
2.3 Outline of the structure of the repair graph $B = (Z, U)$ . . . 26
2.4 Optimal repair example for the three-customer sequence $r = \{0, A, B, C, 0\}$, three AFSs. . . 27
2.5 Trade-off between solution quality and CPU time . . . 29
2.6 Percentage share of CPU time by MSH phases . . . 30

3.1 Typical charging curve, where i and u represent the current and terminal voltage respectively. (Source Hõimoja et al. (2012)) . . . 35
3.2 First segment approximation (FS) . . . 36
3.3 Linear approximations of charging functions. . . . 36
3.4 Approximation vs real data . . . 37
3.5 Example of a feasible eVRP-NL solution . . . 38
3.6 Piecewise linear approximation for the charging function. . . . 39

4.1 General structure of ILS+HC . . . 48
4.2 Piecewise linear charging function and fixed-route for the FRVCP . . . 51
4.3 Piecewise linear approximation for different types of CS charging an EV with a battery of 16 kWh. . . . 57
4.4 Percentage of the routes with/without visits to CSs by instance size. . . . 59
4.5 Analysis of the number of visits to CSs . . . 60
4.6 Histogram of the average battery level (in % of the total battery capacity) after a mid-route charge . . . 61

5.1 Average fixed (i.e., the fixed cost of each technician), variable (i.e., the sum of the total travel cost, fixed charging cost and parking cost), and total cost for each instance for each fleet composition. . . . 81
5.2 Average emission (in Kg CO_2 per Km) and maximum number of visited CSs in a solution of each instance for each fleet composition. . . . 82
5.3 Average gap between the cost of the solutions with and without the option of visiting the CSs 83
5.4 Factors explaining the increment on the cost when the visits to the CSs are forbidden . . . 83

Chapter 1

Introduction

Electric vehicles (EVs), which include hybrids, plug-in hybrids, and battery electric vehicles, are one of the most promising technologies to reduce petroleum dependency and greenhouse gas emissions. Boosted by government- sponsored programs worldwide, the EVs segment has become a small but important part of the global automotive industry. Quoting a recent market study published by Berman & Gartner (2013), "sales of plug-in EVs will grow at a compound annual growth rate of nearly 40 percent over the remainder of the decade, while the overall auto market will expand by only two percent a year".

Freight and personal transportation is one of the segments in which the use of EVs is expected to generate higher economical and environmental impact. Studies suggest that transportation represents between one and two thirds of the total costs of logistics in a company (Tseng et al. 2005) and it is responsible for up to 15% of the sale price of goods (Avella et al. 2004); a high portion of this cost is linked to vehicle acquisition and operating costs. While the acquisition costs of EVs (vehicles + charging infrastructure) remain significantly higher than those of traditional combustion vehicles (CVs), their operation costs are sensibly lower. Indeed, on average the operating cost of an EV is around 0.11€/km, which is much lower than the 0.22€/km of the average operation cost of a CV [1] (Bulk 2009). After analyzing a handful of studies comparing the whole-life cost (or vehicle's total cost of ownership) of EVs and CVs, Pelletier et al. (2016) conclude that in some contexts (e.g., urban distribution, low speeds, and frequent stops) EVs area financially viable alternative to CVs. With the advances on technology and the massive adoption of EVs their economical benefits are expected to rise in the years to come. Nonetheless, in the short-to-mid term the highest benefits derived from using EVs in freight transportation are the environmental returns. As a matter of fact, estimates by Energy Information Administration (2008) suggest that transportation is responsible for 33% of the total CO2 emissions in the United States (US), and Pedersen et al. (2005) estimates this figure to be around 28% in the European Union (EU). In addition to economical and environmental aspects, a massive implementation of EVs in both freight and personal transportation is a key strategic issue in terms of energy security for the EU.

The use of EVs in freight and personal transportation is starting to get momentum. Because of the economic benefits and the environmental regulations, several companies in different sectors have started to use EVs in their operations. The courier sector is probably the leader in EV adoption. For instance, since 2011, La Poste operates at least 250 EVs and signed orders for an additional 10,000 (Kleindorfer et al. 2012). In 2011, UPS purchased 130 hybrid electric delivery trucks (HEVs) and added them to its fleet of 250 HEVs (UPS 2008). Likewise, the Portuguese postal company CTT operates with 10 small electric vans (Post & Parcel 2014).

In the service sector, several companies have started EVs adoption programs of EVs. For instance, the largest french electricity company ENEDIS has an ambitious fleet replacement program, they estimate to have 2,000 EVs between 2015 and 2016, which represents the 10% of their current fleet (ERDF 2014). The spanish electricity company Endesa expects to introduce 3,000 EVs in its fleet by 2020, which will represent 80% of its fleet (H&E 2016). In Colombia, a health care company, Sura-EPS, uses EVs for the

[1] These costs include energy (fuel or electricity), maintenance, and taxes

home healthcare service, in which physicians and nurses visit their patients (Loaiza 2014).

Other interesting cases can be found in the freight transportation. In 2013 Coca-Cola deployed more than 30 EVs in selected cities across the U.S (Priselac 2013). Heineken have started to use the Europe's largest electric truck to distribute the beer within Amsterdam (Heineken 2013). In Mexico, the largest food company, Bimbo, signed orders to buy 100 EVs which will use the energy from a wind farm (Ramirez 2015).

Despite these encouraging implementations, the massive adoption of EVs for freight transportation is still hampered by technical constraints such as low driving ranges and long battery charging times. Indeed, on average an EV has 120 km of autonomy and its charging time ranges between 0.5 and 12 hours (depending on the charging technology). This autonomy is much lower than the 500 km of average autonomy of a CV, and the EV charging times are considerably longer than CV refueling times.

To overcome these constraints, considerable research has been, and still is, devoted via technological advances in battery capacity, electric efficiency, and battery charging speed. However, as pointed out by Felipe et al. (2014), a considerable effort is still needed to develop fleet management tools able to cope with these restricting EV features. One of the fields in which the void is more critical is that of optimization techniques for solving the electric vehicle routing problems (eVRPs).

VRPs are concerned with the design of efficient routes to deliver goods and services from (to) central depots to (from) customer locations satisfying specific business constraints. From an economic perspective, the application of operations research (OR) techniques to solve VRPs has led to savings ranging from 5% to 20% in the global transportation cost in the US and the EU (Toth & Vigo 2001). Similarly, the environmental benefits of efficient routing have been remarkable. In particular, a case study reported by Hall & Partyka (2008) points out that Paragon Routing identified in a United Kingdom brewery annual reductions of more than 3,700 tons of CO2 emissions resulting from more efficient routing and depot allocation.

In the last 50 years, a vast amount of research has been devoted to solve VRP variants and it is fair to say that excellent solution approaches exist for most of them (for a comprehensive overview the reader is referred to Toth & Vigo (2014)). Unfortunately, most of those approaches are difficult, if not impossible, to use in the context in which routes are performed by EVs. Indeed, classical VRP variants consider that routes are performed by CVs. These vehicles have a long and easy-to-restore driving range (because petrol stations are available almost everywhere and tank refueling takes a negligible time). As a result, most classical routing algorithms focus on designing only one (long) route per vehicle and do not need to care for scheduling visits to refueling stations. In contrast, due to the short driving range and long battery recharging times of EVs and the limited availability of charging infrastructure (at least for 2016 standards), routing algorithms for eVRPs need to consider the detours of the EVs to the charging stations (CSs) and the decision of the amount of charge at each visited CS. In addition, the algorithms should take into account as much as possible EV features as: the nonlinear charging function, the compatibility constraint between the EVs and CS and the battery degradation cost, among others. To address these challenges, in recent years, the vehicle routing community has started to develop techniques specially tailored for eVRPs.

To the best of our knowledge, the pioneering work in the field of eVRPs is the recharging vehicle routing problem (RVRP) by Conrad & Figliozzi (2011). The RVRP extends the distance-constrained VRP by assuming that routes are serviced by EVs. In this problem, the EVs recharge their batteries at some of the customers. In 2012, Erdoğan & Miller-Hooks (2012) introduced the green vehicle routing problem (Green VRP). The Green VRP is the first problem where the EVs detour to the CSs to recharge their batteries. To solve the problem they proposed a constructive heuristic. That paper is probably the turning point in the study of eVRPs. Since then, different authors have devoted effort not only to propose more sophisticated methods for the Green VRP (Andelmin & Bartolini 2016, Koč & Karaoglan 2016, Montoya et al. 2015, Schneider et al. 2015, 2014) but also to study extensions, mostly focusing on EV applications. For instance, Goeke et al. (2015), Schneider et al. (2014), Keskin & Čatay (2016) and Desaulniers et al. (2014) extended the problem to consider customer time windows. Felipe et al. (2014) studied a version of the problem where multiple charging technologies are available and partial battery charges are allowed. In their problem, each CSs has a subset of available technologies and the decision maker should plan the routes so a function of

the travel, charging and battery degradation costs is minimized. Hiermann et al. (2016) tackled a slightly different problem in which along with the routes, the decision maker must decide the size and composition of the EV fleet. To compose the fleet, the decision maker chooses between vehicles of different capacity, battery size, and acquisition cost. Goeke & Schneider (2015) work on a VRP with a fleet composed by EVs and CVs. Finally, Sassi et al. (2015) studied the first rich eVRP. In the context of a joint project with La Poste, they tackled a realistic version of the problem where: CSs have different technologies, the fleet is made up of CVs and EVs of different types, there are compatibility constraints between the type of EV and the charging technology of the stations, charging costs depend on the time of the day, and partial charges are allowed.

Despite recent studies in the eVRP field, there is still the need to explore new solution approaches and to study new eVRPs that are closer to reality. Therefore, the purpose of this thesis is to contribute to eVRP field by proposing new methods for current eVRPs, and by studying new eVRPs. Following that objective, each chapter of this manuscript tackles a different eVRP. In consequence, each chapter is self contained in terms of the notation and literature review. A brief description of the dissertation structure follows.

In Chapter 2, we present a simple yet effective two-phase heuristic to tackle the Green VRP. In the first phase our heuristic builds a pool of routes via a set of randomized route-first cluster-second heuristics. In the second phase our approach assembles a Green VRP solution by solving a set partitioning formulation over the columns (routes) stored in the pool. At the core of our first phase lies a novel repair mechanism that optimally inserts visits to stations to restore the feasibility of routes violating the vehicle's autonomy constraint. This procedure solves a constrained shortest path problem on an auxiliary repair graph, that by construction includes only the trips between stations that respect the vehicle's autonomy constraint. To test our approach, we performed experiments on a set of instances from the literature.

The Green VRP and the most of the existing eVRPs in the literature assume that the EVs fully charge their batteries every time they reach a CS, and that the battery charge level is a linear function of the charging time. In practical situations, however, the amount of charge is a decision variable, and the battery charge level is a concave function of the charging time. In Chapter 3, we extend current eVRP models to consider partial charging and non-linear charging functions. We presented a computational study comparing our assumptions with those commonly made in the literature.

In Chapter 4, we propose an iterated local search (ILS) enhanced with heuristic concentration (HC) to tackle the eVRP with non-linear charging function (eVRP-NL). The ILS component uses a variable neighborhood descent (VND) procedure for the local search phase. Considering the importance of the charging decisions (where and how much to charge) in eVRPs, we present a particular local search operator for the charging decisions in the VND. This operator consists in solving a new fixed-route problem: the fixed-route vehicle-charging problem (FRVCP). To solve the FRVCP, we propose a heuristic and a mixed-integer linear programming formulation. The HC component assembles the final solution from the set of all routes found in the local optima reached by the ILS. To test our approach, we propose a new set of instances.

In order to tackle a real problem related with an operation with EVs, Chapter 5 introduces the technician routing problem with conventional and electric vehicles (TRP-CEV), which is inspired in the real operation of ENEDIS. This problem consists in routing a set of technicians to serve a set of geographically scattered requests within their time windows. The technicians use a fixed fleet composed of CVs and EVs. This problem considers the working schedule, lunch break and skills of the technicians. The TRP-CEV involves not only customer sequencing (i.e., routing) and charging decisions but also vehicle-to-technician assignments. The objective function in the TRP-CEV seeks to minimize the total cost, defined as the sum of the travel costs, battery charging fixed costs, the parking cost at CSs, and the fixed cost of using each technician. To tackle the TRP-CEV, we proposed a two-phase parallel matheuristic (PMa). In the first phase, the PMa builds a pool of feasible TRP-CEV routes solving, in parallel, a set of sub-problems. In the second phase the PMa assembles a TRP-CEV solution by solving an extended set covering formulation. Furthermore, we build a set of real instances based on data provided by ENEDIS. We analysed the solutions delivered by our PMa on those instances, and evaluated the quality and structure of the solution as a function of the

percentage of EVs in the fleet.

Finally, Chapter 6 presents a general conclusion and future research directions. Appendices describe the detailed results of the algorithms proposed throughout the thesis.

Chapter 2

A multi-space sampling heuristic for the green vehicle routing problem

The work presented in this chapter has been published in the Transportation Research Part C: Emerging Technologies Journal. The full reference follows:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2015). A multi-space sampling heuristic for the green vehicle routing problem. Transportation Research Part C: Emerging Technologies, 2016, 70, 113 - 128.

The work of this chapter was presented at the following conferences and seminars:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2014). A route-first cluster-second heuristic for the Green Vehicle Routing Problem. In ROADEF 2014, Bordeaux, France.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2014). A simple hybrid heuristic for the Green Vehicle Routing Problem. In VeRoLog 2014, Oslo, Norway.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A Multi-Space Sampling Heuristic for the Green Vehicle Routing Problem . GERAD seminar, Université de Montréal Campus, Montreal, Canada, March 19, 2014
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Contributions to the electric vehicle routing problems. PGMO Seminars, L'École polytechnique, Paris, France, March 18, 2016

2.1 Introduction

The use of alternative fuel vehicles (AFVs) (e.g., electric vehicles, hydrogen vehicle, among others) in freight transportation leads to new optimization problems. One of these problems is the green vehicle routing problem (Green VRP)[1], introduced by Erdoğan & Miller-Hooks (2012). The Green VRP is an extension of the well-known vehicle routing problem, arising when a fleet of AFVs based at a central depot services a set of geographically spread customers. The special feature of this VRP comes from the limited range of AFVs. To ensure the feasible completion of trips, the AFVs may visit alternative fuel stations (AFSs) en-route to refill the tank or recharge the battery.

Formally, the Green VRP is defined on an undirected and complete graph $G = (V, E)$. The vertex set $V = \{0\} \cup I \cup F = \{0, 1, 2, ..., n+a\}$ is made up of a depot (vertex 0), a set of customers $I = \{1, 2, ..., n\}$, and a set $F = \{n+1, n+2, ..., n+a\}$ of a AFSs. It is assumed that the depot can also be used as a refueling station, and that all refueling stations can handle an unlimited number of vehicles. Each vertex $i \in V$ has a service time τ_i. If $i \in I$ then τ_i is the service time at the customer; and if $i \in F \cup \{0\}$ then τ_i is the refueling time, which is assumed to be constant. The set $E = \{(i, j) : i, j \in V, i \neq j\}$ corresponds to edges connecting vertices of V. Each edge (i, j) has two associated nonnegative attributes: a travel time t_{ij} and a distance d_{ij}. The travel speeds are assumed to be constant over the edges. In addition, there is no limit on the number of stops that can be made for refueling. When refueling occurs, it is assumed that the tank is filled to its maximum capacity. The customers are served using a fleet of homogeneous AFVs with tank capacity Q and consumption rate cr. The vehicle driving-range constraint is dictated by the fuel tank capacity and a tour duration constraint T_{max}.

In the Green VRP the objective is to find a set of routes of minimum total distance such that each customer is visited exactly once; the level of the tank when the vehicle arrives at any vertex is nonnegative; each route satisfies the maximum-duration limit; and each route starts and ends at the depot. Figure 2.1 depicts a feasible solution to a Green VRP.

The Green VRP is an NP-hard problem. Indeed, Lenstra & Kan (1981) showed that the classical VRP is NP-hard. Since the VRP is a special case of the Green VRP, we can conclude that the Green VRP is also NP-hard. Moreover, recent studies show that commercial solvers cannot solve to optimality instances of 20 customers in reasonable computational times (Schneider et al. 2014). Therefore, to tackle industrial-scale Green VRP instances we need heuristic approaches.

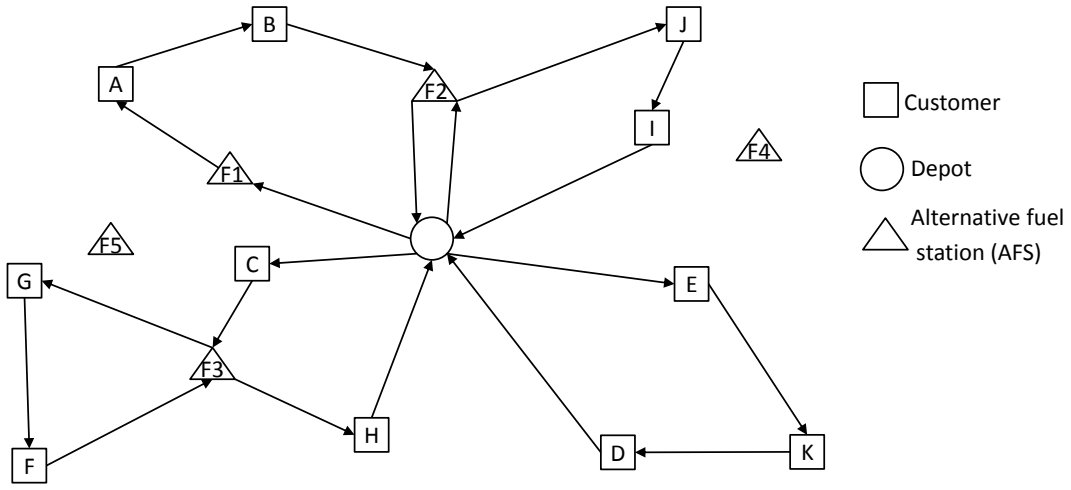


Figure 2.1 – Example of a feasible Green VRP solution

For the solution of the Green VRP we present a multi-space sampling heuristic (MSH), which is a

[1]Although in the green vehicle routing problem literature the most popular acronym is G-VRP, we prefer to use Green VRP

simple yet effective heuristic introduced by Mendoza & Villegas (2013). The idea behind MSH is to sample different solution representation spaces and then to assemble a solution with (parts of) the sampled elements. Implementations of MSH have delivered competitive results to complex routing problems such as the VRP with stochastic demands (Mendoza & Villegas 2013), the VRP with stochastic travel and service times (Gómez et al. 2015), and the combined maintenance and routing problem (Fontecha et al. 2015). The algorithm is built out of two main components: a set of *sampling functions* and an *assembling procedure*. The sampling functions are randomized route-first, cluster-second heuristics. Using these heuristics MSH draws a sample from the TSP solution representation space and extracts from it a sample of the route representation space. Later, MSH uses the sampled routes to assemble a final solution. The assembling procedure is a set partitioning model that runs over the set of routes sampled in the first phase. To implement our MSH, we adapted the randomized route-first, cluster-second heuristics proposed in Mendoza & Villegas (2013) to the Green VRP. This adaptation is far from being trivial, since route feasibility in the Green VRP is difficult to assess. Indeed, a route that is *fuel-infeasible* (that is, infeasible for fuel autonomy) can be repaired in a number of ways by inserting one or more visits to AFSs. Therefore, extracting Green VRP routes from a giant TSP tour is much more complex than in problems previously tackled using MSH. Existing metaheuristics for the Green VRP rely on insertion-based heuristics and neighborhood schemes for repairing fuel-infeasible routes (Erdoğan & Miller-Hooks 2012, Schneider et al. 2014, 2015, Felipe et al. 2014). In general, these strategies consider only one insertion at a time and therefore are exposed to myopic choices. We propose an optimal procedure based on a reformulation of the repair problem as a constrained shortest path problem.

The main contributions of this chapter are threefold: i) we introduce an optimal repair procedure based on a constrained shortest path formulation that inserts refueling stations into Green VRP routes. One of the main advantages of our procedure is that it can be used as a building block in any other Green VRP heuristic; ii) we show how to use our procedure to build a simple and effective MSH heuristic for the Green VRP; and iii) we update best-known solutions (BKSs) to 8 out of 52 standard benchmark instances for the problem.

The remainder of this chapter is organized as follows. Section 2.2 reviews the related literature. Section 2.3 presents a detailed description of our version of the multi-space sampling heuristic. Section 2.4 presents a computational evaluation of the proposed method. Finally, Section 2.5 concludes the chapter. For easy reference, Section 2.6 summarizes the notation of this chapter.

2.2 Literature review

The Green VRP was introduced by Erdoğan & Miller-Hooks (2012). They propose a mixed-integer-linear programming (MILP) formulation and two heuristics. The first heuristic is a modified Clarke and Wright savings algorithm (MCWS) that repairs infeasible routes by inserting AFSs using a savings criterion, and removes redundant AFSs after merging the routes. The second heuristic is a density-based clustering algorithm (DBCA) that first builds clusters and then runs MCWS for each cluster. In their article, the authors propose two sets of test instances: the set of "small" instances has 40 test problems with 20 customers, and the set of "large" instances has 12 instances with 111 to 500 customers. Computational experiments showed that CPLEX 11.2 was unable to solve to optimality even small instances using the MILP model. They also indicated that there were no significant differences in performance between MCWS and DBCA.

The Green VRP is closely related to the classical distance-constrained VRP (Laporte et al. 1985), but the latter does not consider the possibility of extending the vehicle's distance limitation. Another problem that is closely related to the Green VRP is the multi-depot vehicle routing problem with inter-depot routes (MDVRPI) described by Crevier et al. (2007); this problem considers intermediate depots at which vehicles can be replenished with goods. To solve this problem Crevier et al. (2007) presented a heuristic procedure that combines ideas from adaptive memory programming, tabu search, and integer programming. Tarantilis

to avoid confusions with the already established and more-studied generalized vehicle routing problem (GVRP).

et al. (2008) proposed a hybrid guided local search heuristic that outperforms the procedure proposed by Crevier et al. (2007). Muter et al. (2014) proposed a branch and price algorithm for the MDVRPI; it solves to optimality some instances with up to 50 customers.

Schneider et al. (2014) introduced the electric vehicle routing problem with time windows and recharging stations (E-VRPTW), which is an extension of the VRP with a fleet of EVs. The E-VRPTW considers limited vehicle freight capacities, customer time windows, and the possibility of recharging at any of the available stations using an appropriate recharging scheme. Schneider et al. (2014) presented an MILP formulation and a hybrid metaheuristic combining variable neighborhood search and tabu search (VNS/TS). Their VNS/TS explores infeasible solutions with respect to capacity, time windows, and battery-usage constraints. A dynamic penalizing scheme is used to guide the search toward feasible solutions. The VNS component explores 15 neighborhoods based on cyclic exchanges (Thompson & Psaraftis 1993) that transfer between routes sequences of customers of arbitrary length. Furthermore, Schneider et al. (2014) improved the MILP formulation proposed by Erdoğan & Miller-Hooks (2012), and they evaluated their VNS/TS and MILP approaches on the 52-instance testbed proposed by Erdoğan & Miller-Hooks (2012). Computational experiments showed that CPLEX 12.2 was unable to solve to optimality instances with 20 customers using their MILP model, and VNS/TS outperformed the constructive heuristics proposed by Erdoğan & Miller-Hooks (2012).

Schneider et al. (2015) introduced the vehicle routing problem with intermediate stops (VRPIS) that generalizes the Green VRP and the MDVRPI. To solve the VRPIS Schneider et al. (2015) propose an adaptive variable neighborhood search (AVNS). Their AVNS uses a modified savings algorithm to generate an initial solution that is later improved with local search. The algorithm uses an adaptive shaking with twenty-four neighborhood structures, five route selection methods, three vertex-sequence selection methods, and an adaptive mechanism to choose the route and vertex-selection methods. The solution generated at the shaking step is subsequently improved by several greedy local searches. Furthermore, the AVNS has a dynamic penalization scheme to guide the search toward feasible solutions and a simulated annealing acceptance criterion. Since the Green VRP is a special case of the VRPIS, Schneider et al. (2015) tested their approach on the instances of Erdoğan & Miller-Hooks (2012). This method outperformed all previous methods both in terms of solution quality and computational time.

Recently, Felipe et al. (2014) introduced the green vehicle routing problem with multiple technology and partial recharges (GVRP-MTPR). As the name suggests, in this problem charging stations may have different technologies (e.g., charging times) and EVs do not necessarily charge their batteries to the full capacity when they reach a charging point. In their article, the authors presented an MILP formulation, a local search method (48A), and a simulated annealing (SA). Their local search method uses 48 possible combinations of 6 different neighborhoods and selects the best overall solution. Their SA uses a relocate neighborhood to explore the solution space. Every time that the incumbent solution is updated, the SA uses a deterministic local search to try to further improve the solution. Since the GVRP-MTPR is an extension of the Green VRP, Felipe et al. (2014) tested their approaches on the instances of Erdoğan & Miller-Hooks (2012). Their methods outperformed the constructive heuristics proposed by Erdoğan & Miller-Hooks (2012), but are not competitive with the solution approaches of Schneider et al. (2014) and Schneider et al. (2015). A plausible explanation for these results, is that their 48A algorithm and their SA are not specifically tailored to the Green VRP.

Sassi et al. (2014) tackled an electric vehicle routing problem with heterogeneous mixed fleet and time dependent charging costs. In their problem the authors consider a number of realistic features such as: different charging technologies, coupling constraints between vehicles and charging technologies, charging station availability time windows, and charging costs depending on the time of the day. Regarding charging station capacity, the authors impose the maximum admissible power charge constraint over the charging spots located at the depot; however, they assume that the other charging stations have an unlimited capacity. They proposed an MILP formulation, a charging routing heuristic and a local search heuristic. They performed computational experiments on real data instances.

Gonçalves et al. (2011) considered a VRP with pick-up and delivery (VRPPD) with a mixed fleet that

consists of battery electric vehicles (BEVs) and vehicles with internal-combustion engines. The objective is to minimize the total costs (vehicle-related fixed and variable costs). They considered time and capacity constraints and assumed a time for recharging the BEVs, which they calculated from the total distance traveled and the range for one battery charge. However, they did not incorporate the location of the recharging stations into their model. Thus, they basically proposed a mixed-fleet VRPPD with an additional distance-dependent time variable. They performed computational experiments with an MILP formulation on instances with up to 17 customers.

Conrad & Figliozzi (2011) introduced the recharging vehicle routing problem (RVRP) wherein vehicles with a limited range must service a set of customers but may recharge at certain customer locations before continuing their trip. They proposed an MILP formulation of the problem. They performed computational experiments on instances of 40 customers.

Juan et al. (2014) discussed the vehicle routing problem with multiple driving ranges (VRPMDR), an extension of the classical routing problem where the total distance each vehicle can travel is limited and is not necessarily the same for every vehicle. The VRPMDR finds applications in routing electric and hybrid-electric vehicles, which can cover limited distances depending on the running time of their batteries. They proposed an MILP formulation and a multi-round heuristic algorithm that iteratively constructs a solution for the problem.

Finally, Pelletier et al. (2014) presented an overview of the field of goods distribution with EVs, that includes a review of the main transportation science literature on EVs regarding fleet size and mix, vehicle routing problem, and optimal paths.

2.3 Multi-space sampling heuristic

Mendoza & Villegas (2013) originally proposed the multi-space sampling heuristic for the vehicle routing problem with stochastic demands. Despite its simple design, MSH obtains competitive results. MSH has two phases: sampling and assembling. In the sampling phase the algorithm uses a set of randomized TSP heuristics to draw a biased sample from the set $\mathcal{K}$ of TSP-like tours (i.e., giant tours visiting all customers). Following the route-first cluster-second principle (Beasley 1983, Prins et al. 2014), MSH extracts every feasible route that can be obtained without altering the order of the customers of each sampled TSP tour. MSH uses these routes to build a set $\Omega \subset \mathcal{R}$, where $\mathcal{R}$ is the set of all feasible routes. In the assembling phase MSH follows the principle of petal heuristics (Foster & Ryan 1976) and maps set Ω to a solution s (in the set of all feasible solutions to the problem $\mathcal{S}$) by solving a set partitioning formulation. Note that in MSH the knowledge of the problem is embedded in two components: the procedure that extracts routes from the TSP tours during the sampling phase and the set-partitioning formulation used in the assembling phase. The former controls the feasibility and cost of each route while the latter controls the feasibility and cost of the whole solution. To adapt MSH to the Green VRP we designed a tailored route extraction procedure. To favor scalability we also modified the strategy used by the original MSH to build the route pool Ω. For the sake of completeness, in the remainder of this section we present all the components used in our MSH setting a special focus on those we specifically designed for the Green VRP.

2.3.1 General structure

Algorithm 1 describes the general structure of our MSH. The procedure starts by entering the sampling phase (lines 5–19). At each iteration $k \leq K$, the algorithm selects a sampling heuristic from a set $\mathcal{H}$ (line 6) and uses it to build a TSP tour tsp. Then, the algorithm uses a tour splitting procedure (known as `split`) to retrieve a solution $s \in \mathcal{S}$. Differently to the original MSH, our version does not store in Ω all the routes evaluated by `split` during the partitioning process but only the routes belonging to some of the retrieved solutions. To decide if the routes of a solution s should join Ω we use the following condition (lines 9-16): $f(s) \leq f(s^*) \cdot (1 + \lambda)$ where s^* is the best solution found, λ a positive parameter, and $f(\cdot)$ denotes the objective function of a solution. The idea behind this choice is to favor computational scalability by

reducing the size of Ω while assuring a good compromise between the diversity and the quality of the routes in the pool. In the assembly phase (line 20), the heuristic invokes a procedure called `SetPartitioning` to solve a set partitioning formulation over Ω using $f(s^*)$ as an upper bound. The resulting solution $\overline{\mathcal{R}}$ is reported by the heuristic (line 21).

Algorithm 1 Multi-space sampling heuristic: General structure

```
1:  function MULTISPACESAMPLING(G, H, K, λ)
2:      Ω ⟵ ∅
3:      k ⟵ 1
4:      while k ≤ K do
5:          for j = 1 to j = |H| do
6:              h ⟵ H_j
7:              tsp ⟵ h(G)
8:              s ⟵ split (G, tsp)
9:              if k = 1 and j = 1 then
10:                 s* ← s
11:             else if f(s) ≤ f(s*) × (1 + λ) then
12:                 Ω ⟵ Ω ∪ s
13:                 if f(s) < f(s*) then
14:                     s* ⟵ s
15:                 end if
16:             end if
17:             k ⟵ k + 1
18:         end for
19:     end while
20:     R̄ ⟵ SetPartitioning (G, Ω, s*)
21:     return R̄
22: end function
```

2.3.2 Sampling heuristic

To sample $\mathcal{K}$ (line 7 in Algorithm 1), our approach uses randomized versions of three TSP constructive heuristics. Although the strategies used to generate the randomized versions of the three heuristics are directly borrowed from Mendoza & Villegas (2013), for the sake of completeness we briefly describe them here.

Let tsp be an ordered set representing the TSP customer tour being built by a given sampling heuristic, $\mathcal{W}$ the set of customers visited by tsp, and $\mathcal{N} = I \setminus \mathcal{W}$ an ordered set of nonrouted customers. For the sake of simplicity, we assume that sets $\mathcal{W}$ and $\mathcal{N}$ are updated every time a customer is added to tsp. Let l be a random integer in $\{1, ..., min\{L, |\mathcal{N}|\}\}$, where parameter L denotes the *randomization factor* of each heuristic.

The first heuristic is randomized nearest neighbor (RNN). It initially sets $tsp = (0)$ and $u = 0$. At each iteration, RNN identifies the customer v that is the lth nearest customer to u, appends v to tsp, and sets $u = v$. RNN stops when $|\mathcal{N}| = 0$ and appends 0 to tsp to complete the tour. The second heuristic is randomized nearest insertion (RNI). It initializes tsp as a tour starting at the depot and performing a round trip to a randomly selected customer. At each iteration, RNI sorts $\mathcal{N}$ in nondecreasing order of $d_{min}(v)$, where $d_{min}(v)$ is defined to be $min\{d_{u,v}|u \in \mathcal{W}\}$. RNI then inserts $v = \mathcal{N}_l$ (i.e., the lth element in the ordered set $\mathcal{N}$) into the tour tsp in the best possible position (i.e., the position generating the smallest increment in the cost of the tour). RNI stops when $|\mathcal{N}| = 0$. The third heuristic is randomized best insertion (RBI). It initializes tsp as a tour starting at the depot and performing a round trip to a randomly selected

customer. At each iteration RBI sorts $\mathcal{N}$ in nondecreasing order of $\Delta_{min}(v)$, where $\Delta_{min}(v)$ is defined to be $min\{d(u,v)+d(v,w)-d(u,w)|(u,w) \in tsp\}$, and inserts $v = \mathcal{N}_l$ in tour tsp in the best possible position. RBI stops when $|\mathcal{N}| = 0$.

2.3.3 Split

To extract a feasible solution s from tsp (line 8 in Algorithm 1), our approach uses an adaptation of the optimal tour splitting procedure for the VRP introduced by Prins (2004). The splitting procedure builds a directed acyclic graph $G^* = (V^*, A)$ composed of the ordered vertex set $V^* = (v_0, v_1, ..., v_i, ..., v_n)$ and the arc set A. Vertex $v_0 = 0$ is an auxiliary vertex, while vertices $v_1, ..., v_n \in tsp \setminus \{0\}$. The vertices in V^* are arranged in the same order in which they appear in tsp. Arc $(v_i, v_{i+n_r}) \in A$ represents a feasible route $r_{v_i,v_{i+n_r}}$ with distance $d_{r_{v_i,v_{i+n_r}}}$ starting and ending at the depot and visiting customers in the sequence v_{i+1} to v_{i+n_r}. Route $r_{v_i,v_{i+n_r}}$ may not satisfy the fuel constraint (i.e., the route's fuel consumption is greater than Q). If it does not, we try to repair it by inserting visits to AFSs. If the insertion of AFSs increases the duration of the route beyond T_{max} then we do not include the arc associated with the route in G^*. The insertion of visits to AFSs is accomplished by the optimal repair procedure explained in Section 2.3.4. To obtain a feasible solution s for the Green VRP, the procedure finds the set of arcs (i.e., routes) along the shortest path connecting 0 and v_n in G^*. It is worth noting that since G^* is directed and acyclic, building the graph and finding the shortest path can be done simultaneously (Prins 2004).

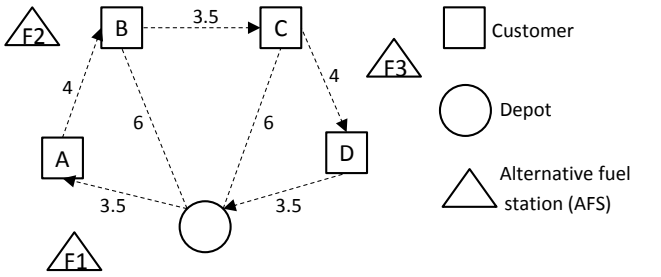
Algorithm 2 shows the tour splitting procedure based on the split algorithm for the capacitated VRP (Prins 2004). After initializing the shortest path labels (lines 2–6), split enters the outer loop (lines 7–41). Each pass through the outer loop sets the tail of an arc and initializes an empty route (lines 8–11). Then we use the inner loop (lines 13–40) to scan all the arcs sharing the same tail node. At each inner-loop iteration, we explore a new arc by simply adding the next customer to the route (line 16). In the next steps (lines 17–23) we compute the weight and time of the arc (i.e., the total distance c_r and total time tp_r of the associated route). If an arc is feasible for the duration constraint (line 24) we check the feasibility of the associated route with respect to the fuel constraint (and store the result in a boolean variable f_r). When a route is infeasible for the fuel constraint but $tp_r \leq T_{max}$, we try to repair the route using procedure `Repair(r,G)` (line 27). For an arc (i,j) to be added to the graph, its corresponding route must be feasible for fuel and time. If the shortest path label of the head node of arc (i,j) can be improved (i.e., $label_{i-1} + c_r \leq label_j$), then we update the shortest path label and the predecessor information (lines 35–38). The algorithm then moves to the next inner-loop iteration or exits the loop. After completing the outer loop we retrieve the solution using the tour tsp and the predecessor labels (for an algorithm to retrieve the solution we refer the reader to Prins (2004)).

To adapt the tour splitting procedure to the Green VRP, it is necessary to introduce two important functions: `checkFuel` (line 24), which evaluates the feasibility of a route with respect to the vehicle's fuel constraint, and `Repair` (line 26), which tries to repair the route, and returns a boolean variable (indicating whether or not the route could be repaired) and the distance of the repaired route. Note, however, that there are some cases where the repaired route is feasible for the fuel constraint but infeasible for the duration constraint. This occurs because the time needed to recover the feasibility of the route (i.e., the travel time to the AFSs and the service time there) increases its planned duration beyond T_{max}; in these cases we do not add the associated arc to the graph.

Figure 2.2 illustrates the tour splitting procedure. Figure 2.2a shows the TSP tour and the relevant information. Figure 2.2b depicts the auxiliary graph G^*; the arcs in bold correspond to the shortest path. Figure 2.2c is a table showing the evaluation of the arcs.

2.3.4 Repair procedure

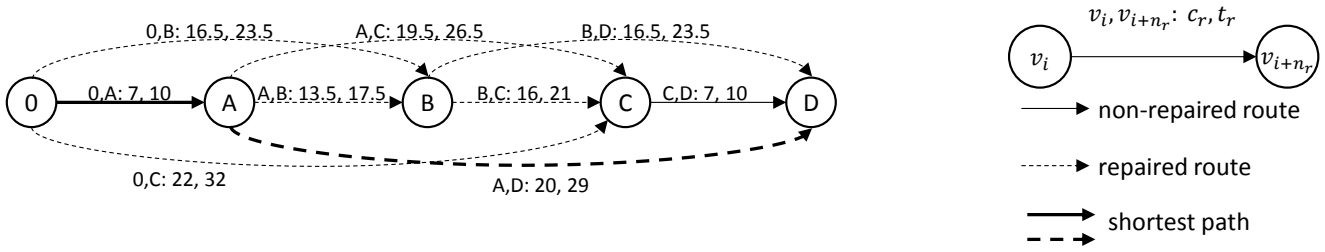
Existing approaches for the Green VRP depend on insertion-based heuristics and neighborhood schemes for repairing fuel-infeasible routes. The MCWS algorithm proposed by Erdoğan & Miller-Hooks (2012)



$Q = 7$
$T_{max} = 33$
$cr = 1$
$speed = 1$
$\tau_i = 1 \; \forall i \in F \cup \{0\}$
$\tau_i = 2 \; \forall i \in I$

Distance matrix between vertices and AFSs			
	F1	F2	F3
0	3.0	6.5	5.5
A	2.5	4.0	7.0
B	6.5	1.0	6.0
C	8.0	4.5	2.5
D	6.5	7.0	2.5

(a) TSP tour and instance data



(b) Auxiliary graph

Arc (route) evaluation						
Arc	Sequence	Feasible*		Repaired**	Distance	Time
		Tmax	Autonomy			
0,A	0, A, 0	✔	✔	N	7.0	10.0
0,B	0, F1, A, F2, B, 0	✔	✔	✔	16.5	23.5
0,C	0, F1, A, F2, B, C, F3, 0	✔	✔	✔	22.0	32.0
0,D	0, F1, A, F2, B, C, F3, D, 0	✘	✔	✘	22.5	34.5
A,B	0, B, F2, 0	✔	✔	✔	13.5	17.5
A,C	0, B, F2, C, F3, 0	✔	✔	✔	19.5	26.5
A,D	0, B, F2, C, F3, D, 0	✔	✔	✔	20.0	29.0
B,C	0, F3, C, F3, 0	✔	✔	✔	16.0	21.0
B,D	0, F3, C, F3, D, 0	✔	✔	✔	16.5	23.5
C,D	0, D, 0	✔	✔	N	7.0	10.0

* ✔: feasible, ✘: infeasible

** **N**: route does not need to be repaired ✔: route can be repaired, ✘: route can not be repaired

(c) Arc evaluation

Figure 2.2 – Splitting a TSP tour into a Green VRP solution

Algorithm 2 Tour splitting

function SPLIT(G,tsp)
  $label_0 \longleftarrow 0$ ▷ $label$: shortest path labels
  for $i = 1$ to n **do**
    $label_i \longleftarrow +\infty$
    $pred_i \longleftarrow 0$ ▷ $pred$: predecessor labels
  end for
  for $i = 1$ to n **do**
    $r \longleftarrow (0)$ ▷ Initialize route r
    $c_r \longleftarrow 0$ ▷ Initialize total distance c_r
    $tp_r \longleftarrow 0$ ▷ Initialize total time tp_r
    $j \longleftarrow i$
    continue $\longleftarrow$ true
    while $j \leq n$ **and** continue=true **do**
      $v \longleftarrow tsp_j$ ▷ Get customer in position j of tsp
      $v' \longleftarrow tsp_{j-1}$ ▷ Get customer in position $j-1$ of tsp
      $r \longleftarrow r \cup \{v\}$
      if $j = i$ **then**
        $c_r \longleftarrow d_{0,v} + d_{v,0}$ ▷ Update total distance
        $tp_r \longleftarrow t_{0,v} + t_{v,0} + \tau_v + \tau_0$ ▷ Update total time
      else
        $c_r \longleftarrow c_r - d_{v',0} + d_{v',v} + d_{v,0}$ ▷ Update total distance
        $tp_r \longleftarrow tp_r - t_{v',0} + t_{v',v} + t_{v,0} + \tau_v$ ▷ Update total time
      end if
      if $tp_r \leq T_{max}$ **then** ▷ Check feasibility for time
        $f_r \longleftarrow$ `checkFuel(`r`)` ▷ Evaluate fuel feasibility
        if $f_r =$ false **then**
          $\langle c_r, f_r \rangle \longleftarrow$ `Repair(`r`,`G`)` ▷ Repair route r
          if $f_r =$ false **then**
            continue $\longleftarrow$ false
          end if
        end if
      else
        continue $\longleftarrow$ false
      end if
      if $label_{i-1} + c_r \leq label_j$ **and** continue=true **then**
        $label_j \longleftarrow label_{i-1} + c_r$ ▷ Update label
        $pred_j \longleftarrow i - 1$ ▷ Update predecessor
      end if
      $j \longleftarrow j + 1$
    end while
  end for
  $s \longleftarrow RetrieveSolution(pred, tsp)$
  return s
end function

starts inserting the AFS with the least insertion cost (i.e., distance added to the tour) in the fuel-infeasible back-and-forth routes. After the routes merging step, MCWS inserts into each fuel-infeasible route the AFS with the least insertion cost and then evaluates the possibility of eliminating redundant AFSs. Schneider et al. (2014) used a penalization scheme for the fuel-infeasible routes and proposed an operator that performs insertions and removals of AFSs. That operator uses a tabu list in order to control insertions already tested. Schneider et al. (2015) used a penalization scheme for the fuel-infeasible routes and presented three operators to improve the location of AFSs within the route. The first operator moves an AFS to a different position within a route. The second operator evaluates for each AFS visit of each route whether visiting a different AFS decreases the routing cost. The third operator aims at removing redundant AFS visits. Finally, Felipe et al. (2014) proposed a constructive heuristic, which considers the insertion of AFSs to fuel-infeasible routes. They presented a neigboorhood scheme to improve the location of an AFS in a route, without modifying the sequence of visits to the customers. Notice that all these strategies consider only one insertion at a time and therefore are exposed to myopic choices. In contrast, our repair procedure optimally chooses which stations to insert and where to insert them while considering all the possible combinations of insertions leading to an energy-feasible route. To accomplish its goal, our procedure relies on a reformulation of the repair problem as a constraint shortest path problem.

Let $\Pi = \{\pi_1, ..., \pi_i, ..., \pi_j, ..., \pi_{n_r+1}\}$ be a route that violates the fuel constraint, and $\pi_1 = 0$. The feasibility of Π may be restored by inserting visits to AFSs. We use an optimal procedure that simultaneously decides which stations must be visited and their optimal insertion position within the route. The procedure can be seen as a constrained shortest path problem in a repair graph $B = (Z, U)$. Figure 2.3 depicts the structure of B[2]. The vertex set $Z = \{\alpha, ..., [i,k], ..., \beta\}$ is made up of two dummy vertices (α and β) that act as copies of the depot, representing the source and sink vertices of B; and the vertices $[i,k]$, which represent a visit to station $k \in F$ after visiting vertex π_i (i.e., the ith element in route Π), where $1 \leq i \leq n_r + 1$ and $1 \leq k \leq a$. To define the edge set U, let us first introduce some key elements. Let $P = \{p_1, ..., p_j, ..., p_{|P|}\}$ be a path in G. For a given path P we define three metrics: its distance $d(P) = \sum_{j=1}^{|P|-1} d_{p_j,p_{j+1}}$, its total planned time $t(P) = \sum_{j=1}^{|P|-1} t_{p_j,p_{j+1}} + \sum_{j=2}^{|P|} \tau_{p_j}$, and its fuel consumption $q(P) = cr \times d(P)$. The arc set U is composed of five types of arcs, that is, $U = \bigcup_{i=1}^{5} U_i$, where:

- U_1: the outgoing arcs of α. An arc $(\alpha, [i,k])$ represents the path $P = (0, F_k)$ if $i = 1$, and $P = (0, ..., \pi_i, F_k)$ if $i > 1$. Its cost and time are defined as $\bar{c}_{u:u\in U_1} = d(P)$ and $\bar{t}_{u:u\in U_1} = t(P) + \tau_0$.
- U_2: the arcs connecting two stations without visiting any customer. Arc $([i,k],[i,l])$ represents the path $P = (F_k, F_l)$. Its cost and time are defined as $\bar{c}_{u:u\in U_2} = d(P)$ and $\bar{t}_{u:u\in U_2} = t(P)$.
- U_3, the arcs connecting two stations and visiting some customers in between. Arc $([i,k],[j,l])$ represents the path $P = (F_k, \pi_{i+1}, ..., \pi_j, F_l)$. Its cost and time are defined as $\bar{c}_{u:u\in U_3} = d(P)$ and $\bar{t}_{u:u\in U_3} = t(P)$.
- U_4, the incoming arcs to β representing the return to the depot after visiting some customers since the last visit to a station. Arc $([i,k],\beta)$ represents the path $P = (F_k, \pi_{i+1}, ..., \pi_{n_r+1}, 0)$. Its cost and time are defined as $\bar{c}_{u:u\in U_4} = d(P)$ and $\bar{t}_{u:u\in U_4} = t(P) - \tau_0$.
- U_5, the incoming arcs to β representing a return to the depot directly from a station after visiting the last customer in route Π. The arc $([n_r+1,k],\beta)$ represents the path $P = (F_k, 0)$. Its cost and time are defined as $\bar{c}_{u:u\in U_5} = d(P)$ and $\bar{t}_{u:u\in U_5} = t(P) - \tau_0$.

We include in U only arcs with $\bar{c}_u \times cr \leq Q$ and $\bar{t}_u \leq T_{max}$. With this graph construction procedure we ensure that all paths from α to β represent a route that visits customers in the same order they appear in Π and is feasible with respect to the fuel constraint. Note, however, that not every path connecting α and β in B represents a route that is feasible in terms of the duration constraint. Therefore, to find an optimal repair for route Π, we need to solve a constrained shortest path problem (CSP), where the maximum travel time T_{max} is the constrained resource. It is interesting to observe that since all the arcs of B are feasible for

[2]This graph resembles that used by Villegas et al. (2010) in a route-first cluster-second heuristic for the single truck and trailer routing problem with satellite depots.

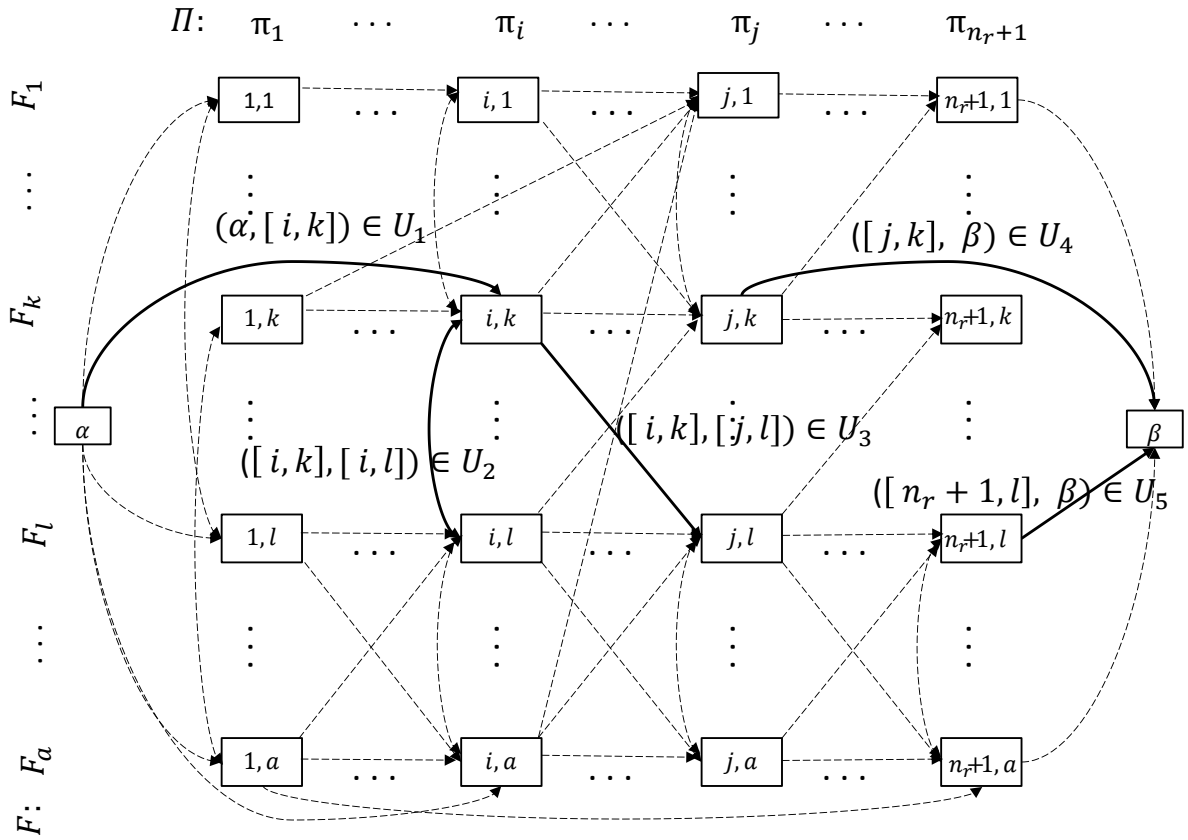
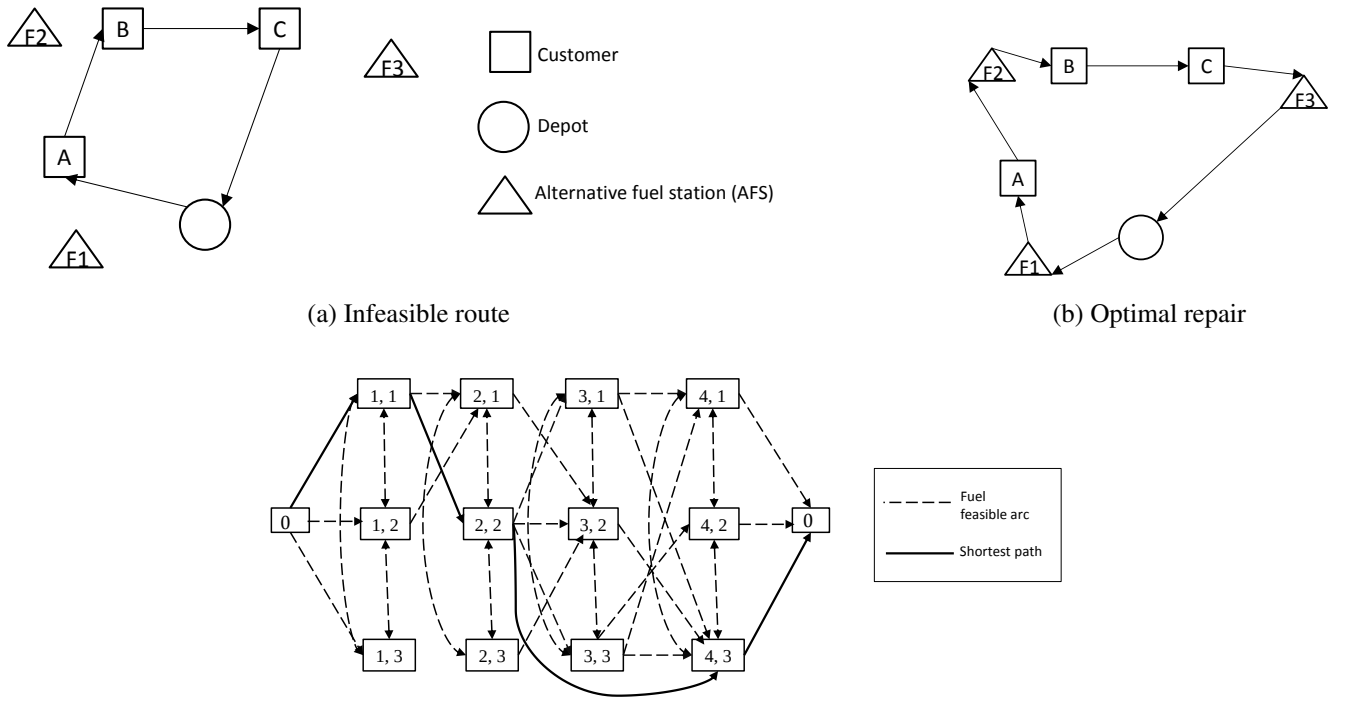


Figure 2.3 – Outline of the structure of the repair graph $B = (Z, U)$

fuel, the tank capacity is not a constraint for the CSP. To solve the CSP we use the pulse algorithm (Lozano & Medaglia 2013*a*). The algorithm is based on the idea of propagating pulses through a network from a source node to a sink node (α and β in our case). At the core of the algorithm lies the ability to effectively and aggressively prune pulses (i.e., prevent their propagation) without jeopardizing the optimal path. The pulse algorithm is one of the state-of-the-art methods for the solution of resource-constrained shortest path problems (Lozano & Medaglia 2013*b*).

Figure 3.3 shows a detailed example of the repair procedure. Figure 2.4a illustrates an infeasible route with three customers and three AFSs for a Green VRP. Figure 2.4b shows the route after the optimal insertion of the AFSs using the repair procedure. Figure 2.4c shows the corresponding repair graph and the shortest path.



(a) Infeasible route

(b) Optimal repair

(c) Graph

Figure 2.4 – Optimal repair example for the three-customer sequence $r = \{0, A, B, C, 0\}$, three AFSs.

2.3.5 Set partitioning

In the assembly phase, MSH maps the set Ω to a solution in S by solving a set partitioning formulation $\left(min_{\overline{R}\subseteq\Omega}\left\{\sum_{r\in\overline{R}} d_r : \cup_{r\in\overline{R}} = V; r_i \cap r_j = 0 \; \forall r_i, r_j \in \overline{R}\right\}\right)$. The objective is then to select the best subset of routes from Ω to build the set of routes $\overline{R}$ (i.e., the final solution) guaranteeing that each customer will be visited by exactly one route.

2.4 Computational experiments

To test our approach, we ran it on the 52-instance testbed proposed by Erdoğan & Miller-Hooks (2012). These instances consist of 5 sets. Four sets contain ten 20-customer instances ranging from 2 to 10 AFSs. The remaining set, resulting from a case study, consists of 12 instances with the number of customers ranging from 111 to 500 and between 21 and 28 AFSs. In this testbed there are instances with infeasible customers. Therefore, Erdoğan & Miller-Hooks (2012), Schneider et al. (2014), Schneider et al. (2015), and Felipe et al. (2014) filter the customers that either cannot be served directly within the maximum route duration or whose direct service requires visiting more than one refueling station. To allow comparison with previously published results, we followed the same convention and filter unfeasible customers. It is important to remark that in the problem definition an AFS vertex could have as successor another AFS, so it is possible to insert visits to more than one AFS between two customer visits. According to Erdoğan & Miller-Hooks (2012), the geographical coordinates given for the customers have to be converted to Cartesian coordinates using the Haversine formula (Bullard & Kiernan 1922) with an average Earth radius of 4,182.45 miles. It is worth noting that although Erdoğan & Miller-Hooks (2012) include an additional constraint on the number of vehicles in their Green VRP formulation, their experiments are conducted with an unlimited-fleet version of the problem. Therefore, our results are directly comparable to theirs.

We implemented our MSH in Java (jre V.1.7.0_51) and used Gurobi Optimizer (version 5.6.0) to solve the set partitioning problem. We set a time limit of $10 \cdot n$ seconds on Gurobi to control the running time of the set partitioning problem. All the experiments were run on a computing cluster with 2.33 GHz Intel

Table 2.1 – Summary results and comparison of our MSH with other methods on the small instances of Erdoğan & Miller-Hooks (2012).

Metric	MCWS /DBCA	VNS /TS	AVNS	48A	MSH(1k)	MSH(5k)	MSH(10k)
Number of BKS	2/40	38/40	40/40	29/40	35/40	40/40	40/40
Avg. Gap (%)	NR	NR	0.15%	NR	0.38%	0.04%	0.01%
Max. Gap (%)	NR	NR	1.73%	NR	3.61%	0.47%	0.13%
Avg. Best Gap (%)	8.72%	0.63%	0.00%	0.46%	0.09%	0.00%	0.00%
Cum. Number of Veh.	245	223	NR	225	222	222	222
Avg. Time (min)	NR	0.65	0.17	0.02	0.01	0.04	0.07
Max. Avg. time (min)	NR	0.88	0.38	0.04	0.02	0.06	0.12
Computer	Pentium 4 3.2GHz	Core I5 2.67GHz	Core I5 2.67GHz	Core I5 2.8GHz		XEON E5410 2.33 GHz	
Runs	NR	10	10	1	10	10	10

NR: not reported

Xeon E5410 processors with 16 GB of RAM running under Linux Rocks 6.1.1. Each replication of the experiments was run on a single processor.

2.4.1 Results

After conducting a parameter tuning campaign, we set L_{RNN}, L_{RNI}, and L_{RBI} to 2, and $\lambda = 1$. We found that these parameters lead to a pool of well-diversified routes. For the sake of brevity, we will not discuss these experiments.

Tables 2.1 and 2.2 summarize the results delivered by our MSH on the small and large Green VRP instances running with 3 configurations: $K = 1,000$; $K = 5,000$; and $K = 10,000$ (defined as MSH(K)). We compare our results to the best result obtained by MCWS and DBCA by Erdoğan & Miller-Hooks (2012), the VNS/TS by Schneider et al. (2014), the AVNS by Schneider et al. (2015), and the 48A and SA by Felipe et al. (2014). The rows of Tables 2.1 and 2.2 indicate the number of times that each method found the BKS, the average and maximum gap between the average solution and the BKS (in %), the average best gap[3] (in %), the cumulative number of vehicles[4], and the average and maximum computational times (in minutes). The results of VNS/TS, AVNS and our MSH are computed over 10 runs and the 48A and SA over a single run. Erdoğan & Miller-Hooks (2012) reported the best solution of several runs with different parameters but did not give the exact number of runs. The detailed results are reported in Appendix A.1.

For small instances, MSH(5k) and MSH(10k) have competitive results with reference to MCWS/ DBCA, VNS/TS, AVNS, and 48A (Table 2.1). They have an average gap of only 0.04% and 0.01%, match the 40 BKSs, and use the lowest cumulative number of vehicles. In terms of CPU time, MSH(5k) and MSH(10k) are only outperformed by MSH(1k) and 48A. MSH(1k) is the fastest method, and outperforms MCWS/DBCA, VNS/TS, and 48A in terms of solution quality.

For large instances, MSH(5k) outperforms MCWSA/ DCBA, VNS/TS, 48A, and SA in terms of both solution quality (i.e., average best gap, and cumulative number of vehicles) and CPU time (Table 2.2). When compared to AVNS, MSH(10k) has a better average best gap with respect to BKSs (0.05% vs. 0.17%) and finds more BKSs (8/12 vs. 4/12), whereas AVNS is faster and it seems to scale better.

It is interesting to observe that for both the small and large instances our three MSH configurations use the lowest cumulative number of vehicles.

In order to give a graphical view of key metrics presented in Tables 2.1 and 2.2, Figure 2.5 presents a head-to-head comparison between the solution methods. It shows the trade-off between solution quality

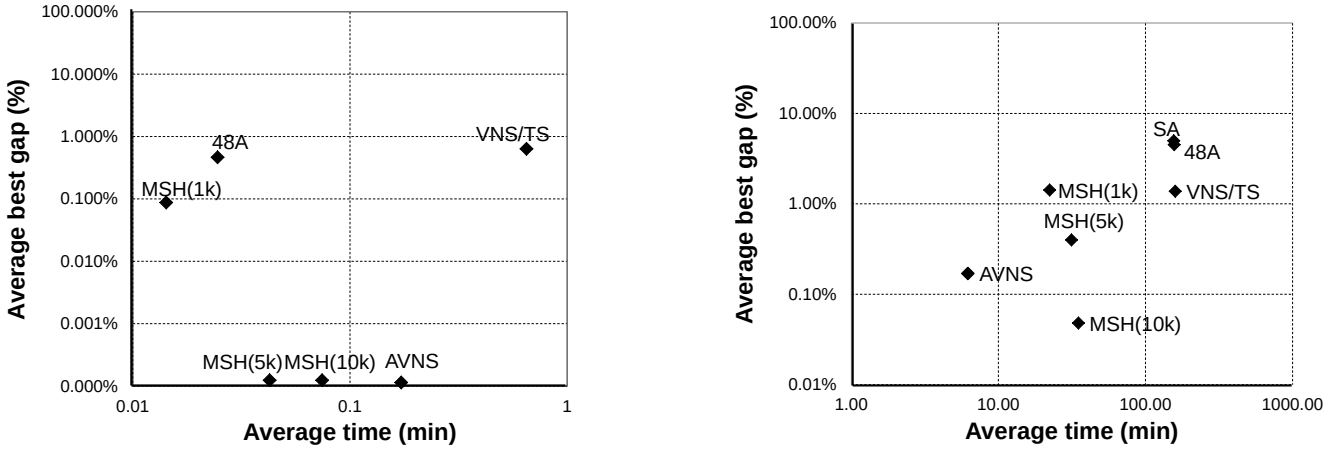
[3]The best gap is the gap between the best solution and the BKS

[4]This metric is the sum of the number of vehicles of all best solutions (Bräysy & Gendreau 2005)

Table 2.2 – Summary results and comparison of our MSH with other methods on the large instances of Erdoğan & Miller-Hooks (2012).

Metric	MCWS /DBCA	VNS /TS	AVNS	48A	SA	MSH (1k)	MSH (5k)	MSH (10k)
Number of BKS	0/12	0/12	4/12	0/12	0/12	0/12	0/12	8/12
Avg. Gap (%)	NR	NR	0.92%	NR	NR	2.64%	1.48%	1.02%
Max. Gap (%)	NR	NR	1.84%	NR	NR	5.77%	4.21%	3.62%
Avg. Best Gap (%)	15.97%	1.38%	0.17%	4.50%	4.97%	1.42%	0.40%	0.05%
Cum. Number of Veh.	508	461	NR	466	459	454	445	444
Avg. Time (min)	NR	159.58	6.20	157.03	156.05	27.92	31.47	35.04
Max. Avg. time (min)	NR	525.52	19.51	514.68	456.26	80.11	84.95	89.95
Computer	Pentium 4 3.2GHz	Core I5 2.67GHz	Core I5 2.67GHz	Core I5 2.8GHz	Core I5 2.8GHz	XEON E5410 2.33GHz		
Runs	NR	10	10	1	1	10	10	10

NR: not reported



(a) Small instances (%)

(b) Large instances

Figure 2.5 – Trade-off between solution quality and CPU time

and CPU time[5]. Each method is represented as a point in the plot. The X coordinate of the point represents the average best gap with respect to the BKS, while the Y coordinate represents the average CPU time. Notice that MCWS and DBCA were not included in the comparison since CPU times for these algorithms were not reported in by Erdoğan & Miller-Hooks (2012).

For the small instances, Figure 2.5a shows that MSH(5k) and MSH(10k) dominate the AVNS and VNS/TS, while MSH(1k) dominates 48A and VNS/TS. Moreover, MSH(5k) has a remarkable performance on small instances, because its CPU time is close to that of 48A (which is the fastest approach in the literature) and its average best gap is 0.00%. On the other hand, on the large instances, MSH(5k) and MSH(10k) dominates SA, 48A and VNS/TS (Figure 2.5b). The comparison between MSH(10k) and AVNS shows that there is not clear dominator: while the former dominates in solution quality, the latter dominates in CPU time. Nonetheless, it is worth recalling that while AVNS uses elaborate components (i.e., an adaptive shaking with twenty-four neighborhood structures, five route selection methods, among others), our MSH uses simple building blocks that, with the notable exception of the repair procedure, are common in the literature. Therefore our MSH is probably easier to implement and extend to tackle other problems.

Finally, we analyzed the CPU time of the two MSH phases (sampling and assembly). Figure 2.6 presents the percentage of time spent on each phase for each instance size (in terms of number of customers). The

[5]This comparison tool was introduced by Vidal et al. (2015)

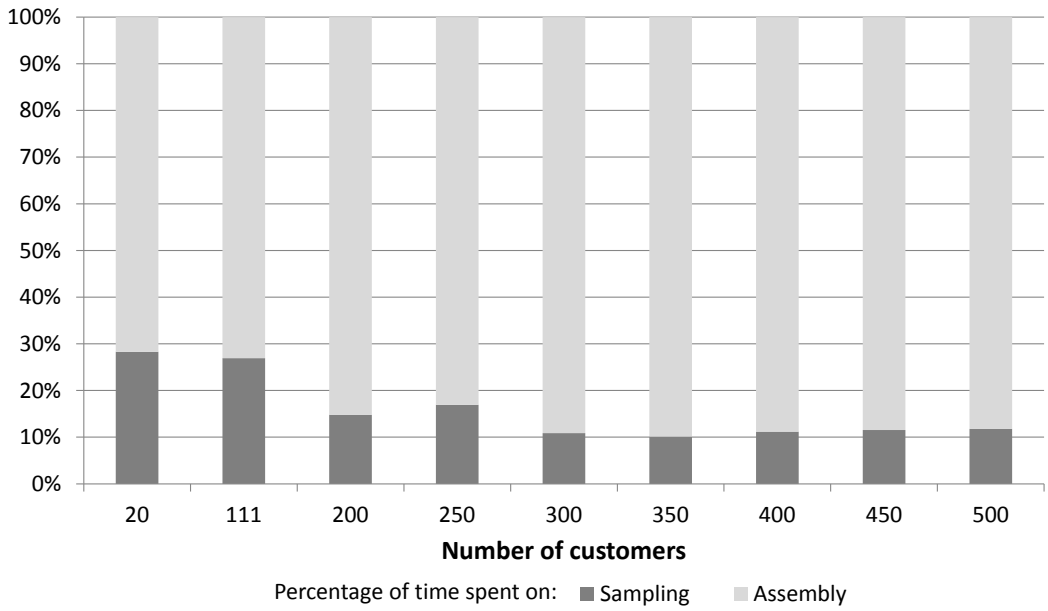


Figure 2.6 – Percentage share of CPU time by MSH phases

results show that, for all instances, assembly is the more time consuming phase, and that the percentage of time spent on the assembly phase increases with the instance size. However, the assembly phase contributes to improve the solutions on average by 4.10%, which is key to obtain competitive results with respect to existing approaches in the literature.

2.5 Conclusion

This chapter proposes a multi-space sampling heuristic for the Green VRP. This approach has three main components: a set of three randomized TSP heuristics, a tour partitioning procedure, and a set partitioning formulation. At the core of the tour partitioning procedure lies a novel repair mechanism that optimally inserts visits to refueling stations to restore the feasibility of routes violating the vehicle's fuel constraint. This procedure solves a constrained shortest path problem on an auxiliary repair graph, that by construction includes only the trips between refueling stations that respect the vehicle's fuel constraint.

The procedure is based on a reformulation of the repair problem as a constrained shortest path problem where i) the maximum travel time of a route is the constrained resource, and ii) each path from the source node to the sink node in the underlying graph models a possible way to repair the route by inserting visits to refuelling stations.

We tested our approach on a 52-instance public testbed for the Green VRP. Our approach found 8 new BKSs for the testbed and matched another 40. When compared to state-of-the-art metaheuristics, the multi-space sampling heuristic obtains competitive results in terms of solution quality and computational time, and it is one of the simplest methods used to solve the Green VRP.

2.6 Nomenclature

2.6.1 Notation for problem definition

G: Green VRP underlying graph $G = (V, E)$.
V: Set of vertices of G.
0: Depot
I: Subset of V representing customers.
F: Subset of V representing alternative fuel stations.
E: Set of edges in G.
d_{ij}: Distance between vertices i and j ($i, j \in V$).
t_{ij}: Travel time between vertices i and j ($i, j \in V$).
n: Number of customers.
a: Number of alternative fuel stations.
T_{max}: Tour duration constraint.
cr: Fuel consumption rate.
τ_i: Service time at a vertex. If $i \in I$ then τ_i is service time at customer; if $i \in F \cup \{0\}$ then τ_i is refueling time at AFS.
Q: Tank capacity.

2.6.2 Notation for multi-space sampling heuristic

G^*: Acyclic graph for the split procedure $G^* = (V^*, A)$.
V^*: Set of vertices of G^*.
A: Set of arcs of G^*.
$\mathcal{K}$: Set of TSP-like tours.
$\mathcal{R}$: Set of all feasible routes.
Ω: Subset of $\mathcal{R}$.
$\mathcal{S}$: Set of all feasible solutions to Green VRP.
$\mathcal{H}$: Set of sampling heuristics.
K: Number of iterations in sampling phase.
tsp: TSP tour ($tsp \in \mathcal{K}$).
s: Green VRP solution ($s \in \mathcal{S}$).
$\mathcal{W}$: Set of customers visited by tsp.
$\mathcal{N}$: Ordered set of nonrouted customers.
L: Randomization factor of each heuristic.
$f(s)$: Objective function of solution s.
s^*: Best solution found during execution of method.
λ: Positive parameter.
c_r: Total distance of route r.
tp_r: Total time of route r.
f_r: Binary variable equal to 1 if route is feasible.
$label_i$: Shortest path label i.
$pred_i$: Predecessor labels i.

2.6.3 Notation for repair procedure

B: Repair graph $B = (Z, U)$.
Z: Set of vertices of B.
U: Set of arcs of B.
Π: Fuel-infeasible route.

$[i, k]$: Vertex of set Z, which represents visit to station $k \in F$ after visit to vertex r_i (i.e., ith element in route Π).
$\bar{c}$: Arc cost of repair graph.
$\bar{t}$: Arc time of repair graph.

Chapter 3

A comparative study of charging assumptions in electric vehicle routing problems

The work presented in this chapter has been submitted in the Transportation Research Part B: Methodological Journal. The full reference follows:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A hybrid metaheuristic for the electric vehicle routing problem with partial charging and nonlinear charging function. Transportation Research Part B: Methodological. Under second round review.

The work of this chapter was presented at the following conferences and seminars:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2015). The electric vehicle routing problem with non-linear charging functions. In ODYSSEUS 2015, Ajaccio, France.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A comparative study of charging assumptions in electric vehicle routing problems. In ROADEF 2016, Compiègne, France.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A comparative study of charging assumptions in electric vehicle routing problems. In EURO 2016, Poznan, Poland.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Contributions to the electric vehicle routing problems. PGMO Seminars, L'École polytechnique, Paris, France, March 18, 2016
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Electric vehicle routing problems with non-linear charging functions. Cirrelt seminar, Université de Montréal Campus, Montreal, Canada, October 18, 2016

3.1 Introduction

As mentioned in the Chapter 1, because of the short driving range, eVRP solutions frequently include routes with planned detours to charging stations (CSs) where the electric vehicles (EVs) recharge. In general, eVRP models make assumptions about the capacity of the CSs, the EV energy consumption, and the EV battery charging process. This chapter focuses on the latter. The objective is to evaluate the impact of different *charging assumptions* commonly used in the literature on the quality and feasibility of eVRP solutions.

The remainder of this chapter is organized as follows. Section 3.2 reviews the charging assumption commonly used in the literature. Section 3.3 presents a mixed-integer linear programming (MILP) formulation for a particular eVRP. With mild modifications, the model can be adapted to work with different charging assumptions. Section 3.4 presents a computational experiment to compare the battery charging assumptions. Finally, Section 3.5 concludes the chapter. For easy reference, Section 3.6 summarizes the notation of this chapter.

3.2 Charging asumption in the literature

To model the battery charging process, eVRP models make assumptions about the *charging policy* and the *charging function approximation*. The former defines how much of the battery capacity can be (or must be) restored when an EV visits a CS, and the latter models the relationship between battery charging time and charging level. With respect to the charging policies, the eVRP literature can be classified into two groups: studies assuming *full* and *partial* charging policies. As the name suggests, in full charging policies, the battery capacity is fully restored every time an EV reaches a CS. On the other hand, in partial charging policies, the amount of charge (and thus the time spent at each charging point) is a decision variable.

In general, the charging functions are nonlinear, because the terminal voltage and current change during the charging process. This process is divided into two phases. In the first phase, the charging current is held constant, and thus the state of charge (SOC) increases linearly with time until the battery's terminal voltage increases to a specific maximum value (see Figure 3.1). In the second phase, the current decreases exponentially and the terminal voltage is held constant to avoid battery damages. The SOC increases then concavely with the time (Pelletier et al. 2015).

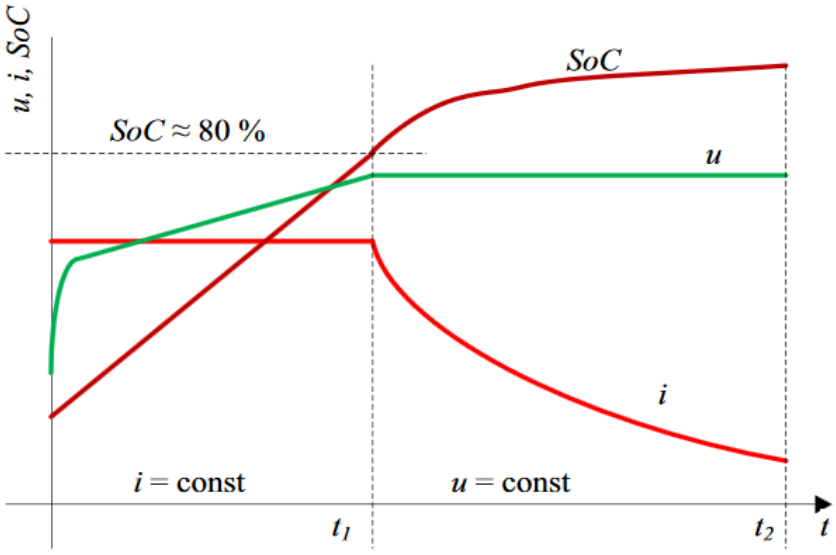


Figure 3.1 – Typical charging curve, where i and u represent the current and terminal voltage respectively. (Source Hõimoja et al. (2012))

Although the shape of the charging functions are known, their exact modeling is very complex because it depends on many factors as: current, voltage, self-recovery and temperature, among others (Wang et al. 2013). The battery state of charge is then often described by differential equations. Since such equations are difficult to integrate to optimization models for transportation problems, researchers rely on approximations of the actual charging functions. We discuss these approximations bellow. For each of them, we present a graphical comparison with respect to real charging data provided by Uhrig et al. (2015). These authors conducted experiments to estimate the charging time for different charge levels with two types of EVs and three types of CSs. It is worth noting that each approximation can be used in a full charge policy (FC) or in a partial charge policy (PC).

First segment (FS) : To avoid dealing with the nonlinear segment, Bruglieri et al. (2014) use a linear approximation that considers only the first segment (Figure 3.2).

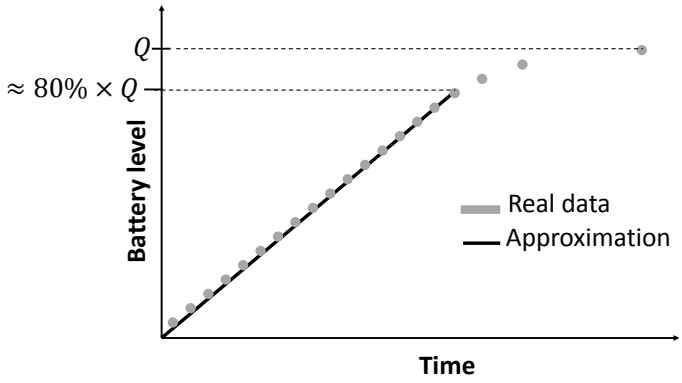
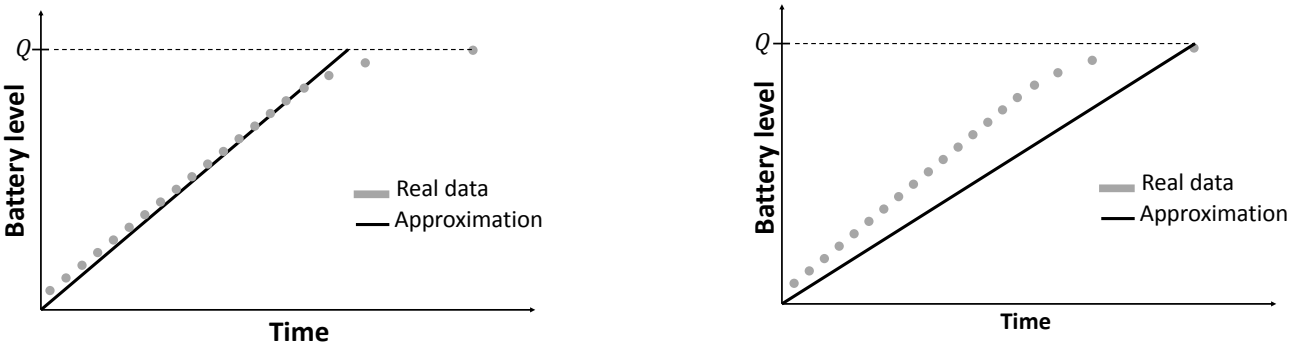


Figure 3.2 – First segment approximation (FS)

Linear approximations (L1 and L2) : Although several authors assume a linear approximation (Felipe et al. 2014, Sassi et al. 2014, Bruglieri et al. 2015, Desaulniers et al. 2014, Schiffer & Walther 2015, Keskin & Čatay 2016), they do not explain how the approximation is estimated. Two options can be considered. In the first one (L1) the charging rate of the function corresponds to the slope of the first segment of the piecewise linear approximation (see Figure 3.3a). This approximation is optimistic, because it assumes that batteries charge up to Q faster than they do in reality. In the second approximation (L2) the charging rate is the slope of the line connecting the first and last observations (see Figure 3.3b) of the charging curve. This approximation tends to be pessimistic, because over a large portion of the curve, the charging rate is slower than in reality.



(a) Linear approximation 1 (L1)

(b) Linear approximation 2 (L2)

Figure 3.3 – Linear approximations of charging functions.

Piecewise linear approximations (PL) : This approximation, proposed by Zündorf (2014) for a shortest path problem with EVs consists in approximating the charging function by a series of linear

segments. This approximation is the closest to the real data (see Figure 3.4). To assess the quality of this approximation, we fit piecewise linear functions to the Uhrig et al. (2015) data and obtained approximations with an average relative absolute error of 0.90%, 1.24%, and 1.90% for CSs of 11, 22, and 44 kW, respectively.

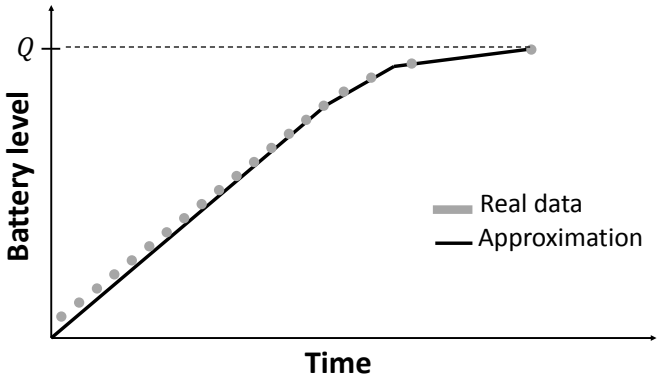


Figure 3.4 – Approximation vs real data

3.3 Setting up the study: problem and formulations

To accomplish our study about the impact of charging assumptions commonly used in the literature on the quality and feasibility of eVRP solutions, we need two elements: an particular version of an eVRP and a solution method of the problem. Note that the current charging function approximations are particular cases of the piecewise linear. Therefore, if we define an eVRP with piecewise linear charging function and select a method to solve it, we have all the elements to accomplish the study. We thus introduce in this section the eVRP with non-linear charging function (eVRP-NL) (which has never been studied in the literature), and use its MILP formulation and a commercial solver as solution method. The remainder of this subsection describes formally the eVRP-NL and introduces its MILP formulation.

3.3.1 Problem description

Formally, the eVRP-NL can be defined on a directed and complete graph $G = (V, A)$. The vertex set $V = \{0\} \cup I \cup F$ is made up of a depot (vertex 0), a set of customers I, and a set of CSs F. Each customer $i \in I$ has a service time p_i. Each CS $i \in F$ has a nonlinear charging function, which is modeled using a piecewise linear approximation. This approximation is defined by a set of breakpoints B, where each breakpoint $k \in B$ is associated to a charging time c_{ik} and a charge level a_{ik} (see Figure 3.6a). The set $A = \{(i, j) : i, j \in V, i \neq j\}$ corresponds to arcs connecting vertices in V. Each arc (i, j) has two associated nonnegative values: a travel time t_{ij} and an energy consumption e_{ij}. The customers are served using an unlimited and homogeneous fleet of EVs. All the EVs have a battery of capacity Q (expressed in kWh) and a maximum tour duration T_{max}. It is assumed that the EVs leave the depot with a fully charged battery, and that all the CSs can handle an unlimited number of EVs simultaneously. Feasible solutions to this eVRP satisfy the following conditions: each customer is visited exactly once; each route satisfies the maximum-duration limit; each route starts and ends at the depot; and the battery level when an EV arrives to and departs from any vertex is between 0 and Q.

Since the distance is directly related to the energy consumption, most work on eVRPs with homogeneous fleet focuses on minimizing the total distance (Schneider et al. 2014, Desaulniers et al. 2014, Hiermann et al. 2016, Keskin & Čatay 2016). However, this objective function neglects the impact of charging operations in the cost of the solutions. This may lead to decisions such as: charging the batteries more than needed, or charging the batteries when their level is high. These decisions directly affect the battery long-term degradation cost (which can be as high as 3 times the energy cost according to Becker et al.

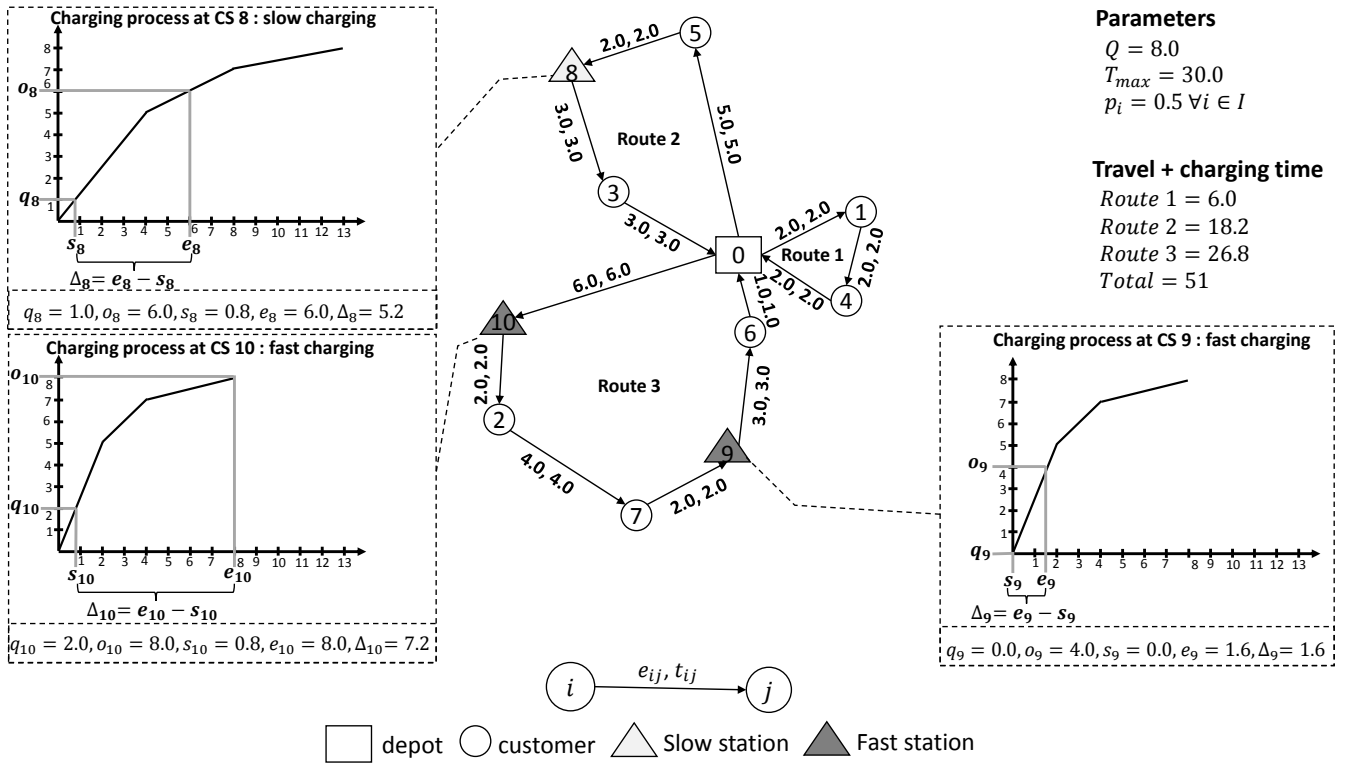


Figure 3.5 – Example of a feasible eVRP-NL solution

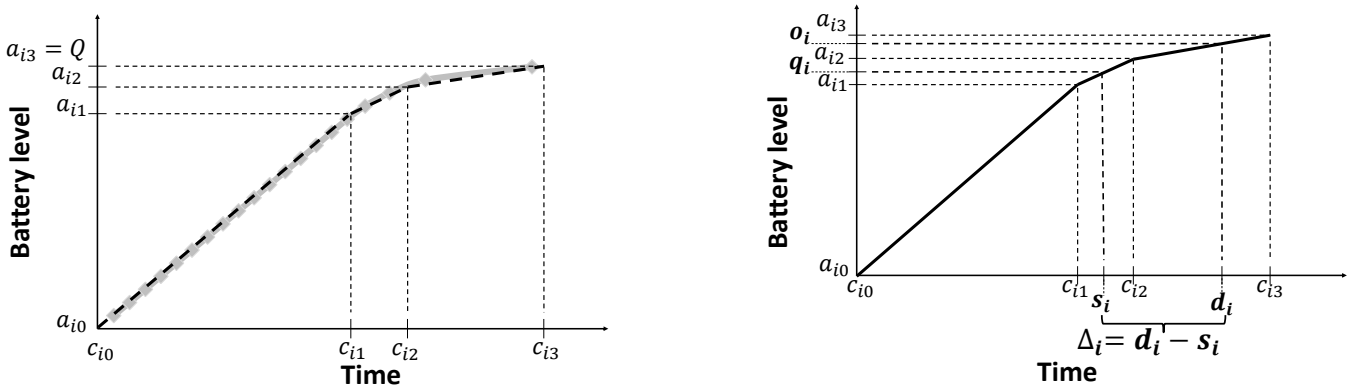
(2009)) or the charging fees at CSs (Bansal 2015). To better capture the impact of charging operations, in the eVRP-NL we minimize the total travel and charging time. This objective function has been studied by Zündorf (2014) and Liao et al. (2016) on related routing problems.

Figure 3.5 presents a numerical example illustrating the eVRP-NL. The figure depicts a solution to an instance with 7 customers and 3 CSs. The CSs have different technologies (slow and fast), and each technology has a particular piecewise linear charging function. In the charging functions, variables q_i and o_i specify the battery levels when an EV arrives at and departs from CS $i \in F$. The charging function maps these variables to charging times s_i and e_i, in order to estimate the time spent at the CS (Δ_i). In this example, Route 1 does not visit any CS, because its total energy consumption is less than the battery capacity. On the other hand Route 2 visits CS 8. In this route, the EV arrives at the CS with a battery level $q_8 = 1.0$, and it charges the battery to a level $o_8 = 6.0$. To estimate the time spent at the CS, we use the piecewise linear charging function: the charging time associated to q_8 and o_8 are $s_i = 0.8$ and $e_i = 6.0$, then the time spent at CS 8 is $\Delta_8 = 6.0 - 0.8 = 5.2$. The duration of Route 2 is the sum of the travel time (13.0), the charging time (5.2), and the service time (1.0), that is, 19.2 which is lower than $Tmax$. The cost of this route is 18.2 (travel time + charging time). Finally, Route 3 visits CSs 10 and 9; and it spends $\Delta_{10} = 7.2$ and $\Delta_9 = 1.6$ time units charging in these CSs, respectively.

3.3.2 MILP formulation

To formulate the eVRP-NL, we introduce set F' containing the set F and β copies of each CS (i.e., $|F'| = |F| \times (1 + \beta)$). The value of $1 + \beta$ corresponds to the number of times that each CS can be visited. For this MILP formulation, we use the following decision variables: variable x_{ij} is equal to 1 if an EV travels from vertex i to j, and 0 otherwise. Variables τ_j and y_j track the time and charge level when the EV departs from vertex $j \in V$. Variables q_i and o_i specify the charge levels when an EV arrives at and departs from CS $i \in F'$, and s_i and d_i are the associated charging times (see Figure 3.6b). Variable $\Delta_i = d_i - s_i$ represents the time spent at CS $i \in F'$. Variables z_{ik} and w_{ik} are equal to 1 if the charge level is between $a_{i,k-1}$ and a_{ik}, with $k \in B \setminus \{0\}$, when the EV arrives at and departs from CS $i \in F'$ respectively. Finally,

variables α_{ik} and λ_{ik} are the coefficients of the breakpoint $k \in B$ in the piecewise linear approximation, when the EV arrives at and departs from CS $i \in F'$ respectively. The MILP formulation follows:



(a) Paramenters of the approximation

(b) Battery charge levels and charging times $i \in F'$

Figure 3.6 – Piecewise linear approximation for the charging function.

$$\min \sum_{i,j \in V} t_{ij} x_{ij} + \sum_{i \in F'} \Delta_i \tag{3.1}$$

subject to

$$\sum_{j \in V, i \neq j} x_{ij} = 1, \qquad \forall i \in I \tag{3.2}$$

$$\sum_{j \in V, i \neq j} x_{ij} \leq 1, \qquad \forall i \in F' \tag{3.3}$$

$$\sum_{j \in V, i \neq j} x_{ji} - \sum_{j \in V, i \neq j} x_{ij} = 0, \qquad \forall i \in V \tag{3.4}$$

$$e_{ij} x_{ij} - (1 - x_{ij})Q \leq y_i - y_j \leq e_{ij} x_{ij} + (1 - x_{ij})Q, \qquad \forall i \in V, \forall j \in I \tag{3.5}$$

$$e_{ij} x_{ij} - (1 - x_{ij})Q \leq y_i - q_j \leq e_{ij} x_{ij} + (1 - x_{ij})Q, \qquad \forall i \in V, \forall j \in F' \tag{3.6}$$

$$y_i \geq e_{i0} x_{i0}, \qquad \forall i \in V \tag{3.7}$$

$$y_i = o_i, \qquad \forall i \in F' \tag{3.8}$$

$$y_0 = Q \tag{3.9}$$

$$q_i \leq o_i, \qquad \forall i \in F' \tag{3.10}$$

$$q_i = \sum_{k \in B} \alpha_{ik} a_{ik}, \qquad \forall i \in F' \tag{3.11}$$

$$s_i = \sum_{k \in B} \alpha_{ik} c_{ik}, \qquad \forall i \in F' \tag{3.12}$$

$$\sum_{k \in B} \alpha_{ik} = \sum_{k \in B} z_{ik}, \qquad \forall i \in F' \tag{3.13}$$

$$\sum_{k \in B} z_{ik} = \sum_{j \in V} x_{ij}, \qquad \forall i \in F' \tag{3.14}$$

$$\alpha_{ik} \leq z_{ik} + z_{i,k+1}, \qquad \forall i \in F', \forall k \in B \setminus \{b\} \tag{3.15}$$

$$\alpha_{ib} \leq z_{ib}, \qquad \forall i \in F' \tag{3.16}$$

$$o_i = \sum_{k \in B} \lambda_{ik} a_{ik}, \qquad \forall i \in F' \tag{3.17}$$

$$d_i = \sum_{k \in B} \lambda_{ik} c_{ik}, \qquad \forall i \in F' \tag{3.18}$$

$$\sum_{k \in B} \lambda_{ik} = \sum_{k \in B} w_{ik}, \qquad \forall i \in F' \tag{3.19}$$

$$\sum_{k\in B} w_{ik} = \sum_{j\in V} x_{ij}, \qquad \forall i \in F' \tag{3.20}$$

$$\lambda_{ik} \leq w_{ik} + w_{i,k+1}, \qquad \forall i \in F,' \forall k \in B \setminus \{b\} \tag{3.21}$$

$$\lambda_{ib} \leq w_{ib}, \qquad \forall i \in F' \tag{3.22}$$

$$\Delta_i = d_i - s_i, \qquad \forall i \in F' \tag{3.23}$$

$$\tau_i + (t_{ij} + p_j)x_{ij} - T_{max}(1 - x_{ij}) \leq \tau_j, \qquad \forall i \in V, \forall j \in I \tag{3.24}$$

$$\tau_i + \Delta_j + t_{ij}x_{ij} - (S_{max} + T_{max})(1 - x_{ij}) \leq \tau_j, \qquad \forall i \in V, \forall j \in F' \tag{3.25}$$

$$\tau_j + t_{j0} \leq T_{max}, \qquad \forall j \in V \tag{3.26}$$

$$\tau_0 \leq T_{max} \tag{3.27}$$

$$x_{ij} = 0, \qquad \forall i, j \in F' : m_{ij} = 1 \tag{3.28}$$

$$\tau_i \geq \tau_j, \qquad \forall i, j \in F' : m_{ij} = 1, j \leq i \tag{3.29}$$

$$\tau_j \leq T_{max} \sum_{i\in V} x_{ij}, \qquad \forall j \in F' \tag{3.30}$$

$$\sum_{i\in V} x_{ih} \geq \sum_{j\in V} x_{jf}, \qquad \forall h, f \in F' : m_{hf} = 1, h \leq f \tag{3.31}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i, j \in V \tag{3.32}$$

$$\tau_i \geq 0, y_i \geq 0 \qquad \forall i \in V \tag{3.33}$$

$$z_{ik} \in \{0, 1\}, w_{ik} \in \{0, 1\}, \alpha_{ik} \geq 0, \lambda_{ik} \geq 0, \qquad \forall i \in F', \forall k \in B \tag{3.34}$$

$$q_i \geq 0, o_i \geq 0, s_i \geq 0, d_i \geq 0, \Delta_i \geq 0, \qquad \forall i \in F' \tag{3.35}$$

The objective function (4.1) seeks to minimize the total time (travel times plus charging times). Constraints (3.2) ensure that each customer is visited once. Constraints (3.3) ensure that each CS is visited at most once. Constraints (3.4) impose the flow conservation. Constraints (3.5) and (3.6) track the battery charge level at each vertex. Constraints (3.7) ensure that, if the EV travels between a vertex and the depot, it has sufficient energy to reach its destination. Constraints (3.8) reset the battery tracking to o_i upon departure from CS $i \in F'$. Constraint (3.9) ensures that the battery charge level is Q at the depot. Constraints (3.10) couple the charge levels when an EV arrives at and departs from any CS. Constraints (3.11–3.16) define the charge level (and its corresponding charging time) when an EV arrives at CS $i \in F'$ (based on the piecewise linear approximation of the charging function). Similarly, constraints (3.17–3.22) define the charge level (and its corresponding charging time) when an EV departs from CS $i \in F'$. Constraints (3.23) define the time spent at any CS. Constraints (3.24) and (3.25) track the departure time at each vertex, where $S_{max} = max_{i\in F'}\{c_{ib}\}$. Constraints (3.26) and (3.27) ensure that the EVs return to the depot no later than T_{max}. Constraints (3.28) and (3.31) help to avoid the symmetry generated by the copies of the CSs. The parameter m_{ij} is equal to 1 if i and $j \in F'$ represent the same CS. Finally, constraints (3.32–3.35) define the domain of the decision variables.

This MILP formulation can be easily adapted to model the other approximations (FS, L1, and L2) and the different charging policies (FC and PC):

— Full charge and piecewise linear approximations (FC-PL): we replace constraints (3.17) and (3.18) by

$$o_i = \sum_{k\in B} \lambda_{ik} a_{ib}, \forall i \in F' \tag{3.36}$$

$$e_i = \sum_{k\in B} \lambda_{ik} c_{ib}, \forall i \in F' \tag{3.37}$$

— Partial charge using the first segment (PC-FS): To run our MILP formulation with this assumption, we modify the input data to include only the first segment.

— Partial charge and linear approximations (PC-L): To run our MILP formulation with PC-L1 and PC-L2, we modify the input data so that in the piecewise linear approximation there is a single segment with the corresponding charging rate.

3.4 Computational experiments

3.4.1 Experimental settings & Test instances

We conducted an experiment to compare the different charging assumptions commonly used in the literature, combining charging policies and charging function approximation. The evaluated assumption are: partial charging and piecewise linear approximation (PC-PL), full charging and piecewise linear approximation (FC-PL), partial charging and first segment (PC-FS), Partial charge and linear approximations (PC-L1 and PC-L2).

With the aim that the conclusions of the comparison are independent of the solution method and consequently of the quality solution, we just compare optimal solutions of the evaluated assumptions. To ensure the optimal solution we solve a set of small instances.

We ran the experiments on a set of 20 randomly generated instances with 10 customers and 2 or 3 CSs. We located the customers using either a random uniform distribution, a random clustered distribution, or a mixture of both. The CSs are located using two criteria: randomly or using a simple p-median heuristic. To generate the charging functions approximations we use the real charging data for a 16 kWh battery provided by Uhrig et al. (2015). Finally, the maximum route duration for every instance was fixed to 10 hours.

3.4.2 Experimental environment & Parameter setting

As mentioned in the last section, the MILP formulation uses β copies of the CSs to model multiple visits to the same CS. Although several authors followed this strategy (Conrad & Figliozzi 2011, Erdoğan & Miller-Hooks 2012, Schneider et al. 2014, Sassi et al. 2014, Goeke & Schneider 2015, Hiermann et al. 2016), they do not explain how the value of β is set. It is worth noting that β plays an important role in the definition of the solution space, and therefore it restricts the optimal solution of the model. For instance, an optimal solution found with $\beta = 3$ may not be optimal for $\beta = 4$. In practice, there is no restriction on the number of times that a CS can be visited, but large values of β result in models that are computationally intractable. To overcome this difficulty, we designed an iterative procedure to solve the MILP formulation for increasing values of β. Starting with $\beta = 0$, at each iteration, our procedure (i) tries to solve the MILP formulation to optimality with a time limit of 100 h, and (ii) sets $\beta = \beta + 1$. The procedure stops when the time limit is reached or an iteration ends with a solution s_β satisfying $f(s_\beta) = f(s^*_{\beta-1})$, where $f(\cdot)$ denotes the objective function and $*$ an optimal solution. We implemented our MILP formulation (and its variant) in Gurobi 5.6

3.4.3 Results

Table 3.1 presents the results. Considering that the PC-PL assumption is the closest assumption to the reality, the results obtained using the others assumptions are compared with reference to the results of the PC-PL. For each charging assumption, we give the objective function value (of), the percentage gap between of and the PC-PL solution (G), the number of routes in the solution (r), and the value of β. Since in practice the charging time is controlled by the nonlinear charging function, the charging decisions of the PC-L solutions are evaluated a posteriori using the piecewise linear approximation. The last rows of Table 3.1 summarize the results. We present, for each assumption, the average and maximum percentage gap, the number of solutions employing more EVs than in the PC-PL solution, and the number of infeasible solutions.

The results show that solutions based on the full charging policy perform poorly in terms of both objective function (+20.11% on average) and number of routes (8 solutions use a larger fleet) with respect to those based on the partial charging policy. This is because the EVs spend more time than necessary at the CSs. The main motivation for a full charging policy is to avoid complex charging-quantity decisions. However, according to our results, the gain in simplicity does not offset the loss of solution quality.

In the PC-FS assumption EVs can charge their batteries up to only around 80% of the actual capacity. Artificially constraining the capacity may force EVs to detour to CSs more often than necessary when traveling to distant customers. Because the maximum route duration is limited, the time spent detouring and recharging the battery reduces the number of customers that can be visited. Consequently, more routes may be needed to service the same number of customers. Our results confirm this intuition: in 3 out of the 20 instances the PC-FS assumption increases the number of routes. Furthermore, in practice some distant customers may not be included in routes unless the EVs can fully use their battery capacity. In our experiments, 9 instances become infeasible under PC-FS. In conclusion, although PC-FS simplifies the problem (avoiding the nonlinear segment of the charging function) it may lead to solutions that are infeasible, or with larger fleets and (on average) 2.70% more expensive.

As mentioned before, PC-L1 assumes that batteries charge faster than they do in reality (Figure 3.3a). As a consequence, routes based on PC-L1 may in practice need more time to reach the planned charge levels. The extra time may make a route infeasible if there is little slack in the duration constraint. Indeed, the post-hoc evaluation shows that for 14 instances, the PC-L1 solutions are infeasible in practice. On the other hand, PC-L2 assumes that batteries charge slower than in reality (Figure 3.3b). Overestimating the charging times does not lead to feasibility issues, but the resulting routes may be overly conservative. For instance, in our experiments PC-L2 leads to solutions that are (on average) 1.45% more expensive, and it increases the number of routes in 2 instances.

Table 3.1 – Comparison of our charging assumptions with charging assumptions from the literature

Instance	PC-PL			FC-PL				PC-FS				PC-L1							PC-L2						
												Solution			Evaluation				Solution			Evaluation			
	of	*r*	*β*	*of*	*G*(%)	*r*	*β*	*of*	*G*(%)	*r*	*β*	*of*	*r*	*β*	*of*	*G*(%)	*r*	*β*	*of*	*r*	*β*	*of*	*G*(%)	*r*	*β*
tc0c10s2cf1	**19.75**	3	2	20.82	5.42	3	2	NFS	NFS	NFS	NFS	19.61	3	2	NFE	NFE	NFE	NFE	20.50	3	2	20.22	2.38	3	2
tc0c10s2ct1	**12.30**	2	0	12.53	1.87	2	0	12.61	2.52	3	0	12.22	2	0	12.42	0.98	2	0	12.46	2	0	**12.30**	0.00	2	0
tc0c10s3cf1	**19.75**	3	2	20.82	5.42	3	2	NFS	NFS	NFS	NFS	19.61	3	2	NFE	NFE	NFE	NFE	20.50	3	2	20.22	2.38	3	2
tc0c10s3ct1	**10.80**	2	0	11.10	2.78	2	0	**10.80**	0.00	2	0	10.79	2	0	11.03	2.13	2	0	10.97	2	0	**10.80**	0.00	2	0
tc1c10s2cf2	**9.03**	3	0	9.19	1.77	3	0	**9.03**	0.00	3	0	9.03	3	0	9.12	1.00	3	0	9.14	3	0	**9.03**	0.00	3	0
tc1c10s2cf3	**16.37**	3	2	21.33	30.30	3	2	NFS	NFS	NFS	NFS	15.99	3	1	NFE	NFE	NFE	NFE	16.89	3	2	**16.37**	0.00	3	2
tc1c10s2cf4	**16.10**	3	2	25.31	57.20	4	3	NFS	NFS	NFS	NFS	15.66	3	2	NFE	NFE	NFE	NFE	16.43	3	2	16.23	0.81	3	2
tc1c10s2ct2	**10.75**	3	1	11.14	3.63	3	0	**10.75**	0.00	3	1	10.75	3	0	10.76	0.09	3	0	10.94	3	0	10.78	0.28	3	0
tc1c10s2ct3	**13.17**	2	2	22.76	72.82	3	3	15.98	21.34	3	2	13.06	2	2	NFE	NFE	NFE	NFE	13.60	2	2	**13.17**	0.00	2	2
tc1c10s2ct4	**13.83**	2	1	17.61	27.33	3	1	NFS	NFS	NFS	NFS	13.34	2	1	NFE	NFE	NFE	NFE	14.17	2	1	14.17	2.46	2	1
tc1c10s3cf2	**9.03**	3	0	9.19	1.77	3	0	**9.03**	0.00	3	0	9.03	3	0	9.12	1.00	3	0	9.14	3	0	**9.03**	0.00	3	0
tc1c10s3cf3	**16.37**	3	1	21.33	30.30	3	2	NFS	NFS	NFS	NFS	15.99	3	1	NFE	NFE	NFE	NFE	16.89	3	2	**16.37**	0.00	3	2
tc1c10s3cf4	**14.90**	3	1	18.43	23.69	4	0	NFS	NFS	NFS	NFS	14.56	2	1	NFE	NFE	NFE	NFE	15.18	3	0	15.18	1.88	3	0
tc1c10s3ct2	**9.20**	3	0	11.14	21.09	3	0	**9.20**	0.00	3	0	9.19	3	0	NFE	NFE	NFE	NFE	10.80	3	0	10.57	14.89	3	0
tc1c10s3ct3	**13.02**	2	0	17.06	31.03	3	0	13.07	0.38	2	1	12.98	2	0	13.16	1.08	2	0	13.60	2	0	**13.02**	0.00	2	0
tc1c10s3ct4	**13.21**	2	0	15.54	17.64	3	1	13.58	2.80	3	1	12.92	2	1	NFE	NFE	NFE	NFE	13.71	2	0	**13.21**	0.00	2	0
tc2c10s2cf0	**21.77**	3	3	25.24	15.94	4	2	NFS	NFS	NFS	NFS	14.53	2	2	NFE	NFE	NFE	NFE	22.78	4	4	22.15	1.75	4	4
tc2c10s2ct0	**12.45**	3	2	15.05	20.88	3	3	**12.45**	0.00	3	2	12.44	3	3	NFE	NFE	NFE	NFE	12.93	3	2	**12.45**	0.00	3	2
tc2c10s3cf0	**21.77**	3	2	25.24	15.94	4	2	NFS	NFS	NFS	NFS	14.53	2	2	NFE	NFE	NFE	NFE	23.02	4	3	22.20	1.98	4	3
tc2c10s3ct0	**11.51**	3	0	13.27	15.29	2	0	**11.51**	0.00	3	0	11.50	3	0	NFE	NFE	NFE	NFE	11.92	3	0	11.54	0.26	3	0
Avg. Difference (%)				20.11				2.70							1.04							1.45			
Max. Difference (%)				72.82				21.34							2.13							14.89			
Solutions with larger fleet					8				3							0							2		
Infeasible solutions				0				9							14							0			

NFS: Non-feasible solution, NFE: Non-feasible evaluation
$G(\%) = (of - of_{PC-PL})/of_{PC-PL} \times 100$

3.5 Conclusion

In this chapter, we reviewed the different assumptions used in the literature to model the charging process in eVRPs. In order to know the impact of the charging assumption in the quality and feasibility of the solutions, we conducted a study comparing the optimal solutions for an eVRP under different battery charging assumptions. To conduct this study, we introduce the eVRP-NL, its MILP formulation (which can adapted to model different charging assumption), and a set of small instances with a real information about

the charging functions.

Our results show that the full charging policy may lead to overly expensive solutions. Furthermore, they show that the linear charging and first segment approximation have a negative impact on the quality and feasibility of the solutions. In consequence, it is advisable to consider charging function approximations that capture the nonlinear behavior of the process when modelling eVRPs.

3.6 Nomenclature

3.6.1 Notation for the MILP formulation

G: eVRP underlying graph $G = (V, A)$.
V: Set of vertices of G.
0: Depot
I: Subset of V representing customers.
F: Subset of V representing CSs.
F': Set that contain the set F and β copies of each CS.
A: Set of arcs in G.
B: Set of breakpoints of the piecewise linear charging function.
p_i: Service time at customer $i \in I$.
c_{ik}: Charging time of the CS $i \in F$ at the breakpoint $k \in B$.
a_{ik}: Charge level of the CS $i \in F$ at the breakpoint $k \in B$.
e_{ij}: Energy consumption between vertices i and j ($i, j \in V$).
t_{ij}: Travel time between vertices i and j ($i, j \in V$).
T_{max}: Tour duration constraint.
Q: Tank capacity.
x_{ij}: Binary variable, equal to 1 if an EV travels from vertex i to j and 0 otherwise.
τ_j: Depart time of the EV from vertex $j \in V$.
y_j: Charge level when the EV departs from vertex $j \in V$.
q_i: Charge level when an EV arrives at CS $i \in F'$.
o_i: Charge level when an EV departs from CS $i \in F'$.
s_i: Charging time associated to the charge level q_i.
d_i: Charging time associated to the charge level o_i.
Δ_i: the time spent at CS $i \in F'$.
z_{ik}: Binary variable, equal to 1 if the charge level is between $a_{i,k-1}$ and a_{ik}, with $k \in B \setminus \{0\}$, when the EV arrives at CS $i \in F'$
w_{ik}: Binary variable, equal to 1 if the charge level is between $a_{i,k-1}$ and a_{ik}, with $k \in B \setminus \{0\}$, when the EV departs from CS $i \in F'$.
α_{ik}: Coefficients of the breakpoint $k \in B$ in the piecewise linear approximation, when the EV arrives at CS $i \in F'$.
λ_{ik}: Coefficients of the breakpoint $k \in B$ in the piecewise linear approximation, when the EV departs from CS $i \in F'$.

Chapter 4

A hybrid metaheuristic for the electric vehicle routing problem with partial charging and nonlinear charging function

The work presented in this chapter has been submitted in the Transportation Research Part B: Methodological Journal. The full reference follows:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A hybrid metaheuristic for the electric vehicle routing problem with partial charging and nonlinear charging function. Transportation Research Part B: Methodological. Under second round review.

The work of this chapter was presented at the following conferences and seminars:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2015). The electric vehicle routing problem with non-linear charging functions. In ODYSSEUS 2015, Ajaccio, France.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Contributions to the electric vehicle routing problems. PGMO Seminars, L'École polytechnique, Paris, France, March 18, 2016
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Electric vehicle routing problems with non-linear charging functions. Cirrelt seminar, Université de Montréal Campus, Montreal, Canada, October 18, 2016

4.1 Introduction

In Chapter 3, we showed the importance of considering the partial charging and charging function approximations that capture the nonlinear behavior of the charging process in the eVRPs. However, it is necessary to propose an approach to solve industrial-sized instances of the electric vehicle routing problem with non linear charging function (eVRP-NL). We thus propose a hybrid metaheuristic combining iterated local search (ILS) and heuristic concentration (HC). The ILS component uses a variable neighborhood descent (VND) procedure for the local search phase. The VND uses three local search operators. The first two operators are classical relocate and 2-Opt moves. On the other hand, the third one is a specialized operator for improving the charging decisions. This operator relies on solving the *fixed-route vehicle-charging problem* (FRVCP). A new problem defining the charging decisions (i.e., where and how much to charge) of an energy-infeasible fixed-route (i.e., a route infeasible with respect to the energy autonomy of the vehicle). To solve the FRVCP, we propose a heuristic and a mixed-integer linear programming (MILP) formulation. Finally, the HC component assembles the final solution from the set of all routes found in the local optima reached by the ILS.

The main contributions of this chapter are threefold. First, we introduce and study a new variant of the fixed-route vehicle-charging problem (FRVCP) that appears as a subproblem of the eVRP-NL. The FRVCP consists in finding the optimal charging decisions (i.e., where and how much to charge) for a route servicing a fixed sequence of customers. Second, we propose a hybrid metaheuristic to solve the eVRP-NL and assess its performance in a new set of benchmark instances. Third, we analyze our solutions and provide some insight into the characteristics of good eVRP-NL solutions.

The remainder of this chapter is organized as follows. Section 4.2 introduces our hybrid metaheuristic. Section 4.3 discusses the FRVCP and presents two approaches to solve it. Section 4.4 presents a computational evaluation of the proposed method. Finally, Section 4.5 concludes the chapter. For easy reference, Section 4.6 summarizes the notation of this chapter.

4.2 Hybrid metaheuristic

To solve the eVRP-NL we developed a hybrid metaheuristic combining ILS (Lourenço et al. 2010) an HC (Rosing & ReVelle 1997). Figure 4.1 presents the general structure of the proposed approach (hereafter referred to as ILS+HC).

To find an initial solution we follow a sequence-first split-second approach which uses a constructive heuristic to build a TSP tour visiting all the customers and a splitting procedure to retrieve an eVRP-NL solution. Then, at each iteration of the ILS we improve the current solution using a variable neighborhood descent (VND) (Mladenović & Hansen 1997) with three local search operators: relocate, 2-Opt, and global charging improvement (GCI). At the end of each ILS iteration, we update the best solution and add the routes of the local optimum to a pool of routes Ω. To diversify the search, we concatenate the routes of the local optimum to build a new TSP tour, and then perturb the new TSP tour. We start a new ILS iteration by splitting the perturbed TSP tour. After K iterations the ILS component stops, and we carry out the HC. In this phase, we solve a set partitioning problem over the set of routes Ω to obtain an eVRP-NL solution. In the remainder of this section, we describe the main components of our method.

4.2.1 Initial solution

We generate the initial TSP tour using the simple and well-known nearest neighbor heuristic (NN). For a description of NN see Rosenkrantz et al. (1974)

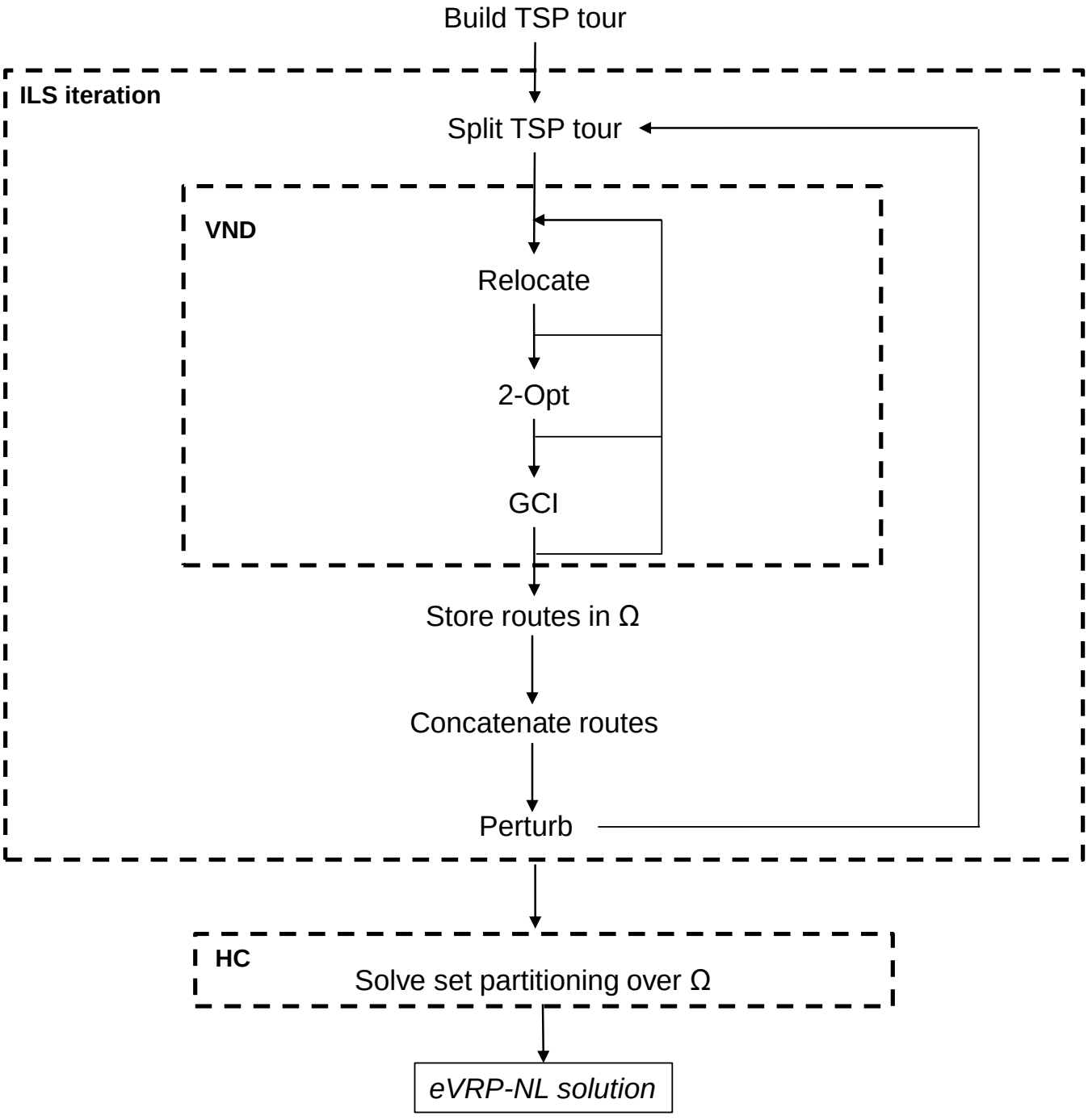


Figure 4.1 – General structure of ILS+HC

4.2.2 Split

To extract a feasible solution from a TSP tour, our approach uses an adaptation of the splitting procedure introduced by Prins (2004). The splitting procedure builds a directed acyclic graph $G^* = (V^*, A^*)$ composed of the ordered vertex set $V^* = (v_0, v_1, ..., v_i, ..., v_n)$ and the arc set A^*. Vertex $v_0 = 0$ is an auxiliary vertex, and each vertex v_i represents the customer in the ith position of the TSP tour. Arc $(v_i, v_{i+n_r}) \in A^*$ represents a feasible route $r_{v_i,v_{i+n_r}}$ with an energy consumption $e_{r_{v_i,v_{i+n_r}}}$, starting and ending at the depot and visiting customers in the sequence v_{i+1} to v_{i+n_r}.

Note that since the TSP tour only includes customers, route $r_{v_i,v_{i+n_r}}$ may be *energy-infeasible* (i.e., the total energy needed to cover the route is greater than Q). In that case, we solve a FRVCP to obtain an *energy-feasible* route by inserting visits to CSs. If visiting CSs increases the duration of the route beyond T_{max}, we do not include the arc associated with the route in A^*. Finally, to obtain a feasible eVRP-NL solution, the splitting procedure finds the set of arcs (i.e., routes) along the shortest path connecting 0 and v_n in G^*.

4.2.3 Variable neighborhood descent

To improve the solution generated by the splitting procedure we use a VND based on three local search operators. The first two operators, namely, relocate and 2-Opt[1], focus on the customer sequencing decisions. In other words, these two operators only alter the sequence of customers and do not have the ability to insert, remove, or change the position of CSs. To update the charging times after a relocate or 2-Opt move we use the rule proposed by Felipe et al. (2014): when visiting a CS, charge the strict minimum amount of energy needed to continue the route until reaching the next CS (or the depot if there is no other CS downstream). If reaching the next CS (or the depot) is impossible, even with a fully charged battery, the move is deemed infeasible. Similarly, if after updating the charging times the resulting route is infeasible in terms of the maximum-duration limit, the move is simply discarded. It is worth nothing that the Felipe et al. (2014) rule is optimal when all CSs are homogeneous; nonetheless, that is not the case in our eVRP-NL.

As its name suggest, the third operator, global charging improvement or GCI, focuses on the charging decisions. GCI is applied to every route visiting at least one CS. The operator works as follows. First, GCI removes from the route all visits to CSs. If the resulting route is energy-feasible, the operator stops. On the other hand, if the route is energy-infeasible, GCI solves a FRVCP trying to optimize the charging decisions (where and how much to charge) for the concerned route. Depending on the configuration, our ILS+HC solves the underlying FRVCP either heuristically or optimally. Full details on the FRVCP and the solution techniques embedded in our approach to solve it are given in Section 4.3.

4.2.4 Perturb

To diversify the search our approach concatenates the routes of the current best solution to build a TSP tour. Then, we perturb the resulting TSP tour with a randomized double bridge operator (Lourenço et al. 2010) and then apply the split procedure to obtain a new eVRP-NL solution.

4.2.5 Heuristic concentration

Finally, the heuristic concentration component solves a set partitioning formulation over the pool of routes Ω. The objective is then to select the best subset of routes from Ω to build the final solution guaranteeing that each customer is visited by exactly one route.

[1] In our implementation we use intra-route and inter-route versions with best-improvement selection.

4.3 The fixed-route vehicle-charging problem

As mentioned above our ILS+HC relies on solving a variant of the FRVCP at two specific points of its execution: when evaluating routes in the split procedure, and when searching the GCI neighborhood. The FRVCP is a variant of the well-known fixed-route vehicle-refueling problem (FRVRP). The FRVRP seeks the minimum-cost refueling policy (which fuel stations to visit and the refueling quantity at each visited station) for a given origin-destination route (Suzuki 2014). Most of the research carried on the FRVRP and its variants applies only to internal combustion vehicles (which have negligible refueling times). Nonetheless, a few FRVCP variants have received attention in the literature. Most of these variants assume full charging policies (Montoya et al. 2015, Hiermann et al. 2016, Liao et al. 2016). To our knowledge, only Sweda et al. (2014) assume a partial charging policy. Their problem differs from ours in three fundamental ways: i) they do not take into account the charging times (because their objective is to minimize the energy and degradation costs), ii) they do not deal with maximum route duration constraints, and iii) in their problem the CSs are already included in the fixed route and no detours are to be planned. In the remainder of this section we introduce our FRVCP and the techniques embedded in our approach to solve it. For the sake of simplicity, in the remainder of the manuscript to refer to our FRVCP variant simply as the FRVCP.

Let $\Pi = \{\pi(0), \pi(1), ..., \pi(i), ..., \pi(j), ..., \pi(n_r)\}$ be an energy-infeasible route, where $\pi(0)$ and $\pi(n_r)$ represent the depot. The route has a total time $\bar{t}$, which is the sum of the travel times plus the service times. The feasibility of Π may be restored by inserting visits to CSs. As mentioned in Section 3.3, each CS $j \in F$ has a piecewise linear charging function defined by a set of breakpoints B. The piecewise linear function is composed by a set of segments. Each segment is defined between the breakpoints $k-1$ and $k \in B$, it has a slope ρ_{jk} (representing a charging rate), and it is bounded between the battery levels a_{jk-1} and a_{jk} (see Figure 4.2a). Figure 4.2b shows the fixed-route Π and the possible visits to the CSs between two vertices of Π. In Figure 4.2b the values $e_{\pi(i-1)\pi(i)}$ and $t_{\pi(i-1)\pi(i)}$ represent the energy consumption and the travel time between vertices $\pi(i-1)$ and $\pi(i) \in \Pi$. Similarly, $e_{\pi(i-1)j}$ and $t_{\pi(i-1)j}$ represent the energy consumption and the travel time between vertex $\pi(i-1) \in \Pi$ and CS $j \in F$, and $e_{j\pi(i)}$ and $t_{j\pi(i)}$ represent the energy consumption and the travel time between the CS $j \in F$ and vertex $\pi(i) \in \Pi$.

In the FRVCP the objective is to find the charging decisions (where and how much to charge) that minimize the sum of the charging times and detour times while satisfying the following conditions: the level of the battery when the EV arrives at any vertex is nonnegative; the charge in the battery does not exceed its capacity; and the route satisfies the maximum-duration limit. Since the FRVRP is NP-hard (Suzuki 2014) and the FRVCP generalizes the FRVRP, we can conclude that the FRVCP is also NP-hard.

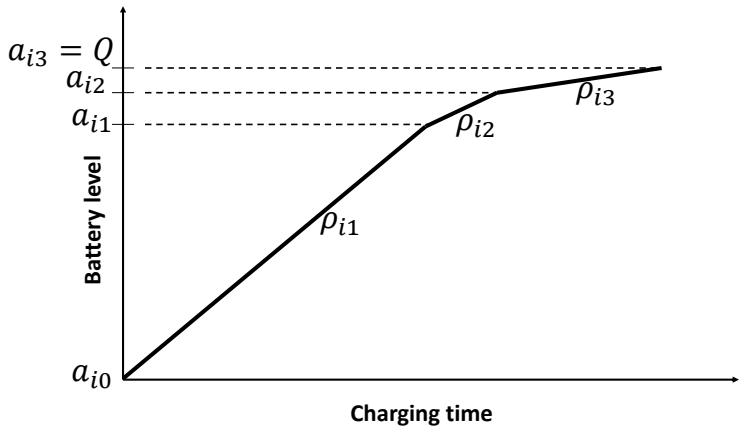
4.3.1 Mixed-integer linear programming formulation

We formulate the FRVCP using the following decision variables: variable $\varepsilon_{\pi(i)j}$ is equal to 1 if the EV charges at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$. Variable $\phi_{\pi(i)}$ tracks the battery level. If $\varepsilon_{\pi(i)j} = 0$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at vertex $\pi(i)$. On the other hand, if $\varepsilon_{\pi(i)j} = 1$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at CS $j \in F$ right before visiting vertex $\pi(i)$. Variable $\theta_{\pi(i)jk}$ is equal to 1 if the EV charges on the segment defined by breakpoints $k-1$ and $k \in B$ at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$. Finally, variables $\delta_{\pi(i)jk}$ and $\mu_{\pi(i)jk}$ are (respectively) the amount of energy charged and the battery level when the charging finishes on the segment between breakpoints $k-1$ and $k \in B$ at CS $j \in F$ before the visit to vertex $\pi(i) \in \Pi$. The MILP formulation of the FRVCP follows:

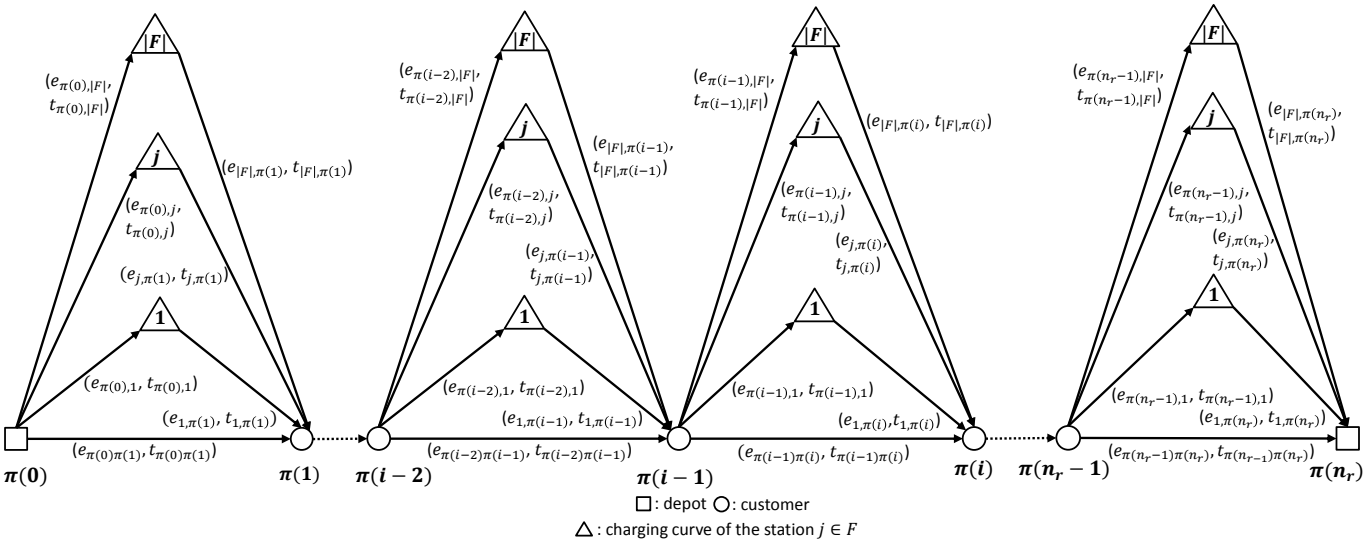
$$\min \sum_{\pi(i)\in\Pi\setminus\{\pi(0)\}} \sum_{j\in F} \sum_{k\in B\setminus\{0\}} \frac{\delta_{\pi(i)jk}}{\rho_{jk}} + \sum_{\pi(i)\in\Pi\setminus\{\pi(0)\}} \sum_{j\in F} \varepsilon_{\pi(i)j}(t_{\pi(i-1)j} + t_{j\pi(i)} - t_{\pi(i-1)\pi(i)}) \tag{4.1}$$

Subject to

$$\phi_{\pi(1)} = Q - \sum_{j\in F} \varepsilon_{\pi(1)j} e_{\pi(0)j} - e_{\pi(0)\pi(1)} \left(1 - \sum_{j\in F} \varepsilon_{\pi(1)j}\right) \tag{4.2}$$



(a) Segments of the piecewise linear charging function



(b) Fixed-route

Figure 4.2 – Piecewise linear charging function and fixed-route for the FRVCP

$$\phi_{\pi(i)} = \phi_{\pi(i-1)} + \sum_{j\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(i-1)jk} - \sum_{j\in F} \varepsilon_{\pi(i-1)j} e_{j\pi(i-1)} -$$

$$\sum_{j\in F} \varepsilon_{\pi(i)j} e_{\pi(i-1)j} - e_{\pi(i-1)\pi(i)} \left(1 - \sum_{j\in F} \varepsilon_{\pi(i)j}\right) \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0), \pi(1), \pi(n_r)\} \qquad (4.3)$$

$$\phi_{\pi(n_r)} = \phi_{\pi(n_r-1)} + \sum_{j\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r-1)jk} +$$

$$\sum_{j\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r)jk} - \sum_{j\in F} \varepsilon_{\pi(n_r-1)j} e_{j\pi(n_r-1)} -$$

$$\sum_{j\in F} \varepsilon_{\pi(n_r)j} (e_{\pi(n_r-1)j} + e_{j\pi(n_r)}) - e_{\pi(n_r-1)\pi(n_r)} \left(1 - \sum_{j\in F} \varepsilon_{\pi(n_r)j}\right) \qquad (4.4)$$

$$\phi_{\pi(n_r-1)} + \sum_{j\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r-1)jk} - \sum_{j\in F} e_{j\pi(n_r-1)} \varepsilon_{\pi(n_r-1)j} -$$

$$\sum_{j\in F} e_{\pi(n_r-1)j} \varepsilon_{\pi(n_r)j} \geq 0 \qquad (4.5)$$

$$\mu_{\pi(i)j1} = \phi_{\pi(i)} + \delta_{\pi(i)j1} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F \qquad (4.6)$$

$$\mu_{\pi(n_r)j1} = \phi_{\pi(n_r-1)} + \sum_{l\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r-1)lk} -$$

$$\sum_{l\in F} e_{l\pi(n_r-1)} \varepsilon_{\pi(n_r-1)l} - \varepsilon_{\pi(n_r)j} e_{\pi(n_r-1)j} + \delta_{\pi(n_r)j1} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F \qquad (4.7)$$

$$\mu_{\pi(i)jk} = \mu_{\pi(i)j,k-1} + \delta_{\pi(i)jk} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0,1\} \qquad (4.8)$$

$$\mu_{\pi(i)jk} \geq a_{jk-1} \theta_{\pi(i)jk} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0,1\} \qquad (4.9)$$

$$\mu_{\pi(i)jk} \leq a_{jk} \theta_{\pi(i)jk} + (1 - \theta_{\pi(i)jk}) Q \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.10)$$

$$\sum_{j\in F} \varepsilon_{\pi(i)j} \leq 1, \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\} \qquad (4.11)$$

$$\theta_{\pi(i)jk} \leq \varepsilon_{\pi(i)j} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.12)$$

$$\delta_{\pi(i)jk} \leq \theta_{\pi(i)jk} Q \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.13)$$

$$\bar{t} + \sum_{\pi(i)\in\Pi\setminus\{\pi(0)\}} \sum_{j\in F} \sum_{k\in B\setminus\{0\}} \frac{\delta_{\pi(i)jk}}{\rho_{jk}} +$$

$$\sum_{\pi(i)\in\Pi} \sum_{j\in F} \varepsilon_{\pi(i)j} (t_{\pi(i-1)j} + t_{j\pi(i)} - t_{\pi(i-1)\pi(i)}) \leq T_{max} \qquad (4.14)$$

$$\phi_{\pi(i)} \geq 0, \qquad \forall \pi_i \in \Pi \setminus \{\pi(0)\} \qquad (4.15)$$

$$\varepsilon_{\pi(i)j} \in \{0,1\}, \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F \qquad (4.16)$$

$$\theta_{\pi(i)jk} \in \{0,1\} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.17)$$

$$\delta_{\pi(i)jk} \geq 0 \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.18)$$

$$\mu_{\pi(i)jk} \geq 0 \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \qquad (4.19)$$

The objective function (4.1) seeks to minimize the total route time (including charging and detour times). Constraints (4.2-4.5) define the battery level when the EV arrives at vertex $\pi(i) \in \Pi$ if $\varepsilon_{\pi(i)j} = 0$; or to CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$, if $\varepsilon_{\pi(i)j} = 1$. Constraints (4.6-4.8) define the battery level when the EV finishes charging at CS $j \in F$ in the segment between breakpoints $k-1$ and $k \in B$ before visiting vertex $\pi(i) \in \Pi$. Constraints (4.9-4.10) ensure that if the EV charges on a given segment, the battery level lays between the values of its corresponding break points ($a_{j,k-1}$ and a_{jk}). Constraints (4.11) state that only one CS is visited between any two vertices of the fixed route. Constraints (4.12) ensure that the EV only uses segments of visited CSs. Likewise, constraints (4.13) ensure that the EV charges only at selected segments of visited CSs. Constraint (4.14) represents the duration constraint of the route. Finally, constraints (4.15-4.19) define the domain of the decision variables.

4.3.2 Solving the FRVCP

To solve the FRVCP, we embedded into our method two different solution approaches. The first consists in solving the MILP introduced in the previous subsection using a commercial solver. The second consist in solving the problem using a greedy heuristic adapted from the literature. The remainder of this subsection describes these approaches.

Approach 1: commercial solver

Because of its verbosity, at first glance the model introduced in Section 4.3.1 may seem too complex to be efficiently solved using out-of-the-box software. Nonetheless, since in practice the number customers per route and the number of available CSs tend to be low, the resulting MILP formulations are within the scope of commercial solvers. For instance, on an problem with 8 CSs, the MILP formulation for a fixed route serving 10 customers has: 539 continuous variables, 352 integer variables, and 1,263 constraints[2]. We optimally solved that model using Gurobi Optimizer (version 5.6.0) in 0.06s. Based on this observation we decided to embed into our ILS+HC a component that relies on a commercial solver to tackle the FRVCP.

To further reduce the size of the MILP formulation and consequently improve solver's performance, we propose four preprocessing strategies that eliminate infeasible CSs insertions. Our strategies rely on the two following premises: (i) the energy consumption and the travel time between vertices satisfy the triangular inequality; and (ii) since the piecewise linear charging function is concave (i.e., $\rho_{j,k-1} \geq \rho_{jk}$), the first segment has the fastest charging rate.

We propose two types of strategies. The first three strategies filter CS insertions that are infeasible independently of how the customers are packed and sequenced in the routes. These strategies are applied only once before running the ILS+HC. On the other hand, the fourth strategy filters CS insertions that are infeasible for a particular fixed route.

Strategy 1 : This strategy estimates the minimum time τ needed to visit CS $j \in F$ between two vertices i and $h \in I \cup \{0\}$. This time is defined as the sum of a lower bound on the travel time (u) and a lower bound on the charging time (v) of any route serving customers i and h. Note that in any route serving customer i, t_{0i} is a lower bound on the route's duration from the start up to customer i. Similarly, note that in any route servicing customer h, t_{h0} is a lower bound on the time needed to complete a route from vertex h. Based on these two observations we can compute the minimum duration of a route visiting CS j between vertices i and h as $u = t_{0i} + t_{ij} + t_{jh} + t_{h0} + p_i + p_h$. To compute the minimum charging time v, we need to compute the minimum amount of energy (ec) that an EV coming from vertex i and traveling to vertex h must charge at CS j. This amount is the charge needed to recover the energy consumed to make the detour to j, that is, $ec = e_{ij} + e_{jh} - e_{ih}$. Because the battery level when the EV arrives at i in any eVRP-NL solution is unknown a priori, we consider that the battery is charged at j using the fastest charging rate (ρ_{0j}). Then $v = \frac{ec}{\rho_{0j}}$. It is clear that if $\tau = u + v > T_{max}$ any route visiting CS j between customers i and h is infeasible. We therefore forbid this insertion in our MILP.

Strategy 2 : This strategy computes a lower bound on the remaining energy that an EV must have at arrival to customer i to be able to visit CS j right after. Note that in terms of energy remaining, the best way to reach vertex i is to visit it right after fully charging at CS $c(i) = \arg\min_{l \in F \cup \{0\}} e_{li}$. If $Q - e_{c(i)i} < e_{ij}$ any route visiting j after i is energy-infeasible and we can safely forbid this insertion in our MILP.

Strategy 3 : Note that $o = Q - e_{ji}$ is a lower bound on the energy remaining when an EV arrives at customer i right after charging at CS j. Note also that o must be enough to at least close the route (reach the depot) or reach the closest CS in terms of energy consumption. If $o < e_{ic(i)}$, where $c(i) = \arg\min_{l \in F \cup \{0\}} e_{il}$, any route visiting j before i is energy-infeasible and we can safely forbid this insertion.

[2]We randomly picked the route from the best solution found to a randomly picked instance

Strategy 4 : This strategy estimates a lower bound on the new duration of a given fixed route if CS $j \in F$ is inserted between vertices $\pi(i)$ and $\pi(i+1) \in \Pi$, $i \neq n_r$. This bound, $\overline{t'}$, is defined as the sum of the new travel time (u) and a lower bound on the charging time (v). It is easy to see that $u = \overline{t} + t_{\pi(i)j} + t_{j\pi(i+1)} - t_{\pi(i)\pi(i+1)}$. Similarly to Strategy 1, to estimate v, we consider that the battery is charged at j using the fastest charging rate. Therefore $v = \frac{ec}{\rho_{0j}}$, where $ec = e_{\pi(i)j} + e_{j\pi(i+1)} - e_{\pi(i)\pi(i+1)}$ is the charge needed to recover the energy consumed in the detour to j. If $\overline{t'} = (u + v) > T_{max}$, inserting j between vertices $\pi(i)$ and $\pi(i+1)$ leads to an infeasible route, so we can safely forbid this insertion.

Applying our preprocessing strategies to the MILP formulation for the 10-customer route of the example above, the model reduces to: 71 continuous variables, 40 integer variables, and 165 constraints. We solved the model in Gurobi in 0.02s.

Approach 2: greedy heuristic

Existing metaheuristics for eVRPs use different approaches to make charging decisions. One popular approach is the *recharge relocation operator* proposed by Felipe et al. (2014) for the green vehicle routing problem with multiple technologies and partial recharges (GVRP- MTPR). This approach considers the insertion of only one CS per route. Starting from an energy-feasible fixed-route the procedure first deletes the current CS. Then, it tries to improve the charging decisions by inserting each CS into each arc of the fixed-route. To decide how much energy to charge at the inserted CS, the algorithm applies a simple rule: charge the minimum amount of energy needed to reach the depot (i.e., to complete the route). We propose here a heuristic to solve the FRVCP based on this approach.

Our heuristic works in two phases: *location of CSs* and *charge setting*. In the first phase, the heuristic iteratively inserts CSs into the arcs of the fixed-route Π in order to ensure the feasibility in terms of energy. In the second phase, the heuristic improves the charging decisions by adjusting the energy charged at each visited CS. Algorithm 3 describes the structure of our heuristic. It uses four important procedures `trackBattery`$(\cdot)$, `sumNegative`$(\cdot)$, `totalTime`$(\cdot)$, and `copyAndInsert`$(\cdot)$. Procedure `trackBattery`$(\cdot)$ computes the battery level Y_i at each vertex $i \in \Pi$, assuming that the EV fully charges its battery at each visited CS. Note that Y_i may take negative values. Procedure `sumNegative`$(\cdot)$ computes the sum of the battery levels with negative values (i.e., $s = \sum_{i \in \Pi} min\{0, Y_i\}$). Procedure `totalTime`$(\cdot)$ computes the total time t of the route (supposing a full charging policy). Finally, procedure `copyAndInsert`$(\cdot)$ takes as input a fixed route, a CS, and a position in the route; and returns a copy of the fixed route with the CS inserted at the given position.

The heuristic starts by the location phase (line 2-27). After computing s for the current fixed-route Π, the heuristic enters the outer loop (line 7-27). In each pass through the inner loop (line 8-25), the heuristic: i) evaluates the insertion of a CS into each arc of Π assuming that the EV fully charges its battery, and ii) selects the insertion that maximizes s (lines 14-18). If $s = 0$ (i.e., the route is energy-feasible), the heuristic selects the insertion that minimizes t (line 19-23). Then, the heuristic performs the selected insertion (line 26). If the route Π is still energy-infeasible (i.e., $s < 0$), the heuristic starts again at line 8 and tries to insert additional CSs until feasibility in terms of energy is reached.

In the charge setting phase (line 28), the heuristic invokes procedure `ruleMinEnergy`(Π) to set the energy charged at each CS following the Felipe et al. (2014) rule. Finally, the heuristic evaluates if the route satisfies the maximum-duration constraint (29-33). The heuristic returns a boolean variable indicating whether or not the fixed-route is feasible (f), and the route Π with the newly inserted CSs.

4.4 Computational experiments

In this section, we present three computational studies. The first study compares the quality of the solutions obtained by two versions of our metaheuristic. The second study evaluates the CPU time of our

Algorithm 3 Greedy heuristic

```
1: function GREEDYHEURISTIC(Π^0,F)
2:     Π ⟵ Π^0
3:     Y ⟵ trackBattery(Π)
4:     s ⟵ sumNegative(Y)
5:     t ⟵ ∞
6:     f ⟵ false
7:     while s < 0 do
8:         for j = 1 to |F| do
9:             for i = 0 to n_r − 1 do
10:                Π' ⟵ copyAndInsert(Π, F_j, i)
11:                Y' ⟵ trackBattery(Π')
12:                s' ⟵ sumNegative(Y')
13:                t' ⟵ totalTime(Π')
14:                if s' > s then
15:                    s ⟵ s'
16:                    u ⟵ j
17:                    v ⟵ i
18:                end if
19:                if s' = 0 and t' < t then
20:                    t ⟵ t'
21:                    u ⟵ j
22:                    v ⟵ i
23:                end if
24:            end for
25:        end for
26:        Π ⟵ copyAndInsert(Π, F_u, i)
27:    end while
28:    ⟨t, Π⟩ ⟵ RuleMinEnergy(Π)
29:    if t ≤ T_max then
30:        f ⟵ true
31:    else
32:        Π ⟵ Π^0
33:    end if
34:    return f, Π
35: end function
```

metaheuristic, and assesses the impact of the preprocessing strategies. Finally, the third study analyses the charging decisions of the best solutions found.

4.4.1 Test instances for the eVRP-NL

To test our approach, we generated a new 120-instance testbed built using real data of EV configuration and battery charging functions. In order to ensure feasibility, we opted to generate our instances instead of adapting a existing dataset from the literature. To build the instances we first generated 30 sets of customer locations with {10, 20, 40, 80, 160, 320} customers. For each instance size, we generated 5 sets of customers location. We located the customers in a geographic space of 120 x 120 km using either a random uniform distribution, a random clustered distribution, or a mixture of both. For each of the 30 sets of locations we chose the customer location strategy using a uniform probability distribution. Our main motivation to choose a 120 x 120 geographic area was to build instances representing a semi-urban operation. These operations are the best suited applications for eVRPs. Indeed, in city operations routes tend to be sufficiently short to be covered without mid-route charging. On the opposite side, in rural operations routes tend to be long enough to required multiple mid-route charges but access to charging infrastructure is very limited (at least for 2016 standards).

From each of the 30 sets of locations we built 4 instances varying the level of charging infrastructure availability and the strategy used to locate the CSs. We considered two levels of charging infrastructure availability: low and high. To favor feasibility, for each combination of number of customers and infrastructure availability level we handpicked the number CSs as a proportion of the number of customers. We located the CSs either randomly or using a simple p-median heuristic. Our p-median heuristic starts from a set of randomly generated CS locations and iteratively moves those locations trying to minimize the total distance between the CSs and the customers. In our instances, we included three types of CSs: slow, moderate, and fast. For each CS we randomly selected the type using a uniform probability distribution.

The EVs in our instances are Peugeot Ion. This EV has a consumption rate of 0.125 kWh/km, and a battery of 16 kWh. Note that an EV with this characteristics is well suited to service applications such as homecare routing. In reality the exact energy consumption on an arc (e_{ij}) varies with parameters such as the cumulative elevation gain, the external temperature, the speed, and the use of peripherals (e.g., the radio). Nonetheless, for the sake of simplicity we followed the classical approach in the literature and assumed that the energy consumption on an arc is simply the EV's consumption rate multiplied by the arc's distance. To generate the charging functions we fit piece-wise linear functions to the real charging data for a 16 kWh battery provided by Uhrig et al. (2015). Figure 4.3 depicts our piece-wise linear approximations. Finally, the maximum route duration for every instance was fixed to 10 hours. Our 120 instances are publicly available at `www.vrp-rep.org` (Mendoza et al. 2014)[3].

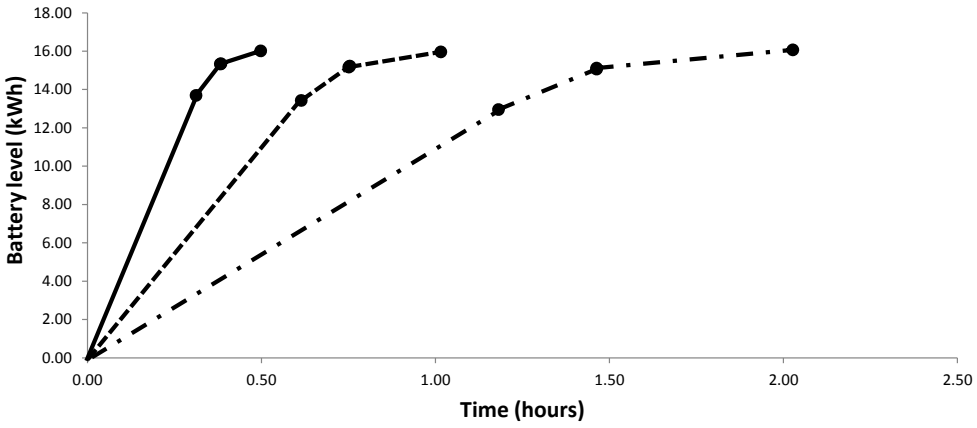
4.4.2 Parameter settings & experimental environment

As discussed in Section 4.3.2 our ILS+HC can be configured to solve the FRVCP using two different approaches, namely, the commercial solver and the greedy heuristic. We tested our algorithm with two different configurations. The first configuration (ILS(S)+HC) uses the commercial optimizer to solve the FRVCP in the GCI neighborhood, while the second configuration (ILS(H)+HC) uses the greedy heuristic. In both configurations, the algorithm solves the FRVCP in the split procedure using the greedy heuristic. This choice was guided by computational performance.

To fine tune the number of iterations of the ILS component (i.e., K), we conducted a short computational study. Our results showed that $K = 80$ provides the best trade-off between solution quality and computational performance.

We implemented our ILS in Java (jre V.1.8.0) and used Gurobi Optimizer (version 5.6.0) to solve the FRVCP and the set partitioning problem in the HC component. We set a time limit of 800 seconds on

[3]The instances will be made effectively public after the completion of the reviewing process



Type of CSs	Slow	Moderate	Fast
Charging power (kWh/h)	11	22	44
Piecewise linear approximation	— · —	------	———

Figure 4.3 – Piecewise linear approximation for different types of CS charging an EV with a battery of 16 kWh.

Table 4.1 – Comparison of the two versions of the metaheuristic on small instances with proven optima

Metric	**ILS(H)+HC**	**ILS(S)+HC**
Number of optimal solutions	15/20	20/20
Avg. Gap (%)	1.97	0.34
Max. Gap (%)	17.07	1.87
Avg. Best Gap (%)	1.20	0.00

Gurobi to control the running time of the HC phase. All the experiments were run on a computing cluster with 2.33 GHz Inter Xeon E5410 processors with 16 GB of RAM running under Linux Rocks 6.1.1. The results delivered by the two ILS+HC configurations are compared over 10 runs. Each replication of the experiments was run on a single processor.

4.4.3 Solution accuracy: optimal vs. heuristic charging decisions

Since the eVRP-NL is a new problem, there are no results or algorithms to benchmark against. To get an idea of the quality of the solutions delivered by our ILS+HC, we ran its two versions, ILS(S)+HC and ILS(H)+HC, on the twenty 10-customer instances and compared their results with optimal solutions found using the MILP presented in Chapter 3. Table 4.1 summarizes the results comparing the two algorithms under the light of four metrics: the number of optimal solutions found, the average and maximum gap with respect to the optimal solution over the 10 runs, and the average best gap[4]. Appendix A.2 presents detailed results for the 20 instances.

The results suggest that our algorithms are able to deliver high-quality solutions for the eVRP-NL. As shown in the table, ILS(S)+HC matched the 20 optimal solutions while ILS(H)+HC matched 15. The spread in the number of optimal solutions and the values reported for Avg. Best Gap tip the balance towards ILS(S)+HC over ILS(H)+HC in terms accuracy. This result is not surprising since one could expect that making optimal, instead of heuristic, charging decisions would translate into higher quality solutions. A close look at the Avg. and Max. gaps reveals a less foreseeable result: ILS(S)+HC exhibits a significantly

[4]The best gap is the gap between the best solution found over 10 runs and the optimal solution

Table 4.2 – Comparison of the two versions of the metaheuristic on large instances

Metric	Gurobi	ILS(H)+HC	ILS(S)+HC
Number of solutions	25/100	100/100	100/100
Number of BKSs	7/100	7/100	100/100
Avg. Gap (%)	41.36	7.51	1.51
Max. Gap (%)	NA	28.68	4.44
Avg. Best Gap (%)	NA	5.28	0.00

more stable behavior than ILS(H)+HC.

To compare the performance of ILS(S)+HC and ILS(H)+HC on more industrial-sized instances we ran both algorithms on the remaining 100 benchmarks. Table 4.2 summarizes the results. In the comparison we employed the same metrics introduced above, replacing the number of optimal solutions by the number of best known solutions (BKSs) found. As a reference, we included in the table the results delivered by the commercial solver running the MILP for 10h. It is worth mentioning that Gurobi reported integer solutions for only 25 out of the 100 instances. None of these 25 solutions has a certificate of optimality. Detailed results for each instance can be found in Appendix A.2

The results in Table 4.2 confirm the conclusions drawn on our first experiment: optimally solving the FRVCP in the ICG neighborhood leads to a more accurate and stable solution method. According to our data, ILS(S)+HC not only found the 100 BKSs but also reported significantly lower Avg. and Max. gaps (1.51% and 4.44% vs. 7.51% and 28.68%).

The cost of making optimal charging decisions

In Section 4.4.3 we showed the benefits of making optimal charging decisions when solving the eVRP-NL. This benefits, however, comes at the price of losing computational performance. To estimate this loss, we measured the execution times of three versions of our algorithm on the 120 instances: ILS(S)+HC, ILS(H)+HC, and ILS(S')+HC. The latter is a version of ILS(S)+HC running without our preprocessing strategies (see Section4.3.2). Table 4.3 reports for each instance size, the average execution time (in seconds) of each algorithm. In addition, the last column of the table reports the average speedup between ILS(S′)+HC and ILS(S)+HC (measured as the ratio of their execution times).

Not surprisingly, the results show that ILS(H)+HC is the fastest approach. It is interesting to note that in relative terms, the gap in computational performance between ILS(H)+HC and ILS(S)+HC decreases with the size of the instance. This behavior can be explained by the positive correlation between the instance size and the time needed to solve the set partitioning model in the HC phase. Indeed, the latter heavily depends on the size of the pool Ω which, in turn, is more correlated to the size of the instance than to the strategy used to generate the routes. In other words, on large instances both algorithms spend around the same (considerable) amount of time solving the HC phase. As a consequence, the performance gap in the ILS phase becomes less remarkable.

A second interesting conclusion from this study is that the preprocessing strategies have a remarkable positive impact on the computational performance of the algorithm. As the table shows, on average, the strategies are responsible for a 1.65 speedup on the execution time. The data also suggests that the speedup is independent of the size of the instance. A plausible explanation for this behavior is that for a given instance both ILS(S)+HC and ILS(S')+HC evaluate around the same number of routes during the ILS phase.

Characteristics of good eVRP-NL solutions

We analyze in this section the characteristics of the BKSs found in our experiments. We aim to provide the reader with some insight that may be useful when designing new solution methods for the eVRP-NL.

Table 4.3 – Average computing time (in seconds) of different variants of the metaheuristic

Instance size	ILS(H)+HC	ILS(S′)+HC	ILS(S)+HC	Speedup
10	0.64	8.54	5.62	1.52
20	1.75	17.47	10.56	1.65
40	8.48	64.16	35.35	1.82
80	39.35	148.76	80.11	1.86
160	289.08	976.84	568.02	1.72
320	2,568.94	5,759.67	4,397.64	1.31
Average	484.71	1,162.57	849.55	1.65
Max	4,766.36	10,335.56	7,636.50	2.53
Min	0.49	3.71	2.36	1.16

ILS(S)+HC and ILS(S′)+HC are the hybrid metaheuristic with and without preprocessing strategies, respectively

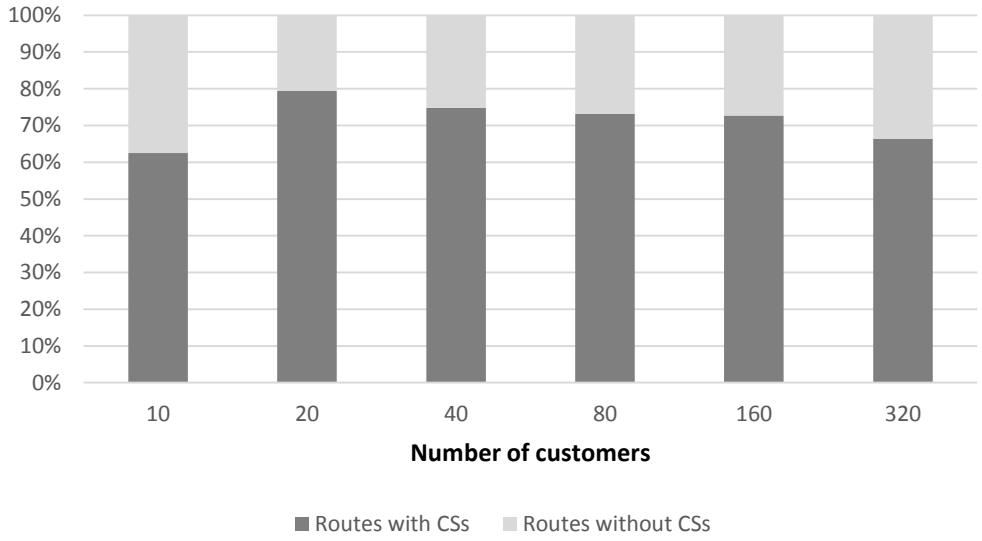
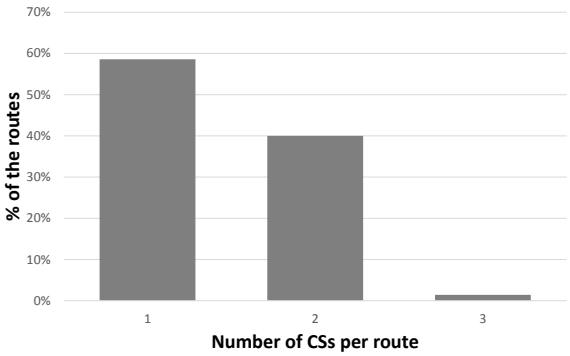


Figure 4.4 – Percentage of the routes with/without visits to CSs by instance size.

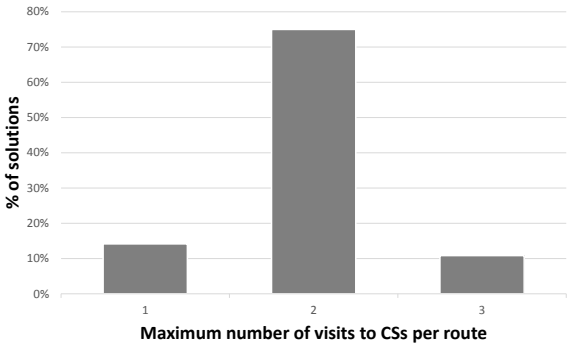
In total, our BKSs are made up of 1,426 routes. Our first analysis concerns the fraction of those routes that exploit mid-route charging. Figure 4.4 presents the percentage of routes with and without visits to CSs grouped by instance size. The data shows that on average 71.47% of the routes in the BKSs visit at least one CS. This percentage is roughly the same for each instance size. This figure provides two insights. First, mid-route charging is a key element of good eVRP-NL solutions (probably because it gives algorithms flexibility to better pack the customers into the routes). Second, since charging decisions concern most of the routes making up a good solution, they play a critical role in its quality. The latter observation helps explaining the spread on accuracy between ILS(S)+HC and ILS(H)+HC found in the computational studies reported in Section 4.4.3.

The second analysis concerns the number of mid-route charges per route. Figure 4.5a and 4.5b present histograms of the number mid-route charges per route and the maximum number of mid-route charges per route on a solution. Figure 4.5a shows that among the routes performing mid-route charging, 58.58% do it once, 40.00% twice, and 1.43% three times. Although a large portion of the routes perform a single mid-route charge, 85.83% of the solutions contain at least one route performing more than one mid-route charge (Figure 4.5b). This figures suggest that models and methods for the eVRP-NL can benefit from relaxing the at-the-most-one-visit-to-a-CS-per-route constraint that is sometimes used in the eVRP literature.

The third analysis concerns the energy recovered through mid-route charges. Figure 4.6 presents the



(a) Histogram of number of visits to CS per route.



(b) Histogram of the maximum number of visited CSs in the routes of each solution.

Figure 4.5 – Analysis of the number of visits to CSs

histogram of the average battery level (in % of the total battery capacity) after a mid-route charge. The numbers show that over 90% of the mid-route charges are partial charges (i.e., they do not fully charge the battery). This figure tips the importance of embedding components capable of making partial charging decisions into eVRP-NL solution methods[5]. A second interesting observation from the data displayed in Figure 4.6 comes from the percentage (around 12%) of mid-route charges that restore the battery above 80% of its capacity. As mentioned in the study reported in Chapter 3, one common assumption in the eVRP literature is that the battery can only be charge on the linear segment of the charging curve (which ends at roughly 80% of the battery capacity). Our data suggest that good eVRP-NL solutions often include routes with mid-route charges that take the battery level up to the non-linear part of the charging function.

4.5 Conclusion

This chapter introduces an extension of the electric vehicle routing problem which considers more-realistic assumptions about the battery charging process: the electric vehicle routing problem with nonlinear charging function (eVRP-NL). To solve the problem we propose an iterated local search (ILS) enhanced with heuristic concentration (HC). At the heart of the proposed method lays a neighborhood scheme consisting in solving a new variant of the fixed-route vehicle-charging problem (FRVCP). This problem consists in optimizing the charging decisions (where and how much to charge) of a route serving a fixed sequence of customers. Depending on the configuration, our ILS+HC solves the underlying FRVCP either using a greedy heuristic or a commercial solver. To improve the performance of the solver, we proposed four pre-processing strategies that eliminate infeasible detours to CSs. To assess the performance of our method, we built a set of 120 instances based on real EV and battery charging data. Tested on 20 small 10-customer instances, our method matched the optimal solutions for every instance. Experiments conducted on larger instances proved the value of equipping eVRP-NL algorithms with components capable of making optimal charging decisions. Finally, we analyzed the solutions delivered by our method aiming to provide fellow researchers with some insight into the characteristics of good eVRP-NL solutions. Our analysis concluded that good eVRP-NL solutions tend to use multiple mid-route charges, exploit partial recharges, and employ the non-linear segment of the battery charging function.

[5]Note that up to 2016 this was rather the exception than the rule in the eVRP literature

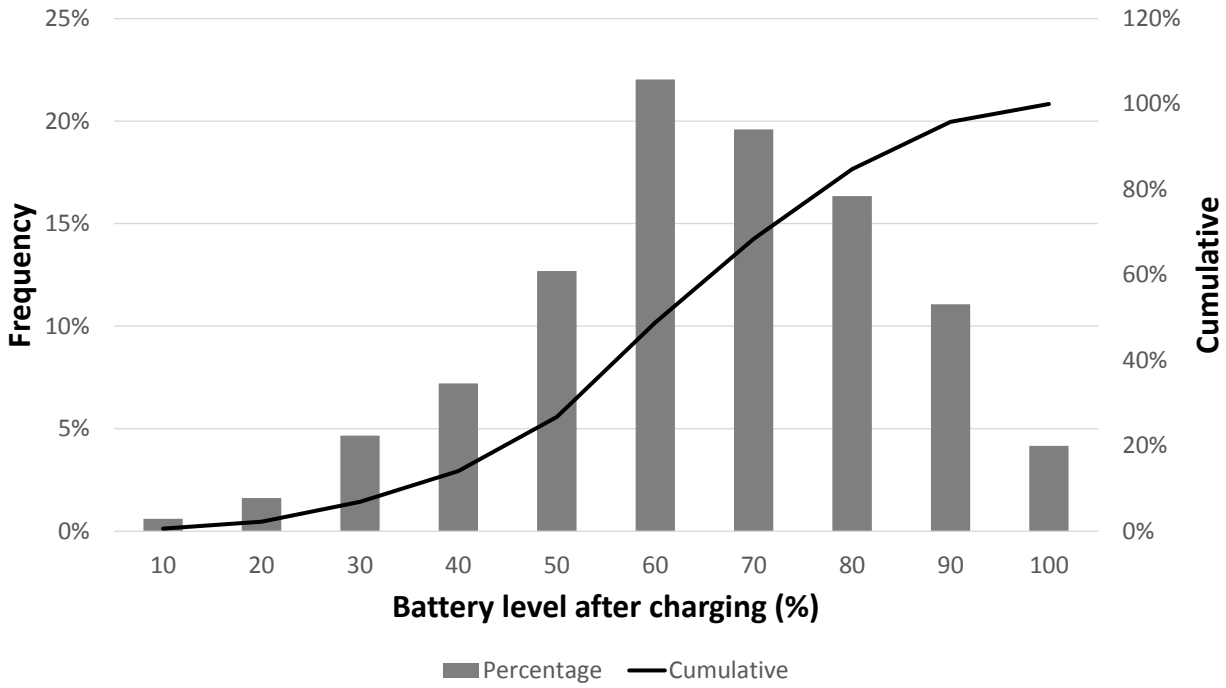


Figure 4.6 – Histogram of the average battery level (in % of the total battery capacity) after a mid-route charge

4.6 Nomenclature

4.6.1 Notation for problem description

G: eVRP underlying graph $G = (V, A)$.
V: Set of vertices of G.
0: Depot
I: Subset of V representing customers.
F: Subset of V representing CSs.
A: Set of arcs in G.
B: Set of breakpoints of the piecewise linear charging function.
p_i: Service time at customer $i \in I$.
c_{ik}: Charging time of the CS $i \in F$ at the breakpoint $k \in B$.
a_{ik}: Charge level of the CS $i \in F$ at the breakpoint $k \in B$.
e_{ij}: Energy consumption between vertices i and j ($i, j \in V$).
t_{ij}: Travel time between vertices i and j ($i, j \in V$).
T_{max}: Tour duration constraint.
Q: Tank capacity.

4.6.2 Notation for hybrid metaheuristic

G^*: Acyclic graph for the split procedure $G^* = (V^*, A^*)$.
V^*: Set of vertices of G^*.
A^*: Set of arcs in G^*.
Ω: Pool of feasible eVRP-NL routes.

4.6.3 Notation for the fixed-route vehicle-charging problem

Π: Energy-infeasible route.
$\bar{t}$: Total time of the route Π, which is the sum of the travel times plus the service times.
ρ_{jk}: Slope of the segment defined between the breakpoints $k-1$ and $k \in B$.
$\varepsilon_{\pi(i)j}$: Binary variable, equal to 1 if the EV charges at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$.
$\phi_{\pi(i)}$: Battery level. If $\varepsilon_{\pi(i)j} = 0$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at vertex $\pi(i)$. On the other hand, if $\varepsilon_{\pi(i)j} = 1$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at CS $j \in F$ right before visiting vertex $\pi(i)$.
$\theta_{\pi(i)jk}$: Binary variable, equal to 1 if the EV charges on the segment defined by breakpoints $k-1$ and $k \in B$ at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$.
$\delta_{\pi(i)jk}$: Amount of energy charged when the charging finishes on the segment between breakpoints $k-1$ and $k \in B$ at CS $j \in F$ before the visit to vertex $\pi(i) \in \Pi$.
$\mu_{\pi(i)jk}$: Battery level when the charging finishes on the segment between breakpoints $k-1$ and $k \in B$ at CS $j \in F$ before the visit to vertex $\pi(i) \in \Pi$.
τ: Minimum time needed to visit CS $j \in F$ between two vertices i and $h \in I \cup \{0\}$.
u: Lower bound on the travel time.
v: Lower bound on the charging time.
ec: Charge needed to recover the energy consumed to make the detour to $j \in F$.
o: Lower bound on the energy remaining when an EV arrives at customer $i \in I$ right after charging at CS $j \in F$.
$c(i)$: Closest station to the customer $i \in I$ (including the depot).
$\overline{t'}$: Lower bound on the new duration of a given fixed route if CS $j \in F$ is inserted between vertices $\pi(i)$ and $\pi(i+1) \in \Pi$, $i \neq n_r$.
Y_i: Battery level at each vertex $i \in \Pi$, assuming that the EV fully charges its battery at each visited CS.
s: Sum of the battery levels with negative values.
t: Total time of the route (supposing a full charging policy).

Chapter 5

The technician routing problem with conventional and electric vehicles

The work of this chapter was presented at the following conferences and seminar:

- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2015). The technician routing problem with conventional and electric vehicles. In VeRoLog 2016, Nantes, France.
- Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). Electric vehicle routing problems with non-linear charging functions. Cirrelt seminar, Université de Montréal Campus, Montreal, Canada, October 18, 2016

The research of this chapter benefited from the support of the “FMJH Program Gaspard Monge in optimization and operations research", and from the support to this program from EDF.

5.1 Introduction

Nowadays, service and utility companies have started to use electric vehicles (EVs) on their operations. Considering that EVs are a new technology, and that the companies currently have a fleet of conventional vehicles (CVs), the incorporation of EVs into their fleets is made gradually. For instance, the french electricity distribution company ENEDIS has started to incorporate 2,000 EVs in its fleet of 20,000 vehicles (ERDF 2014). Similarly, the spanish electricity company Endesa has started to introduce EVs in its fleet; they expected to have 3,000 EVs by 2020 (H&E 2016). The incorporation of EVs into existing fleets of CVs gives birth to what is commonly known in the literature as vehicle routing problems with electric and conventional vehicles (Goeke & Schneider 2015).

In this chapter, we introduce what we call the technician routing problem with conventional and electric vehicles (TRP-CEV). The TRP-CEV is largely inspired by a problem faced by ENEDIS in their daily operations. The problem consists in routing a set of technicians to serve a set of geographically scattered requests within their time windows. The problem also considers the working schedules, the lunch breaks, and skills of the technicians. The technicians use a fixed fleet composed of CVs and EVs of different types. Each type of EV has its own autonomy charging protocol. The latter determines in which charging stations (CSs) the EV can charge its battery. We assume that the charging functions are nonlinear as proposed in Chapter 3. In the TRP-CEV decisions include the vehicle-to-technician assignment, the sequencing of the routes followed by the technicians, and the battery charging program for the EVs (where and how much to charge). The objective function seeks to minimize the total cost, defined as the sum of the travel costs, battery charging fixed costs, the parking cost at CSs, and the fixed cost of each technician. Following the definition of (Lahyani et al. 2015) the TRP-CEV lays in the category of rich vehicle routing problem (RVRP) [1].

It is widely accepted that large-scale instances of complex RVRPs are out of the reach of exact methods. Moreover, in industrial contexts in which the problem must be solved on a daily (or even weekly) basis, companies tend to prefer the more-balanced trade off between solution quality and speed offered by metaheuristics. Since one of our primary objectives is to provide ENEDIS with an optimization algorithm that can eventually be deployed to production, we leaned towards designing a parallel matheuristic (PMa) for their problem. The proposed approach works in two phases. In the first phase, the algorithm decomposes the TRP-CEV into a set of "easier-to-solve" vehicle routing problems with time windows and lunch breaks (VRPs-TWLB) and solves these problems in parallel using a greedy randomized adaptive search procedure (GRASP). In the second phase, the algorithm assembles a solution to the TRP-CEV by solving a set-covering formulation over a set of high-quality routes found for the independent VRP-TWLB.

The main contribution of this chapter are fourfold. First, we introduce the TRP-CEV. Second, we propose a parallel matheuristic to solve the TRP-CEV which can be extended to other problems with competitive results. Third, we develop a set of real instances based on the ENEDIS operation. Fourth, we study the impact of different fleet compositions on relevant variables for ENEDIS.

The remainder of this chapter is organized as follows. Section 5.2 reviews the related literature. Section 5.3 describes the problem. Section 5.4 introduces our parallel matheuristic. Section 5.5 introduces the fixed-route vehicle-charging problem with time windows, which arises as a subproblem of TRP-CEV. Section 5.6 presents the results of a computational experiment with ENEDIS real instances. Finally, Section 5.7 concludes the chapter. For easy reference, Section 5.8 summarizes the notation of this chapter.

5.2 Literature review

The TRP-CEV combines features from different problems in the literature: the technician routing problem (TRP), the vehicle routing problem with lunch breaks (VRP-LB) and the heterogeneous vehicle routing

[1] According to (Lahyani et al. 2015), if a VRP has at least nine physical characteristics, this problem can be classified as a RVRP

problem (HVRP). In the remainder of this chapter, we present a brief review of literature on these problems.

One of the first problems in the field of TRPs is the field technician scheduling problem (FTSP) by (Xu & Chiu 2001). The FTSP consists in routing a set of technicians to serve a set of geographically scattered requests. Each request has a time window, and a priority (represented by a weight). Each technician has a working schedule, and a skill proficiency level for each request. The objective functions for the FTSP is to maximize the number of served requests weighted their priority and minimize the total working time. Dohn et al. (2009) proposed an interesting variant of the TRP, where some requests require collaboration between technicians, and therefore the technicians must initiate the execution simultaneously. Pillac et al. (2013) studied a version of the TRP where the technician-task compatibility includes spare parts and tools. The technicians have the opportunity to replenish their tools and spare parts at a central depot at any time. Kovacs et al. (2012) consider that technicians have different proficiency levels in several skills, and the option that technicians can be grouped into teams to serve specifics request. Recently, Zamorano & Stolletz (2016) extend the TRP considering multiple periods. In their problem, technicians are assigned every day to a team, and the composition of the team remains fixed for the duration day. For a detailed review of the TRP and other related problems the reader is referred to (Castillo-Salazar et al. 2016). As can be seen the TRPs have many variations. In the case of the TRP-CEV, it only consider the technician-task compatibility that depends of the required skills of the requests and the skills of the technicians.

VRPs with lunch breaks arise mainly in service industries. For instance, in home healthcare operations where nurses and doctors visit a set of customers at home (Liu et al. 2016, Cheng & Rich 1998). In the reparation or maintenance service, where the technicians serve a set of requests (Kovacs et al. 2012, Bostel et al. 2008). To the best of our knowledge, all studies tacking VRP-LB consider a time window to start the lunch break and a specified duration. Futhermore, most of the studies assume that the lunch break is taken at a customer location (Liu et al. 2016, Cheng & Rich 1998, Kovacs et al. 2012). The only exception is the study by Bostel et al. (2008), where the authors assume that the lunch break is taken at specified locations (restaurants). In contrast with the previous studies, for the TRP-CEV the lunch break is defined between two time points in the working time, where the technician must stops any activity to take the lunch break.

In the last 30 years, the HVRPs have been one the most studied problem in the VRP field (Koç et al. 2016, Baldacci et al. 2008). However, only recently researchers have started to study the HVRP with EVs. Goeke & Schneider (2015) introduced the electric vehicle routing problem with time windows and mixed fleet (E-VRPTWMF), which considers a fleet of CVs and EVs. Sassi et al. (2015) extended the E-VRPTWMF considering different features motivated by a real-life industrial application. In their problem, the CSs have different technologies and time windows, the charging cost depends on the time of the day, and EVs can charge partially their batteries. Finally, Hiermann et al. (2016) introduced the electric fleet size and mix vehicle routing problem with time windows and recharging stations (E-FSMFTW), where the fleet is composed of different types of EVs. The objective of the E-FSMFTW is to select the size and composition of the EV fleet. Although, other previous studies have assumed mixed fleet with EVs and CVs, the TRP-CEV is the only to assume the nonlinear charging behavior of the charging process.

Despite in the literature there are several studies about the problems that compose the TRP-CEV, as far as we know no other authors have studied a problem that combines all those components. Moreover, the TRP-CEV is the only problem that introduces EVs in the technician operation context, taking into account the existing real applications of EVs in operation with technician (ERDF 2014, H&E 2016).

5.3 Problem description

Formally, the TRP-CEV is defined on a directed and complete graph $G = (N, A)$. The vertex set $N = \{0\} \cup C \cup F$ is made up of a depot (vertex 0), a set of requests C, and a set of CSs F. Each request $i \in C$ has a service time p_i and a time window $[ec_i, lc_i]$, and requires a technician with specific skills from the set of skills S. Each CS $i \in F$ has a parking cost pc_i. The request are served by a set of T technicians. Each technician $u \in T$ has the following characteristics: a fixed cost f_u; a subset of skills $s_u \subseteq S$; a

working schedule between st_u and et_u, that includes a lunch break between sl_u and el_u; and a consumption factor fc_u associated to his or her driving profile (e.g., sport, regular and eco). The technicians drive a fixed fleet of heterogeneous vehicles composed by different types of CVs and EVs. The set of vehicles types K is made up of the subset of types of CVs (K_c) and the subset of types of EVs (K_e). There are m_k vehicles of type $k \in K$ available in the fleet. Each type of vehicle $k \in K$ has an unitary travel cost tc_k (expressed in €$/km$). Additionally, EVs of type $k \in K_e$ have a fixed cost g_k for recharging the battery (expressed in € and representing the degradation cost), and a battery capacity Q_k (expressed in kWh). Due to technological incompatibilities, each type of EV $k \in K_e$ can charge only in a subset F_k of the CS. The charging function of EVs is modelled using a discrete function. Parameter a_{ikl} represents the charging time of a vehicle of type $k \in K_e$ charging from zero to l percent of the battery at CS $i \in F$ (with $l \in B = 1, 2, ..., 100$). Finally, the set $A = \{(i, j) : i, j \in N, i \neq j\}$ corresponds to arcs connecting vertices in N. Each arc (i, j) has three associated nonnegative values: a travel time t_{ij}, a distance d_{ij}, and an energy consumption e_{ijk} for each type of EV $k \in K_e$.

In the TRP-CEV the objective is to find a set of routes of minimum total cost, defined as the sum of the total travel cost, fixed charging cost, parking cost, and the fixed cost of each technician, satisfying the following conditions: each request is served exactly once within its time window by a technician with the required skills; the level of the battery when the EVs arrive at any vertex is nonnegative; the EVs only charge at CSs with compatible technology; each technician works only during his or her working schedule; each technician takes a lunch break; and each route starts and ends at the depot.

5.3.1 Mixed-integer linear programming formulation

To allow several visits to the CSs, we introduce the set F' that contains the set F and copies of each CS, and the set $N' = \{0\} \cup C \cup F'$. Furthermore, since we need to track the information of each vehicle, we introduce the set of vehicles $V = V_c \cup V_e$ that contains all the CVs (V_c) and EVs (V_e) available (i.e., $|V_c| = \sum_{k \in K_c} m_k$ and $|V_e| = \sum_{k \in K_e} m_k$).

Furthermore, we introduce the following binary parameters: n_{il} is equal to 1 if request $i \in C$ needs a technician with skill $l \in S$, and 0 otherwise. Likewise, m_{ul} is equal to 1 if technician $u \in T$ has skill $l \in S$, and 0 otherwise. Finally, h_{ik} is equal to 1 if the EV $k \in V_e$ can charge at CS $i \in F'$. In addition, the real parameter S_{max} is the maximum charging time.

To model the TRP-CEV using a mixed-integer linear programming (MILP) formulation, we use the following decision variables: x_{ijku} is equal to 1 if technician $u \in T$ travels from vertex i to $j \in N'$ in vehicle $k \in V$, and 0 otherwise. Variable τ_{iu} tracks the arrival time of the technician $u \in T$ at vertex $i \in N'$. Variable y_{ik} tracks the battery level of EV $k \in V_e$ upon departure from vertex $i \in N'$. Variables q_{ik} and o_{ik} represent the battery level when EV $k \in V_e$ arrives at and departs from CS $i \in F'$. Variables α_{ikl} and β_{ikl} are equal to 1 if the battery level is $l \in B$ when vehicle $k \in V_e$ arrives at and departs from the CS $i \in F'$ respectively. Variable Δ_{ik} represents the time spent by vehicle $k \in V_e$ at CS $i \in F'$. Variable w_{iju} is equal to 1 if technician $u \in T$ takes the lunch break on his way from i to $j \in N'$, and 0 otherwise. Variable z_{iju} is equal to 1 if the technician $u \in T$ takes the lunch break after travelling from vertex i to $j \in N'$ and 0 otherwise. Variable z'_{iju} is equal to 1 if the technician $u \in T$ takes the lunch break before travelling from vertex i to $j \in N'$, and 0 otherwise. The MILP follows:

$$\textbf{Min} \sum_{i \in N'} \sum_{j \in N'} \sum_{k \in V} \sum_{u \in T} x_{ijku} \cdot d_{ij} \cdot tc_k + \sum_{i \in F'} \sum_{k \in V_e} \Delta_{ik} \cdot pc_i + \sum_{i \in N'} \sum_{j \in F'} \sum_{k \in V_e} \sum_{u \in T} x_{ijku} \cdot g_k + \sum_{j \in N'} \sum_{k \in V} \sum_{u \in T} x_{0jku} \cdot f_u \tag{5.1}$$

Subject to:

$$\sum_{i\in N'}\sum_{k\in V}\sum_{u\in T} x_{ijku} = 1, \qquad \forall j \in C \tag{5.2}$$

$$\sum_{i\in N'} x_{ijku} \leq 1, \qquad \forall j \in F', \forall k \in V_e, \forall u \in T \tag{5.3}$$

$$\sum_{k\in V}\sum_{u\in T} x_{ijku} \leq 1, \qquad \forall i, j \in N' \tag{5.4}$$

$$\sum_{j\in N'}\sum_{u\in T} x_{ijku} \leq 1, \qquad \forall i \in N', \forall k \in V \tag{5.5}$$

$$\sum_{j\in N'}\sum_{k\in V} x_{ijku} \leq 1, \qquad \forall i \in N', \forall u \in T \tag{5.6}$$

$$\sum_{j\in N'} x_{0jku} \leq 1, \qquad \forall k \in V, \forall u \in T \tag{5.7}$$

$$\sum_{j\in N'} x_{jiku} - \sum_{j\in N'} x_{ijku} = 0, \qquad \forall i \in N', \forall k \in V, \forall u \in T \tag{5.8}$$

$$\sum_{i\in N'}\sum_{k\in V}\sum_{u\in T} x_{ijku} \cdot m_{ul} \geq n_{jl}, \qquad \forall j \in C, \forall l \in S \tag{5.9}$$

$$\sum_{i\in N'} x_{ijku} \leq h_{jk}, \qquad \forall j \in F', \forall k \in V_e, \forall u \in T \tag{5.10}$$

$$e_{ijk} \cdot fc_u \cdot x_{ijku} - (1 - x_{ijku}) \cdot Q_k \leq y_{ik} - y_{jk} \leq e_{ijk} \cdot fc_u \cdot x_{ijku} + (1 - x_{ijku}) \cdot Q_k, \qquad \forall i \in N', \forall j \in C, \forall k \in V_e, \forall u \in T \tag{5.11}$$

$$e_{ijk} \cdot fc_u \cdot x_{ijku} - (1 - x_{ijku}) \cdot Q_k \leq y_{ik} - q_{jk} \leq e_{ijk} \cdot fc_u \cdot x_{ijku} + (1 - x_{ijku}) \cdot Q_k, \qquad \forall i \in N', \forall j \in F', \forall k \in V_e, \forall u \in T \tag{5.12}$$

$$y_{ik} \geq e_{i0k} \cdot fc_u \cdot x_{i0ku}, \qquad \forall i \in N', \forall k \in V_e, \forall u \in T \tag{5.13}$$

$$y_{0k} = Q_k, \qquad \forall k \in V_e \tag{5.14}$$

$$y_{ik} = o_{ik}, \qquad \forall i \in F', \forall k \in V_e \tag{5.15}$$

$$q_{ik} \leq o_{ik}, \qquad \forall i \in F', \forall k \in V_e \tag{5.16}$$

$$q_{jk} \leq Q_k \cdot \sum_{i\in N'}\sum_{u\in T} x_{ijku}, \qquad \forall j \in F', \forall k \in V_e \tag{5.17}$$

$$o_{jk} \leq Q_k \cdot \sum_{i\in N'}\sum_{u\in T} x_{ijku}, \qquad \forall j \in F', \forall k \in V_e \tag{5.18}$$

$$\frac{q_{ik}}{Q_k} \cdot 100 \geq \sum_{l\in B} l \cdot \alpha_{ikl}, \qquad \forall i \in F', \forall k \in V_e \tag{5.19}$$

$$\frac{q_{ik}}{Q_k} \cdot 100 - 1 \leq \sum_{l\in B} l \cdot \alpha_{ikl}, \qquad \forall i \in F', \forall k \in V_e \tag{5.20}$$

$$\sum_{l\in B} \alpha_{jkl} \leq \sum_{i\in N'}\sum_{u\in T} x_{ijku}, \qquad \forall j \in F', \forall k \in V_e \tag{5.21}$$

$$\frac{o_{ik}}{Q_k} \cdot 100 \leq \sum_{l\in B} l \cdot \beta_{ikl}, \qquad \forall i \in F', \forall k \in V_e \tag{5.22}$$

$$\frac{o_{ik}}{Q_k} \cdot 100 + 1 \geq \sum_{l\in B} l \cdot \beta_{ikl}, \qquad \forall i \in F', \forall k \in V_e \tag{5.23}$$

$$\sum_{l\in B} \beta_{jkl} \leq \sum_{i\in N'}\sum_{u\in T} x_{ijku}, \qquad \forall j \in F', \forall k \in V_e \tag{5.24}$$

$$\Delta_{ik} = \sum_{l\in B} a_{ikl} \cdot \beta_{ikl} - \sum_{l\in B} a_{ikl} \cdot \alpha_{ikl}, \qquad \forall i \in F', \forall k \in V_e \tag{5.25}$$

$$\sum_{i\in N'}\sum_{j\in N'} w_{iju} = \sum_{i\in N'}\sum_{k\in V} x_{0jku}, \qquad \forall u \in T \tag{5.26}$$

$$w_{iju} \leq \sum_{k \in V} x_{ijku}, \qquad \forall i \in N', \forall j \in N', \forall u \in T \tag{5.27}$$

$$\tau_{iu} + w_{iju} \cdot (el_u - sl_u) + (t_{ij} + p_i) \cdot x_{ijku} - (et_u + (el_u - sl_u)) \cdot (1 - x_{ijku}) \leq \tau_{ju}, \qquad \forall i \in C, \forall j \in N', \forall k \in V_e, \forall u \in T \tag{5.28}$$

$$\tau_{iu} + w_{iju} \cdot (el_u - sl_u) + \Delta_{ik} + t_{ij} \cdot x_{ijku} - (S_{max} + et_u + (el_u - sl_u)) \cdot (1 - x_{ijku}) \leq \tau_{ju}, \qquad \forall i \in F', \forall j \in N', \forall k \in V_e, \forall u \in T \tag{5.29}$$

$$\tau_{iu} + p_i \leq sl_u \cdot \sum_{j \in N'} w_{iju} + et_u \cdot (1 - \sum_{j \in N'} w_{iju}), \qquad \forall i \in C, \forall k \in V, \forall u \in T \tag{5.30}$$

$$\tau_{iu} + \Delta_{ik} \leq sl_u \cdot \sum_{j \in N'} w_{iju} + et_u \cdot (1 - \sum_{j \in N'} w_{iju}), \qquad \forall i \in F', \forall k \in V_e, \forall u \in T \tag{5.31}$$

$$\tau_{iu} \geq el_u \cdot \sum_{j \in N'} w_{jiu} + st_u \cdot (1 - \sum_{j \in N'} w_{jiu}), \qquad \forall i \in N', \forall k \in V, \forall u \in T \tag{5.32}$$

$$sl_u \cdot z_{iju} - \tau_{iu} - p_i + et_u \cdot (1 - z_{iju}) \geq t_{ij} \cdot z_{iju}, \qquad \forall i, j \in N', \forall u \in T \tag{5.33}$$

$$\tau_{ju} - el_u \cdot z'_{iju} \geq t_{ij} \cdot z'_{iju}, \qquad \forall i, j \in N', \forall u \in T \tag{5.34}$$

$$z_{iju} + z^{'}_{iju} = w_{iju}, \qquad \forall i, j \in N', \forall u \in T \tag{5.35}$$

$$\tau_{iu} - t_{0i} \geq st_u, \qquad \forall i \in N', \forall u \in T, \tag{5.36}$$

$$\tau_{iu} + p_i + t_{i0} \leq et_u, \qquad \forall i \in C, \forall u \in T \tag{5.37}$$

$$\tau_{iu} + \Delta_{ik} + t_{i0} \leq et_u, \qquad \forall i \in F', \forall k \in V_e, \forall u \in T \tag{5.38}$$

$$ec_i \leq \tau_{iu} \leq lc_i, \qquad \forall i \in C, \forall u \in T \tag{5.39}$$

$$x_{ijku} \in \{0, 1\} \qquad \forall i, j \in N', \forall k \in V_e, \forall u \in T \tag{5.40}$$

$$x_{ijku} \in \{0, 1\} \qquad \forall i \in N', j \in N' \setminus F', \forall k \in V_c, \forall u \in T \tag{5.41}$$

$$\alpha_{ikl} \in \{0, 1\}, \;\; \beta_{ikl} \in \{0, 1\}, \qquad \forall i \in F', \forall k \in V_e, \forall l \in B \tag{5.42}$$

$$\tau_{iu} \geq 0, \qquad \forall i \in N', \forall u \in T \tag{5.43}$$

$$y_{ik} \geq 0, \qquad \forall i \in N', \forall k \in V_e \tag{5.44}$$

$$q_{ik} \geq 0, \;\; o_{ik} \geq 0, \;\; \Delta_{ik} \geq 0, \qquad \forall i \in F', \forall k \in V_e \tag{5.45}$$

$$w_{iju} \in \{0, 1\}, \;\; z_{iju} \in \{0, 1\}, \;\; z'_{iju} \in \{0, 1\}, \qquad \forall i, j \in N', \forall u \in T \tag{5.46}$$

The objective function (5.1) seeks to minimize the total cost (travel cost, parking cost, fixed charging cost, fixed cost of the technicians). Constraints (5.2) ensure that each request is visited once. Constraints (5.3) ensure that each copy if a CSs is visited at most once. Constraints (5.4) ensure that each arc is used by at most one vehicle and technician. Constraints (5.5-5.7) ensure that at most only one technician is allocated to one vehicle (and vice versa). Constraints (5.8) impose the flow conservation. Constraints (5.9) ensure that each request is visited by a technician with the appropriate skills. Constraints (5.10) ensure that EV $i \in V_e$ only charges at CSs with compatible technology. Constraints (5.11) and (5.12) track the battery level at each vertex for the EVs. Constraints (5.13) ensure that, if the EV travels between a vertex and the depot, it has enough remaining energy to reach its destination. Constraints (5.14) impose that the EVs depart from the depot with fully charged battery. Constraints (5.15) reset the battery tracking to o_i upon departure from CS $i \in F'$. Constraints (5.16) couple the battery level when an EV arrives to and departs from any CS. Constraints (5.17) and (5.18) ensure that the charge level is bounded by the battery capacity when an EV arrives at and departs from CS $j \in F'$. Constraints (5.19-5.24) define the battery level in percentage when vehicle $k \in V_e$ arrives at or departs from the CS $i \in F'$. Constraints (5.25) define the time spent charging at a CS. Constraints (5.26) ensure that the technician $u \in T$ has to take the lunch break. Constraints (5.27) ensure that the lunch break is taken between two visited vertices. Constraints (5.28) and (5.29) track the arrive time at each vertex. Constraints (5.30)-(5.32) ensure that technician $u \in T$ takes the lunch break between sl_u and el_u. Constraints (5.33-5.35) ensure that the technician takes the lunch after or before travelling to vertex $j \in N'$. Constraints (5.36-5.38) impose the working schedule of the technician. Constraints (5.39) ensure that every request is served within its time window. Finally, constraints (5.40-5.46) define the domain of the decision variables. Note that variables x_{ijku} are not defined for $j \in F'$ and $k \in V_c$

5.4 Parallel matheuristic

Considering the complexity of th TRP-CEV in terms of the number of constraints, we decide to use an approach based on decomposition. As the name suggest, a decomposition approach decomposes the problem in a set of sub-problems that consider less information or less constraints than the original problem. Each sub-problem is solved individually, and the solutions found for each sub-problem are used to find a solution of the original problem. This kind of approaches has been successful in the solution of different rich VRPs (Doerner & Schmid 2010).

One alternative to address the set of sub-problems is to solve them in parallel using metaheuristics. Parallel metaheuristics are commonly used to solve VRPs (Crainic 2008). They focus either on decomposing the solution approach (e.g., using different local search strategies in parallel), or on decomposing the problem domain (e.g., into geographic zone, or type of vehicle). We follow the latter approach. Successful examples of this approach include (Taillard 1993) that proposes a parallel tabu search for the capacitated VRP. In his method, the problem is divided spatially into regions and a problem is solved individually for each region with tabu search. Doerner et al. (2005) follow the same decomposition principle but use and savings based ants system (Reimann et al. 2004) for each sub-problem. Taillard (1999) propose a heuristic column generation for the HVRP. This approach decomposes the HVRP in a set of homogeneous VRP (one for each type of vehicle), and solve each VRP using an adaptive memory programming procedure. Then, set partitioning formulation assembles a HVRP solution.

Our method shares the spirit of Taillard (1999). Algorithm 4 describes the general structure of our PMa. In order to avoid symmetry problems, the algorithm starts by calling the procedure `groupTechnicians(`T`)` –line 2–. This procedure groups the technicians that have the same characteristics (i.e., skills, fixed cost, working schedule and lunch break), and generates the set E of technician types. Then, the algorithm invokes the procedure `buildCouples(`E`,`K`)` –line 3–. It builds the set P of all possible couples computing the cartesian product $P = E \times K = \{(u, k)|u \in E, k \in K\}$. Then, the algorithm starts in a parallel phase –lines 5-9–. In this phase, for each couple $p \in P$ it solves a VRP-TWLB$_p$ or an eVRP-TWLB$_p$ on

each thread (the problem to be solved depends of the type of vehicle used a CV or an EV). To solve each (e)VRP-TWLB$_p$ we ignore the fleet size and the limit on the number of technicians. Before to solve each (e)VRP-TWLB$_p$, we call the procedure `feasibleRequests(`G, p`)` that identifies the feasible requests C_p for the couple $p \in P$. The details of this procedure are explained in Section 5.4.1. Then, the algorithm invokes the metaheuristic `GRASP(`p, C_p, G`)` to solve the corresponding (e)VRP-TWLB$_p$ –line 7–. This metaheristic returns a set Ω_p of feasible routes for the underlying problem. The set Ω_p joins the global set Ω. After completing the parallel phase, the algorithm calls the procedure `SetCovering(`G, Ω, K, E`)`– line 10–, which solves an extended set covering formulation over Ω to find a feasible TRP-CEV solution. This formulation includes constraints on the number of technicians and vehicles. Finally, the resulting solution σ is reported by PMa –line 11–.

Algorithm 4 Parallel matheuristic: General structure

```
1:  function ParallelMatheuristic(G,T,K)
2:      E ⟵ groupTechnicians(T)
3:      P ⟵ buildCouples(E, K)
4:      Ω ⟵ ∅
5:      parallel for each p ∈ P
6:          C_p ⟵ feasibleRequests(G, p)
7:          Ω_p ⟵ GRASP(p, C_p, G)
8:          Ω ⟵ Ω ∪ Ω_p
9:      end for
10:     s ⟵ SetCovering(G, Ω, K, E)
11:     return σ
12: end function
```

5.4.1 Identifying feasible requests

Considering that each couple $p \in P$ has specific features associated to the type of technician, and the type of vehicle, before solving the (e)VRP-TWLB$_p$ we identify the feasible requests C_p for the couple $p \in P$ using a set of criteria. To explain those criteria, we use the notation introduced in Section 5.3, and we denote u^* and k^* be the type of technician and type of vehicle of couple $p \in P$.

Skill criterion: If the type of technician u^* has at least the required skills of the request $i \in C$ (i.e., $m_{u^*l} \geq n_{il}, \forall l \in S$), then the request $i \in C$ is part of the set of feasible requests (i.e, $C_p = C_p \cup \{i\}$).

Time window - working schedule criterion: If the latest time lc_i of time window request $i \in C$ minus the travel time from depot to request i is greater than the start time of the working schedule of the type of technician u^* (i.e, $lc_i - t_{0i} \geq st_{u^*}$); and the earliest time lc_i of time window request $i \in C$ plus the its service time plus the travel time from request i to depot is lower than the end time of the working schedule of the type of technician u^* (i.e, $ec_i + p_i + t_{i0} \leq et_{u^*}$), then $C_p = C_p \cup \{i\}$.

Time window - lunch break criterion: If the time window $[ec_i, lc_i]$ of request $i \in C$ is not completely contained in the lunch break of the type of technician u^* (i.e., $[ec_i, lc_i] \not\subset [sl_u, el_u]$), then $C_p = C_p \cup \{i\}$.

Autonomy criterion: If the battery capacity Q_{k^*} of the type of EV k^* is greater than the minimum energy needed to reach request $i \in C$ from any CSs or the depot (i.e., $Q_{k^*} \geq min_{j \in F \cup \{0\}}\{e_{jik^*}\}$), then $C_p = C_p \cup \{i\}$.

5.4.2 GRASP

To solve the (e)VRP-TWLB$_p$ associated to the couple $p \in P$, we developed a GRASP (Feo & Resende 1995). This metaheuristic is a memory-less multi-start method in which local search is applied to several

initial solutions constructed with a greedy randomized heuristic. GRASP can be hybridized in different ways, for instance by replacing the local search with an other metaheuristic such as tabu search, simulated annealing, variable neighborhood search (VND), among others (Resende 2008). In this case, to build the initial solution we use a randomized Solomon heuristic (RSH). Then at each iteration of the GRASP, we improve the current solution using a VND (Mladenović & Hansen 1997). At each iteration we add a set of routes to the pool of routes Ω_p. To favour the diversity of the routes in this pool, we add to Ω_p the routes of the initial solution found by RSH and the routes of the local optimum after the VND. The execution of GRASP stops after I iterations and we add the routes of Ω_p to the global pool Ω (i.e., $\Omega = \Omega_p \cup \Omega$). A brief description of the components of the GRASP follows:

Initial solution

We generate initial solutions for the (e)VRP-TWBL$_p$ using a randomized Solomon heuristic (Solomon 1987). In this heuristic, the seed request is randomly selected, and instead of selecting the best insertion, an insertion is randomly selected among the κ best insertions.

To evaluate the feasibility of a request insertion, we first check if it satisfies the time window, working schedule, and lunch break constraints. Using the concepts introduced by Savelsbergh (1992), this verification is done in constant time. Then, if request is to be inserted on a route covered by an EV, we evaluate if the insertion satisfies the energy constraint. If it does not, we try to turn the insertion feasible by inserting visits to CSs into the route. If we cannot achieve our goal, we discard that request insertion.

The problem of inserting visits to CSs into an energy-infeasible route (i.e., a route that is infeasible in terms of the energy autonomy of the vehicle) can be modelled using a fixed-route vehicle-charging problem (FRVCP) as in Chapter 4. In this case, the FRVCP has to consider time windows and the working schedule. We refer to this extension of the FRVCP simply as the FRVCP with time windows (FRVCP-TW). To favour the computational efficiency during the construction of initial solutions, we use a simple heuristic to solve the FRVCP-TW. This heuristic is explained in Section 5.5.

Variable neighborhood descent

To improve the solution generated by the RSH we use a VND based on three local search operators [2]. The first two operators, namely, relocated and swap, focus on the request sequencing decisions. On the other hand, the third operator is an ejection chain, aimed at reducing the number of routes (Rousseau et al. 2002). To evaluate the feasibility of the moves performed by the operators, we use the same steps employed during the generation of the initial solutions.

Global charging improvement

If the GRASP is solving an eVRP-TWLB, after the VND we try to improve the charging decisions using a procedure based on the approach introduced in Section 4.3.2. This procedure called global charging improvement, or GCI, is applied to every route visiting at least one CS. The procedure works as follows. First, it removes from the route all visits to CSs. If the resulting route is energy-feasible, the procedure stops. If the route is energy-infeasible, the procedure solves a FRVCP-TW on the concerned route. In this case, to solve the FRVCP-TW we use a commercial optimizer to solve an MILP formulation of the problem. Full details on the MILP formulation of the FRVCP-TW are given in Section 5.5.

5.4.3 Set covering

To find a feasible TRP-CEV solution, we solve a set-covering (SC) formulation over the pool of routes Ω. To formulate the SC we use the following parameters: c_r is the cost of the route $r \in \Omega$, b_{ri} is equal to

[2] In our implementation we use intra-route and inter-route versions with best-improvement selection.

1 if route $r \in \Omega$ serves request $i \in C$ and 0 otherwise, pv_{rk} is equal to 1 if route $r \in \Omega$ is performed by a vehicle of type $k \in K$ and 0 otherwise, h_{ru} is equal to 1 if route $r \in \Omega$ is performed by a technician of type $u \in E$ and 0 otherwise, and n_u is the number of technicians of type $u \in E$. We denote by χ_r a binary decision variable that takes the value of 1 if route $r \in \Omega$ is selected and 0 otherwise. We formulate the SC as follows:

$$\min \sum_{r \in \Omega} c_r \cdot \chi_r \tag{5.47}$$

Subject to

$$\sum_{r \in \Omega} b_{rj} \cdot \chi_r \geq 1 \qquad \forall j \in C \tag{5.48}$$

$$\sum_{r \in \Omega} pv_{rk} \cdot \chi_r \leq m_k \qquad \forall k \in K \tag{5.49}$$

$$\sum_{r \in \Omega} h_{ru} \cdot \chi_r \leq n_u \qquad \forall u \in E \tag{5.50}$$

$$\chi_r \in \{0, 1\} \qquad \forall r \in \Omega \tag{5.51}$$

The objective function (5.47) seeks to minimize the total cost. Constraints (5.48) ensure that each request is served at least once. Constraints (5.49) impose the maximum number of vehicles of each type. Constraints (5.50) impose the maximum number of technicians of each type. Finally, constraints (5.51) define the domain of the decision variables.

Considering that the requests must be served exactly once, a set-partitioning problem may seem more appropriate. However, as pointed out by Pillac et al. (2013) for a similar technician problem, finding a good combination of routes that visit all requests exactly once could be complex in some cases. Therefore, we opted to use a SC formulation instead of a set-partitioning formulation. In the case that the solution found by the SC formulation serves a request more than once, we remove the most costly duplicate visits using a greedy heuristic.

5.5 The fixed-route vehicle charging problem with time windows

As previously mentioned, our GRASP for the eVRP-TWLB relies on solving the FRVCP-TW when we have to take the charging decision of a given route. This problem is an extension of the FRVCP introduced in Section 4.3, that includes the time window constraints, and the working schedule of the technician.

Let $\Pi = \{\pi(0), \pi(1), ..., \pi(i), ..., \pi(j), ..., \pi(n_r)\}$ be a route that violates the energy constraint, where $\pi(0)$ and $\pi(n_r)$ represent the depot. The feasibility of Π may be restored by inserting visits to CSs. The route Π is performed by a technician $u \in T$ using an EV of type $k \in K_e$. The technician has a working schedule $[st_u, et_u]$ and a lunch break $[sl_u, el_u]$. For the route Π, we consider as fixed the decisions between which vertices the technician takes the lunch break. Let $\pi(s)$ and $\pi(e)$ be the requests served right before and after the lunch break. To model the lunch break into the route Π, we create a dummy vertex $\pi(l)$, which has a time window $[sl_u, sl_u]$, and a service time $p_l = el_u - sl_u$. Then, we define $\overline{\Pi} = \{\pi(0), ..., \pi(s), \pi(l), \pi(e), ..., \pi(n_r)\}$ as the route that includes the dummy vertex $\pi(l)$. Using the notation introduced in section 5.3, if the technician takes the lunch break before travelling from $\pi(s)$ to $\pi(e)$, then $d_{\pi(s)\pi(l)} = 0$, $t_{\pi(s)\pi(l)} = 0$, $e_{\pi(s)\pi(l)k} = 0$, and $d_{\pi(l)\pi(e)} = d_{\pi(s)\pi(e)}$, $t_{\pi(l)\pi(e)} = t_{\pi(s)\pi(e)}$, $e_{\pi(s)\pi(l)k} = e_{\pi(s)\pi(e)k}$. On the contrary, if the technician takes the lunch break after travelling from $\pi(s)$ to $\pi(e)$, then $d_{\pi(s)\pi(l)} = d_{\pi(s)\pi(e)}$, $t_{\pi(s)\pi(l)} = t_{\pi(s)\pi(e)}$, $e_{\pi(s)\pi(l)k} = e_{\pi(s)\pi(e)k}$, and $d_{\pi(l)\pi(e)} = 0$, $t_{\pi(l)\pi(e)} = 0$, $e_{\pi(s)\pi(l)k} = 0$.

In the FRVCP-TW the objective is to find the charging decisions that minimize the sum of the travel cost, the fixed charging cost, and the parking cost, satisfying the following conditions: the level of the

battery when the vehicle arrives at any vertex is nonnegative; the requests must be visited within their time windows; and the route satisfies the working schedule and the lunch break of the technician.

We model the FRVCP-TW using an MILP formulation with the following decision variables: variable $\varepsilon_{\pi(i)j}$ is equal to 1 if the vehicle charges at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$. Variable $\phi_{\pi(i)}$ tracks the battery level. If $\varepsilon_{\pi(i)j} = 0$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at vertex $\pi(i)$. On the other hand, if $\epsilon_{\pi(i)j} = 1$, $\phi_{\pi(i)}$ is the battery level when the vehicle arrives at CS $j \in F$ right before visiting vertex $\pi(i)$. Variable $\gamma_{\pi(i)j}$ represents the energy charged at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$. Variables $q_{\pi(i)j}$ and $o_{\pi(i)j}$ represent the battery level when the EV arrives at and departs from CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$. Variables $\alpha_{\pi(i)jl}$ and $\beta_{\pi(i)jl}$ are equal to 1 if the battery level is $l \in B$ when the EV arrives at and departs from the CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$, respectively. Variable $\delta_{\pi(i)j}$ represents the time spent at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$. Variable $\tau_{\pi(i)}$ represent the arrival time at the vertex $\pi(i) \in \overline{\Pi}$. The MILP formulation of the FRVCP-TW follows:

$$\min \sum_{\pi(i)\in\overline{\Pi}\setminus\{\pi(0)\}} \sum_{j\in F} \Big(tc_k \cdot (d_{\pi(i-1)j} + d_{j\pi(i)} - d_{\pi(i-1)\pi(i)}) \cdot \varepsilon_{\pi(i)j} + g_k \cdot \varepsilon_{\pi(i)j} + pc_j \cdot \delta_{\pi(i)j}\Big) \tag{5.52}$$

Subject to

$$\phi_{\pi(1)} = Q_k - \sum_{j\in F} e_{\pi(0)jk} \cdot \varepsilon_{\pi(1)j} - e_{\pi(0)\pi(1)k} \cdot (1 - \sum_{j\in F} \varepsilon_{\pi(1)j}) \tag{5.53}$$

$$\phi_{\pi(i)} = \phi_{\pi(i-1)} + \sum_{j\in F} \gamma_{\pi(i-1)j} - \sum_{j\in F} e_{j\pi(i-1)k} \cdot \varepsilon_{\pi(i-1)j} - \sum_{j\in F} e_{\pi(i-1)jk} \cdot \varepsilon_{\pi(i)j} - e_{\pi(i-1)\pi(i)k} \cdot (1 - \sum_{j\in F} \varepsilon_{\pi(i)j}) \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0), \pi(1), \pi(n_r)\} \tag{5.54}$$

$$\phi_{\pi(n_r)} = \phi_{\pi(n_r-1)} + \sum_{j\in F} \gamma_{\pi(n_r-1)j} + \sum_{j\in F} \gamma_{\pi(n_r)j} - \sum_{j\in F} e_{j\pi(n_r-1)k} \cdot \varepsilon_{\pi(n_r-1)j} - \sum_{j\in F} \Big(e_{\pi(n_r-1)jk} + e_{j\pi(n_r)k}\Big) \cdot \varepsilon_{\pi(n_r)j} - e_{\pi(n_r-1)\pi(n_r)k} \cdot \Big(1 - \sum_{j\in F} \varepsilon_{\pi(n_r)j}\Big) \tag{5.55}$$

$$\phi_{\pi(n_r-1)} + \sum_{j\in F} \gamma_{\pi(n_r-1)j} - \sum_{j\in F} e_{j\pi(n_r-1)k} \cdot \varepsilon_{\pi(n_r-1)j} - \sum_{j\in F} e_{\pi(n_r-1)jk} \cdot \varepsilon_{\pi(n_r)j} \geq 0 \tag{5.56}$$

$$\phi_{\pi(i)} - Q_k \cdot (1 - \varepsilon_{\pi(i)j}) \leq q_{\pi(i)j} \leq \phi_{\pi(i)} + Q_k \cdot (1 - \varepsilon_{\pi(i)j}) \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.57}$$

$$o_{\pi(i)j} = q_{\pi(i)j} + \gamma_{\pi(i)j} \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.58}$$

$$\sum_{l\in B} l \cdot \alpha_{\pi(i)jl} \leq \frac{q_{\pi(i)j}}{Q_k} \cdot 100 \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.59}$$

$$\sum_{l\in B} l \cdot \alpha_{\pi(i)jl} \geq \frac{q_{\pi(i)j}}{Q_k} \cdot 100 - 1 \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.60}$$

$$\sum_{l\in B} l \cdot \beta_{\pi(i)jl} \geq \frac{o_{\pi(i)j}}{Q_k} \cdot 100 \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.61}$$

$$\sum_{l\in B} l \cdot \beta_{\pi(i)jl} \leq \frac{o_{\pi(i)j}}{Q_k} \cdot 100 + 1 \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.62}$$

$$\sum_{l\in B} \alpha_{\pi(i)jl} \leq \varepsilon_{\pi(i)j} \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.63}$$

$$\sum_{l\in B} \beta_{\pi(i)jl} \leq \varepsilon_{\pi(i)j} \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.64}$$

$$\delta_{\pi(i)j} = \sum_{l\in B} a_{jkl}\beta_{\pi(i)jl} - \sum_{l\in B} a_{jkl}\alpha_{\pi(i)jl} \quad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \tag{5.65}$$

$$\tau_{\pi(i)} \geq \tau_{\pi(i-1)} + p_{\pi(i-1)} + \sum_{j\in F} \varepsilon_{\pi(i)j} \cdot \Big(t_{\pi(i-1)j} + t_{j\pi(i)}\Big) +$$

$$t_{\pi(i-1)\pi(i)} \cdot \left(1 - \sum_{j \in F} \varepsilon_{\pi(i)j}\right) + \sum_{j \in F} \delta_{\pi(i)j} \qquad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0)\}, \forall j \in F \qquad (5.66)$$

$$ec_{\pi(i)} \leq \tau_{\pi(i)} \leq lc_{\pi(i)} \qquad \forall \pi(i) \in \overline{\Pi} \setminus \{\pi(0), \pi(n_r)\} \qquad (5.67)$$

$$st_u \leq \tau_{\pi(i)} \leq et_u \qquad \forall \pi(i) \in \overline{\Pi} \qquad (5.68)$$

$$\sum_{j \in F} \varepsilon_{\pi(i)j} \leq 1 \qquad \forall \pi(i) \in \overline{\Pi} \qquad (5.69)$$

$$\gamma_{\pi(i)j} \leq Q_k \cdot \varepsilon_{\pi(i)j} \qquad \forall \pi(i) \in \overline{\Pi}, \forall j \in F \qquad (5.70)$$

$$\gamma_{\pi(i)j} \geq 0, \delta_{\pi(i)j} \geq 0 \qquad \forall \pi(i) \in \overline{\Pi}, \forall j \in F \qquad (5.71)$$

$$\varepsilon_{\pi(i)j} \in \{0, 1\} \qquad \forall \pi(i) \in \overline{\Pi}, \forall j \in F \qquad (5.72)$$

$$\phi_{\pi(i)} \geq 0 \qquad \forall \pi(i) \in \overline{\Pi} \qquad (5.73)$$

$$\alpha_{\pi(i)jl} \in \{0, 1\}, \beta_{\pi(i)jl} \in \{0, 1\}, \qquad \forall \pi(i) \in \overline{\Pi}, \forall j \in F, \forall l \in B \qquad (5.74)$$

$$q_{\pi(i)j} \geq 0, \quad o_{\pi(i)j} \geq 0, \qquad \forall \pi(i) \in \overline{\Pi}, \forall j \in F \qquad (5.75)$$

The objective function (5.52) seeks to minimize the total cost given by the sum of the travel cost, the fixed charging cost, and the parking cost. Constraints (5.53-5.56) define the battery level when the EV arrives at vertex $\pi(i) \in \overline{\Pi}$ if $\varepsilon_{\pi(i)j} = 0$, or to CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$, if $\varepsilon_{\pi(i)j} = 1$. Constraints (5.57 and 5.58) define the battery level when the EV arrives at and departs from CS $j \in F$. Constraints (5.59-5.64) define the battery level in percentage when the EV arrives at or depart from CS $j \in F$. Constraints (5.65) define the time spent at each visited CS. Constraints (5.66) track the arrival time at each vertex. Constraints (5.67) ensure that every vertex is visited within its time window. Constraints (5.68) impose the working schedule of the technician. Constraints (5.69) state that only one CS is visited between any two vertices of the fixed route. Constraints (5.70) ensure that the EV charges only at a visited CSs. Finally, constraints (5.71-5.75) define the domain of the decision variables.

To solve the FRVCP-TW, we use two different solution approaches. The first solves the MILP formulation introduced using a commercial optimizer (we use this approach in the GCI component). The second is a greedy heuristic (we use this approach in the initial solution and the VND component).

5.5.1 Greedy heuristic

We extended here the heuristic proposed in Section 4.3.2 to include the time window constraints. Algorithm 5 describes the structure of our heuristic. It uses five important procedures `trackTimeVariables`$(\cdot)$, `trackBattery`$(\cdot)$, `sumNegative`$(\cdot)$, `totalCost`$(\cdot)$, and `copyAndInsert`$(\cdot)$. Procedure `trackTimeVariables`$(\cdot)$ computes the earliest departure time $D_{\pi(i)}$ and the latest feasible arrival times $L_{\pi(i)}$ at each vertex $\pi(i) \in \overline{\Pi}$. Procedure `trackBattery`$(\cdot)$ computes the battery level $Y_{\pi(i)}$ at each vertex $\pi(i) \in \overline{\Pi}$. Note that $Y_{\pi(i)}$ may take negative values. Procedure `sumNegative`$(\cdot)$ computes the sum of the battery levels with negative values (i.e., $sn = \sum_{\pi(i) \in \overline{\Pi}} min\{0, Y_{\pi(i)}\}$). Procedure `totalCost`$(\cdot)$ computes the total cost c of the route. Finally, procedure `copyAndInsert`$(\cdot)$ takes as input a fixed route, a CS, and a position in the route; and returns a copy of the fixed route with the CS inserted at the given position.

The heuristic starts computing $D_{\pi(i)}$, $L_{\pi(i)}$ for each vertex, and the sum sn of the battery levels with negative values for the current fixed-route $\overline{\Pi}$ (line 2-7). Then the heuristic enters the outer loop (line 8-38). In each pass through the inner loop (line 9-31), the heuristic: i) computes a slack time st at CS F_j between the vertices $\pi(i)$ and $\pi(i+1) \in \overline{\Pi}$, and evaluates if the slack time is positive (line 11-12), ii) computes Y' and sn, considering that the amount of charge at the inserted CS F_j is $min\{\omega(st), ec, Q_k\}$, where $\omega(st)$ is the the amount of energy that the EV can charge in st time units, and ec is the energy needed to complete the fixed route (reach the depot) from CS F_j, iii) selects the insertion that maximizes sn (lines 18-22). If $sn = 0$ (i.e., the route is energy-feasible), the heuristic selects the insertion (CS and position) that minimizes the cost of the route c (line 23-28). If there is at least one feasible insertion (i.e, insertion=true), then the heuristic performs the selected insertion (line 32-34). When no feasible insertion exists, the route can not be

repaired and the algorithm stops (line 35-36). If after inserting the CS, the route Π is still energy-infeasible (i.e., $sn < 0$), the heuristic starts again at line 9 and tries to insert additional CSs until feasibility in terms of energy is reached.

Algorithm 5 Greedy heuristic

```
1: function GREEDYHEURISTIC(Π̄^0, F)
2:     Π̄ ⟵ Π̄^0
3:     < D, L > ⟵ trackTimeVariables(Π̄)
4:     Y ⟵ trackBattery(Π̄)
5:     sn ⟵ sumNegative(Y)
6:     c ⟵ ∞
7:     insertion ⟵ false
8:     while sn < 0 do
9:         for j = 1 to |F| do
10:            for i = 0 to n_r − 1 do
11:                st ⟵ L_{π(i+1)} − D_{π(i)} − t_{π(i)F_j} − t_{F_j,π(i+1)}
12:                if st ≥ 0 then
13:                    Π̄′ ⟵ copyAndInsert(Π̄, F_j, i)
14:                    Y′ ⟵ trackBattery(Π̄′)
15:                    sn′ ⟵ sumNegative(Y′)
16:                    c′ ⟵ totalCost(Π̄′)
17:                    insertion ⟵ true
18:                    if sn′ > sn then
19:                        sn ⟵ s′
20:                        u ⟵ j
21:                        v ⟵ i
22:                    end if
23:                    if sn′ = 0 and c′ < c then
24:                        sn ⟵ s′
25:                        c ⟵ c′
26:                        u ⟵ j
27:                        v ⟵ i
28:                    end if
29:                end if
30:            end for
31:        end for
32:        if insertion= true then
33:            Π̄ ⟵ copyAndInsert(Π̄′, F_u, i)
34:            < D, L > ⟵ trackTimeVariables(Π̄)
35:        else
36:            return false, Π̄
37:        end if
38:    end while
39:    return true, Π̄
40: end function
```

5.6 Computational experiments

5.6.1 Test instances for the TRP-CEV

We generate a set of large instances using the data of ENEDIS operation. Maintenance and repair operation of ENEDIS is divided into geographic zones. Each zone has a depot where the technicians start and end their routes. We have 14 test instances classified according to the type geographic zone: four urban, five semi-urban and five rural instances. The number of requests ranges between 54 and 167. The location, time windows, service time, and required skills of the requests come from ENEDIS data. To complete the instances, we added the actual CSs located in the geographic zone of the requests. We obtained the information about of the location and type of technology (i.e., protocol and charging mode) from the open data platform of the French government (Quest 2014). To compute the distance and travel time matrices, we use road network data from Open Street Maps (OSM). Furthermore, to build the energy consumption matrices, we use the energy consumption equations model of De Cauwer et al. (2015). As input of those equations, we use fine-grained information on the roads (e.g., distance, speed, elevation) from OSM and NASA, weather information from Metéo France, and real EV parameters (e.g, mass, rolling resistance).

The number of available technicians ranges between 9 and 12. The number of considered skills is two. The skills, working schedule, and lunch break of each technician correspond to real data of the operation. For confidentiality reasons, the cost of this problem was modified but we respect the magnitude and the structure of the ENEDIS operation costs. We consider one type of CVs (Renault Kangoo) and two types of EVs (Renault Kangoo ZE and Peugeot Ion). We took all data related to EVs features from technical sources and specialized websites (Renault 2014, Peugeot 2015, Automobile-propre 2015, Uhrig et al. 2015). In order to better understand the results of the following sections, it is important to know that the travel cost of a Renault Kangoo is much higher than that of a Renault Kangoo ZE and a Peugeot Ion (i.e., 1 vs. 0.08 and 0.06 €$/km$). For each instance, we generate 5 scenarios changing the percentage of EVs in the fleet (0%, 20%, 40%, 60%, 80% and 100%).

In order to compare the results of our PMa against optimal solutions, we generate a set of small instances with 5 or 10 requests, and 2 or 3 technicians and vehicles. These small instances can be solved using the MILP formulation of the TRP-CEV. To generate each small instance, we randomly selected a large instance, and then randomly drawn from it, requests, technicians, and types of vehicles (ensuring that at least one vehicle is an EV). Our large and small instances are publicly available at `www.vrp-rep.org` (Mendoza et al. 2014).

5.6.2 Parameter settings & experimental environment

The parameters of our PMa is the number of iterations of the GRASP component (i.e., I) and the κ best insertion of the randomized Solomon heuristic. To fine tune I and κ, we conducted a short computational study. For the sake of brevity we do not discuss the study in this chapter. Our results showed that $I = 50$ and $\kappa = 10$ provides the best trade-off between solution quality and computational performance. Although, $I = 50$ may seem as a small number of iterations, notice that the GRASP component is used to feed the pool of routes Ω for each couple $< k, u >$. A large pool of routes impacts the performance of the solution of the SC formulation.

We implemented our PMa in Java (jre V.1.8.0) and used Gurobi Optimizer (version 6.0) to solve the FRVCP-TW and the SC formulation. We set a time limit of 800 seconds on Gurobi to control the running time of the SC. All the experiments were run on a computing cluster with 2.33 GHz Intel Xeon E5410 processors with 16 GB of RAM running under Linux Rocks 6.1.1.

5.6.3 Performance of PMa

Since the TRP-CEV is a new problem, there are no results or algorithms to benchmark against. To get an idea of the quality of the solutions delivered by our PMa, we conducted different experiments. We ran

our PMa on the small instances and compared their results with optimal solutions found using the MILP formulation. Furthermore, we ran our PMa on a set of 14 private industrial-scale instances of ENEDIS and compared our results with solutions found by the routing tool used by the company. Finally, we ran our PMa on a set of instances of a problem related with the TRP-CEV, and compared our results with the results obtained by the approach proposed in the literature.

Results on small TRP-CEV instances

Table 5.1 compares the results of our PMa with the optimal solution on the small instances of the TRP-CEV. In this table, we compare the results under the light of four metrics: the number of optimal solutions found, the average gap with respect to the optimal solution over the 10 runs, the average best gap[3], and the average computing time. The results of this table suggest that PMa is able to deliver high-quality TRP-CEV solutions for small instances. PMa matched the 10 optimal solutions and the values reported for Avg. Gap shows that our approach is stable for those instances. Appendix A.3 presents detailed results for the 10 instances.

Table 5.1 – Comparison of the PMa on small instances with proven optima

Metric	**PMa**
Number of optimal solutions	10/10
Avg. Gap (%)	0.00
Avg. Best Gap (%)	0.00
Avg. Time (s)	0.55

Results on private industrial-scale instances

Table 5.2 compares the results of our PMa with the results of the professional routing tool currently deployed at ENEDIS on the set of large instances. It is worth noting that their tool cannot deal with EVs. Therefore, the instances used in this section only consider CVs. Table 2 uses the same metrics of Table 1, but the number of optimal solutions is replaced by the number of best known solutions (BKSs) found by the routing software or by our PMa. Since we only have one solution of the routing tool for each instance, we only compute the average gap for the routing software [4].

Table 5.2 – Comparison of the PMa with the routing software used by ENEDIS

Metric	**Routing software**	**PMa**
Number of BKSs	0/14	14/14
Avg. Gap (%)	6.56	2.71
Avg. Best Gap (%)	-	0.00
Avg. Time (min)	-	3.39

The results in Table 5.2 show that our PMa consistently outperforms the current routing tool used by the company. It not only found the 14 BKSs but also reported a significantly lower average gap (2.71% vs. 6.56%).

[3]The best gap is the gap between the best solution found over 10 runs and the optimal solution

[4]For confidentiality reasons, we do not report detailed results for each instance in this testbed

Results for the E-FSMFTW instances

Hiermann et al. (2016) recently introduced the closely-related E-FSMFTW. This problem consists in routing an unlimited heterogeneous fleet of EVs to serve a set of geographically scattered requests within their time windows. For this problem Hiermann et al. (2016) assume that customers have a positive demand, the EVs have limited hauling capacity, all CSs have the same technology, the EVs charge fully their battery at CSs, the charging function is linear, the consumption rate is constant, and the EVs have a fixed cost. The objective function of E-FSMFTW seeks to minimize the sum of the cost of the fixed cost of used vehicles plus the total travelled distance.

We adapted our PMa to solve the E-FSMFTW instances. It is worth noting that we do not aim to establish new state-of-the-art results for the E-FSMFTW. We rather want to show that our PMa can also be competitive on a related problem with only mild modifications.

Hiermann et al. (2016) proposed a set of 108 small instances with the number of customers ranging from 5 to 15, and a set of 168 large instances with 100 customers and 21 CSs. To solve those instances, Hiermann et al. (2016) propose an adaptive large neighbourhood search (ALNS) and a branch-and-price (BnP). Their BnP was able to obtain the optimal solution within the time limit of two hours for all the small instances.

Tables 5.3 and 5.4 summarize the results delivered by our PMa and their ALNS on the small and large E-FSMFTW instances. For the small instances, we compare the results obtained by PMa and ALNS with the optimal solutions obtained by BnP. For the large instances, we compare the results obtained by PMa and ALNS with the BKSs taken from Hiermann et al. (2016) [5] and updated with some new BKSs found by our PMa. The results of the PMa and ALNS are computed over 10 runs. It is important to mention, that Hiermann et al. (2016) only report the average cost of the solutions that they obtained for each large instances. Appendix A.4 reports the detailed results of this experiment.

Table 5.3 – Comparison of our PMa with the ALNS by Hiermann et al. (2016) on small instances

Metric	**ALNS**	**PMa**
Number of optimal solutions	108/108	81/108
Avg. Gap (%)	0.55	0.32
Avg. Best Gap (%)	0.00	0.31
Avg. Time (min)	0.32	0.06
Computer	Core2 Quad with 2.4 GHz	XEON E5410 with 2.33 GHz

Table 5.4 – Comparison of our PMa with the ALNS by Hiermann et al. (2016) on large instances

Metric	**ALNS**	**PMa**
Number of BKS	NR	69
Avg. Gap (%)	1.18%	2.20%
Avg. Best Gap (%)	NR	1.52%
Avg. Time (min)	22.66	25.17
Computer	Intel Core2 Quad CPU Q6600 with 2.4 GHz	XEON E5410 with 2.33 GHz

On the small instances, PMa delivered competitive results with respect to ALNS. Although, PMa just matched 81 optimal solutions, its average gap is smaller than the achieved by ALNS (0.32% vs 0.55%). In terms of CPU time, PMa is faster than the ALNS.

[5] Hiermann et al. (2016) report the BKSs obtained using BnP or any run of their experiments

The results for large instances show that PMa obtains competitive results, especially considering that it was not originally conceived to tackle the E-FSMFTW (Table 5.4). Our PMa matched 69/168 BKSs; 61 are new BKSs [6]. However, the PMa has a higher average gap (1.18% vs 2.20%). After carefully analyzing the results, we believe this behavior may be explained by the characteristic of the Hiermann et al. (2016) instances. Indeed, in their instances, since some of the EVs have an autonomy of about 50 km (which is much lower than in reality). In consequence, it is possible that in order to reach some isolate customers it may be interesting (from a cost point of view) to visit more than one CSs between two customers. Now, this possibility is ruled out in our PMa. Note that in the definition of our FRVCP-TW visiting more than one CS between two customers (hereafter called multiple consecutive visits) is forbidden. This was a deliberated design decision motivated by three interrelated reasons. First, forbidding multiple consecutive visits vastly simplifies and speeds up solving the FRVCP-TW (either with the heuristic or the solver). Second, we conceived our algorithm to work on a problem where there is fixed cost for charging that largely offsets the driving cost of an EV. Therefore, multiple consecutive visits are naturally discourage. Third, our algorithm is meant to be used in a real-world context, where EVs have larger autonomies and, therefore, are less likely to require multiple consecutive visits to reach a customer. We believe that the trade off between missing some high-quality routes during the search and having a faster and simpler algorithm is positive.

5.6.4 Managerial insight

ENEDIS is starting the migration from a fleet of CVs to a mixed fleet of CVs + EVs. This process raises a number of managerial questions regarding the fleet composition and the best strategies to exploit EVs. To help managers answering those questions, we conducted a study analyzing the impact of the fleet composition and the access to charging infrastructure in the feasibility and quality of TRP-CEV solutions. To conduct this study, we used the $84 (= 14 \times 6)$ instances described in Section 5.6.1. For each instance we ran our PMa 10 times and report average results. We analyze the data under the light of 4 metrics that are important to ENEDIS managers: the total cost (and how it is split between fixed and variable costs), the total CVs emissions [7], the number of visited CSs, and the number of routes.

Fleet composition analysis

We analyse in this section the behavior of the metrics when the proportion of EVs in the fleet changes. Figure 5.1 shows of each fleet composition: the average fixed, variable and total cost. The symbol N.F. indicates that in none of the 10 runs, our PMa could find a feasible solutions for the instance under the given fleet composition. The latter is the case for every rural instance when the fleet is entirely composed of EVs. Notice that a pathological case is observed on instance `rural_21`. A close look at the instance data reveals that the requests are particularly scattered over a large geographical region. In consequence, EVs need to charge their batteries quite often. Since charging the battery is a time-consuming process, the resulting routes are long (in terms of travel time) and difficult (if not impossible) to conciliate with the technicians' availability schedules.

The data on the fixed costs reported in Figure 5.1 shows an interesting and somehow unexpected behavior. It is worth reminding that in our problem the fixed costs are directly (and exclusively) related to the number of routes. Intuition says that on instances in which the fleet is composed mainly of EVs more routes would be needed to service requests. The reason is that routes covered with EVs tend to serve fewer requests than routes covered with CVs (because the EV spends a portion of their available time on charging operations). The data, however, shows that (at least on our instances) the fixed cost remain constant across the different fleet compositions.

[6] Notice that, because of Hiermann et al. (2016) just report the average solution, we could not compute the number of BKSs and the average best gap for the ALNS

[7] This values were calculated by multiplying the distance travelled by the CVs by a constant emission rate available in their technical information

A more intuitive result comes from the data on the variable costs. As Figure 5.1 shows, the variable costs decrease when the proportion of EVs in the fleet composition increases. This result is largely explained by the considerably higher cost per kilometre allocated to CVs with respect to EVs. It is worth noting that variable costs are higher on rural instances than in semi-urban and urban instances. The reason is simply that in rural operations vehicles travel larger distances. For instances, the average travelled distance per served request is 9.52 km on rural instances compared to 3.74 km and 1.94 km for the semi-urban and urban instances, respectively.

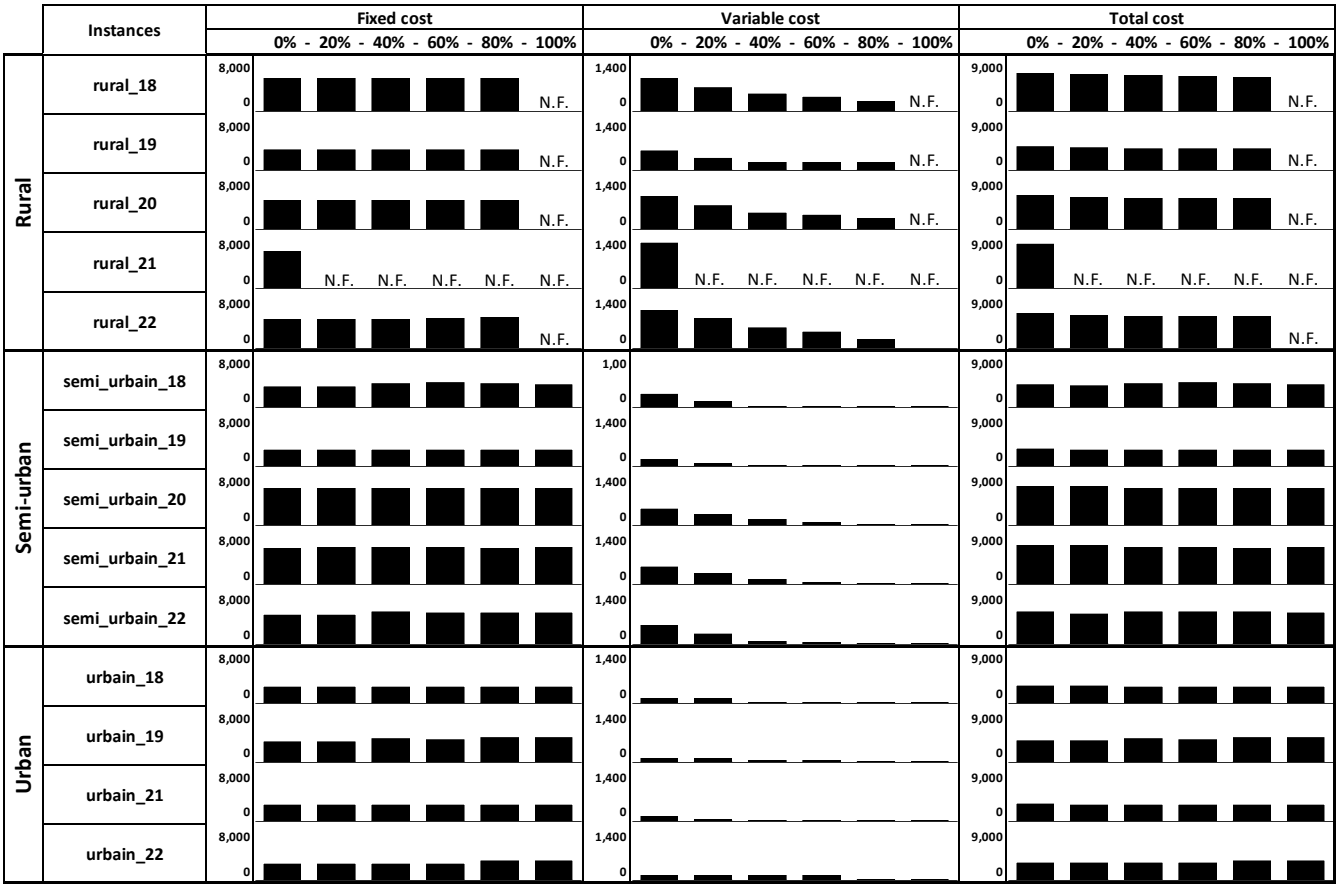
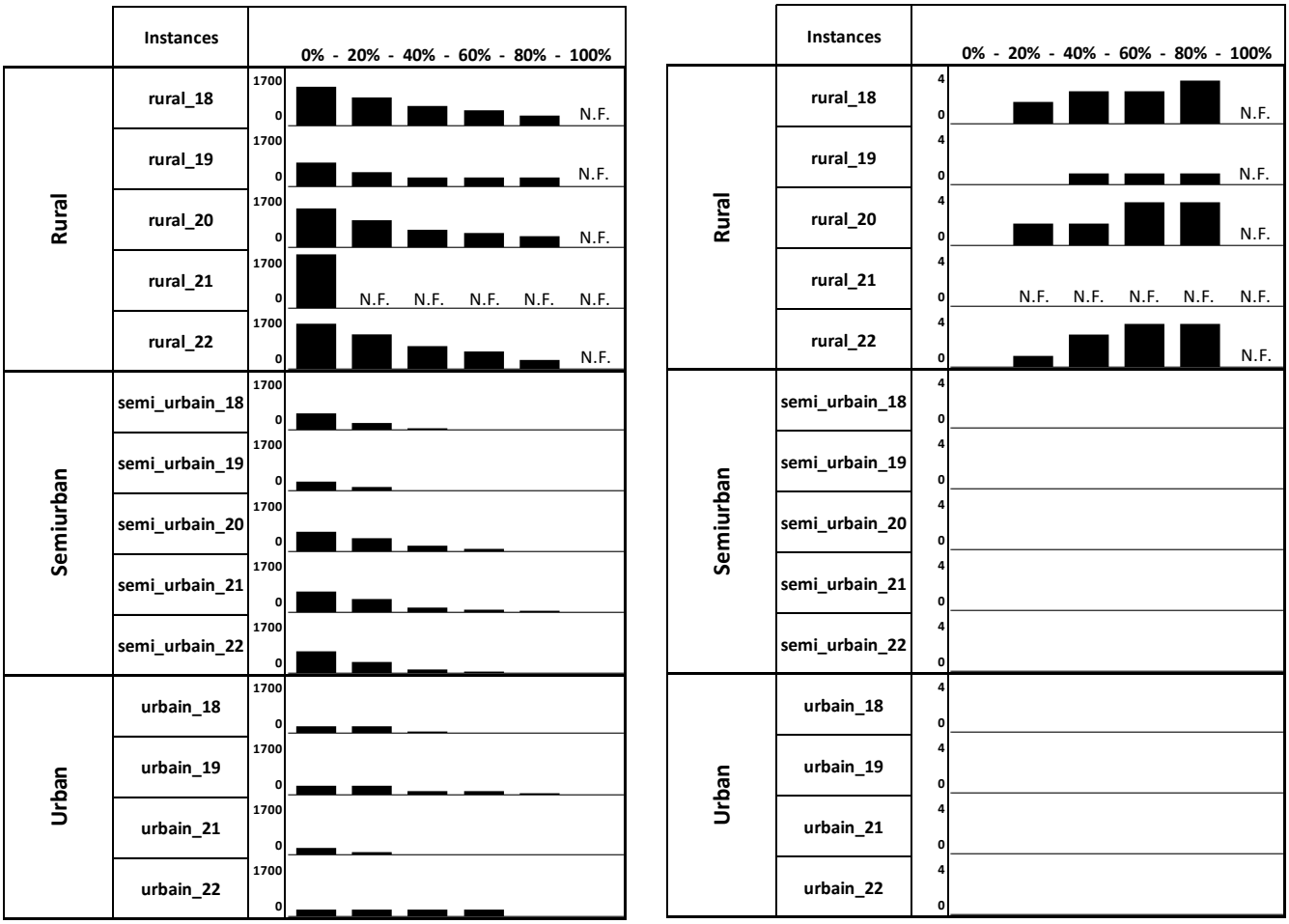


Figure 5.1 – Average fixed (i.e., the fixed cost of each technician), variable (i.e., the sum of the total travel cost, fixed charging cost and parking cost), and total cost for each instance for each fleet composition.

Figure 5.2a shows the Kg of CO_2 emitted (on average) by the CVs employed in the solutions to each instance for each fleet composition. As expected, the emissions decrease when the proportion of EVs in the fleet increases. Notice that rural instances represent the most significant potential for emission reductions because vehicles travel longer distances. Finally, Figure 5.2b shows the maximum number of visited CSs in a solution over the 10 runs of each instance for each fleet composition. Note that only the solutions of the rural instances include routes visiting CSs. This behavior is expected because requests are geographically dispersed in those instances. This result confirms eVRPs arise mainly in the rural and semi-urban operations.

Access to charging infrastructure

Up to today, ENEDIS does not own charging infrastructure outside their depots. Therefore, the mid-route recharges take place in public CSs. For a number of reasons ranging from difficult access to high charging costs, ENEDIS would prefer to avoid mid-route charging at public stations. To help them evaluate the impact of disallowing mid-route charging outside the depot on the quality and feasibility of TRP-CEV



(a) Average emission.

(b) Maximum number of visited CSs.

Figure 5.2 – Average emission (in Kg CO_2 per Km) and maximum number of visited CSs in a solution of each instance for each fleet composition.

solutions, we conducted a small study. We ran our PMa on the rural instances with 20% to 80%[8] of EVs in the fleet, considering that the EVs departs from the depot with the battery full. We compared these results to those obtained in the experiments reported in Section 5.6.4.

We were able to obtain feasible solutions for all the tested instances. However, our results suggest that disallowing mid-route charging has a impact on the quality of the solution. Figure 5.3 shows the average gap between the cost of the solutions with and without the option of visiting CSs [9].

The increment on the cost is explained by two factors: i) the solutions employed more vehicles, and ii) CVs have to travel more distance (which leads to higher travel costs). Figure 5.4a shows the comparison of the maximum number of routes over the 10 runs with and without the option of visit the CSs. Although in most of cases the number of vehicles is the same, for instances `rural_18` and `rural_22` the solutions employ an additional route when EVs are 80% of the fleet. This explains the large gap shown in Figure 5.3 for these instances. Figure 5.4b shows the gap in the average travelled distance of the CVs on the instances without mid-route charge with respect to the original solutions reported in Section 5.6.4. Since the EVs can not use the CSs, their autonomy is limited to the battery level when they leave the depot. Then, the CVs have to visit more request, and thus they have to travel more distance.

[8]Note that re-running the algorithm on instances with no EVs in the fleet would lead to the same solutions and that running the algorithm on instances with only EVs on the fleet would lead again to unfeasible solutions

[9]$G(\%) = (cost_{withoutCSs} - cost_{withCSs})/cost_{withCSs} \times 100$

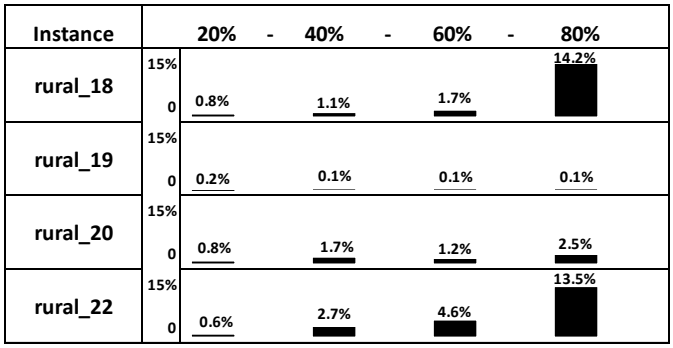
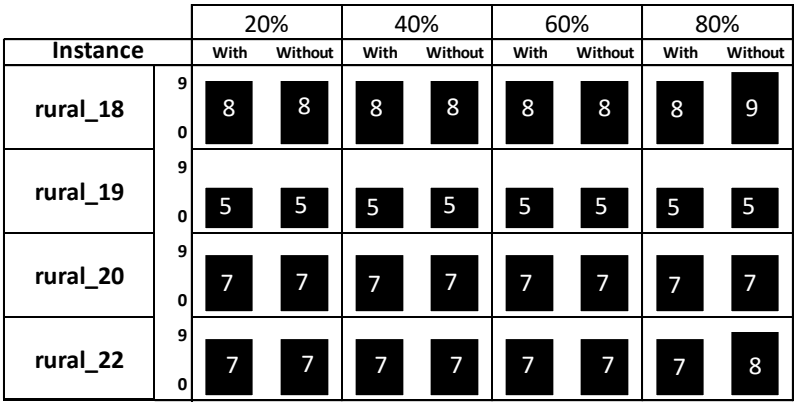
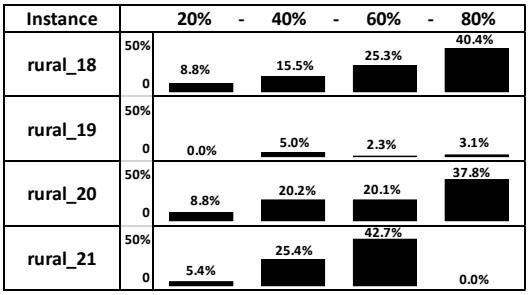


Figure 5.3 – Average gap between the cost of the solutions with and without the option of visiting the CSs



(a) Comparison of the maximum number of routes with/without the option of visiting CSs.



(b) Gap in the average travelled distance of the CVs on the instances without mid-route charge with respect to the original solutions reported in Section 5.6.4.

Figure 5.4 – Factors explaining the increment on the cost when the visits to the CSs are forbidden

5.7 Conclusion

In this chapter, we introduce what we call the technician routing problem with conventional and electric vehicles (TRP-CEV). The TRP-CEV is largely inspired by a problem faced by ENEDIS in their daily operations. This problem consists in routing a set of technicians to serve a set of requests, using a fleet composed by EVs and CVs of different types. This problem considers several real-life constraints such as: time windows on the requests; technician skills, working schedules and lunch breaks; skills incompatibility

between technicians and requests; incompatibility between CSs and EVS; nonlinear battery charging functions; and a limited number of vehicles for each type. Decisions that must be taken simultaneously are the vehicle-to-technician assignment, the sequencing of the routes, and the battery charging program.

To solve the TRP-CEV we proposed a parallel matheuristic (PMa). This approach decomposes the TRP-CEV into a set of VRP-TW and eVRP-TW with lunch break (one for each couple technician-vehicle), and solves each sub-problem in parallel using a greedy randomized adaptive search procedure (GRASP). Later, PMa uses a set covering model to assemble a TRP-CEV solution. We built a set of real instances based on the ENEDIS operation. The aim of these instances is to evaluate the performance of the PMa, and to analyse the features of the solutions under the light of some important metrics for ENEDIS. The proposed PMa provides competitive results when compared against the current routing software used by the company and state-of-the-art methods for a related problem (the EFSMTW). Finally, we analyzed the solutions delivered by PMa for the TRP-CEV aiming to provide ENEDIS team with some insight about the behaviour of costs, emissions, and visits to CSs for different compositions of the fleet. Our analysis concluded that the cost decreases in function of the number of EVs in the fleet, and that the visits to the CSs are relevant only for the rural instances.

5.8 Nomenclature

5.8.1 Notation for the problem description

G: TRP-CEV underlying graph $G = (N, A)$.
N: Set of vertices of G.
0: Depot.
C: Subset of N representing the requests.
F: Subset of N representing the CSs.
S: Set of skills.
T: Set of technicians.
K: Set of types of vehicles.
K_e: Set of types of electric vehicles.
K_c: Set of types of conventional vehicles.
A: Set of arcs in G.
B: Set of battery levels in percentage.
p_i: Service time at request $i \in C$.
pc_i: Parking cost at CS $i \in F$.
$[ec_i, lc_i]$: Time window at request $i \in C$.
f_u: Fixed cost of technician $u \in T$.
s_u: Subset of skills of technician $u \in T$.
$[st_u, et_u]$: Working schedule of technician $u \in T$.
$[sl_u, el_u]$: Lunch break of technician $u \in T$.
fc_u: Consumption factor of the technician $u \in T$.
m_k: Number of vehicles of type $k \in K$ in the fleet.
tc_k: Unitary travel cost of type of vehicle $k \in K$.
g_k: Fixed cost of recharging the battery of vehicles type $k \in K_e$
Q_k: Battery capacity of vehicles of type $k \in K$.
F_k: Subset of CS, where of vehicles of type $k \in K$ can charge.
a_{ikl}: Charging time of a vehicle of type $k \in K_e$ charging from zero to $l \in B$ percent of the battery at CS $i \in F$.
d_{ij}: Distance between vertices i and j ($i, j \in N$).
t_{ij}: Travel time between vertices i and j ($i, j \in N$).
e_{ijk}: Energy consumption between vertices i and j ($i, j \in V$) of the type of vehicle $k \in K_e$.

5.8.2 Notation for the MILP formulation

F': Set containing the set F and the copies of CSs.
N': Set of vertex containing the set F'.
V: Set of vehicles.
V_e: Set of electric vehicles.
V_c: Set of conventional vehicles.
n_{il}: Binary parameter, equal to 1 if request $i \in C$ requires a technician with skill $l \in S$, and 0 otherwise.
m_{ul}: Binary parameter, equal to 1 if technician $u \in T$ has skill $l \in S$, and 0 otherwise.
h_{ik}: Binary parameter, equal to 1 if the EV $k \in V_e$ can charge at CS $i \in F'$.
S_{max}: Maximum charging time of any charging function.
x_{ijku}: Binary variable, equal to 1 if technician $u \in T$ travels from vertex i to j in vehicle $k \in V$, and 0 otherwise.
τ_{iu}: Arrival time of technician $u \in T$ at vertex $i \in N'$.
y_{ik}: Battery level of EV $k \in V_e$ upon departure from vertex $i \in N'$.
q_{ik}: Battery level when EV $k \in V_e$ arrives at CS $i \in F'$.
o_{ik}: Battery level when EV $k \in V_e$ departs from CS $i \in F'$.
α_{ikl}: Binary variable, equal to 1 if the battery level is $l \in B$ when vehicle $k \in V_e$ arrives at CS $i \in F'$.
β_{ikl}: Binary variable, equal to 1 if the battery level is $l \in B$ when vehicle $k \in V_e$ departs from CS $i \in F'$.
Δ_{ik}: Time spent by vehicle $k \in V_e$ at CS $i \in F'$.
w_{iju}: Binary variable, equal to 1 if technician $u \in T$ takes the lunch break between the vertex $i \in N'$ and $j \in N'$ and 0 otherwise.
z_{iju}: Binary variable, equal to 1 if the technician $u \in T$ takes the lunch break after travelling from vertex i to $j \in N'$ and 0 otherwise.
z'_{iju}: Binary variable, equal to 1 if the technician $u \in T$ takes the lunch break before travelling from vertex i to $j \in N'$, and 0 otherwise.

5.8.3 Notation for the PMa

E: Set of technician types.
P: Set of couples type of vehicle - type of technician.
C_p: Set of feasible request for couple $p \in P$.
Ω_p: Pool of routes found for couple $p \in P$.
Ω: Global pool of routes.
σ: TRP-CEV solution.
I: Number of iterations of GRASP.
κ: The list size of the best insertion of the randomized Solomon heuristic.
c_r: Cost of the route $r \in \Omega$.
b_{ri}: Binary parameter, equal to 1 if route $r \in \Omega$ serves request $i \in C$ and 0 otherwise.
pv_{rk}: Binary parameter, equal to 1 if route $r \in \Omega$ is performed by a vehicle of type $k \in K$ and 0 otherwise.
h_{ru}: Binary parameter, equal to 1 if route $r \in \Omega$ is performed by a technician of type $u \in E$ and 0 otherwise.
n_u: Number of technicians of type $u \in E$.
χ_r: Binary variable, equal to 1 if route $r \in \Omega$ is selected and 0 otherwise.

5.8.4 Notation for the FRVCP-TW

Π: Route that violates the energy constraint.
$\overline{\Pi}$: Route that includes the dummy vertex associated to the lunch break.
$\varepsilon_{\pi(i)j}$: Binary variable, equal to 1 if the vehicle charges at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
If $\epsilon_{\pi(i)j} = 1$, $\phi_{\pi(i)}$ is the battery level when the vehicle arrives at CS $j \in F$ right before visiting vertex $\pi(i)$.
$\phi_{\pi(i)}$: Battery level if $\varepsilon_{\pi(i)j} = 0$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at vertex $\pi(i)$.
$\alpha_{\pi(i)jl}$: Binary variable, equal to 1 if the battery level is $l \in B$ when the vehicle arrives at $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
$\beta_{\pi(i)jl}$: Binary variable, equal to 1 if the battery level is $l \in B$ when the vehicle arrives at and departs from CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
$\gamma_{\pi(i)j}$: Energy charged at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
$q_{\pi(i)j}$: Battery level when the EV arrives CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
$o_{\pi(i)j}$: battery level when the EV departs from CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$
$\delta_{\pi(i)j}$: Spent time at CS $j \in F$ before visiting vertex $\pi(i) \in \overline{\Pi}$.
$D_{\pi(i)}$: Earliest departure time at vertex $\pi(i) \in \overline{\Pi}$.
$L_{\pi(i)}$: Latest feasible arrival time at vertex $\pi(i) \in \overline{\Pi}$.
$Y_{\pi(i)}$: Battery level at vertex $\pi(i) \in \overline{\Pi}$.
c: Total cost of the route.
sn: Sum of the battery levels with negative values.

Chapter 6

General conclusions & perspectives

This final chapter summarizes the contributions of the thesis and offers future research directions.

This thesis is devoted to study the electric vehicle routing problems (eVRPs). These problems extend classical VRPs to consider the main features of electric vehicles (EVs). Despite some researchers have started to study these kind of problems, the gap with classical models and, most importantly, real-world applications is still significant. Therefore, the purpose of this thesis was to contribute to closing that gap.

In Chapter 2, we proposed a simple yet effective two-phase heuristic to tackle the Green VRP. In the first phase our heuristic builds a pool of routes via a set of randomized route-first cluster-second heuristics with an optimal station insertion procedure. In the second phase our approach assembles a Green VRP solution by solving a set partitioning formulation over the columns (routes) stored in the pool. To test our approach, we performed experiments on a set of 52 instances from the literature. The results showed that our heuristic is competitive with state-of-the-art methods. Our heuristic unveiled 8 new best-known solutions, matched another 40, and delivered solutions with an average gap of 0.14% for the 4 remaining instances.

The Green VRP and most of the existing eVRP studied in the literature rely on one (or both) of the following assumptions: (i) the vehicles fully charge their batteries every time they reach a charging station (CS), and (ii) the battery charge level is a linear function of the charging time. In practical situations, however, the amount of charge is a decision variable, and the battery charge level is a concave function of the charging time. In Chapter 3, we extended current eVRP models to consider partial charging and nonlinear charging functions. We presented a computational study comparing our assumptions with those commonly made in the literature. Our results suggested that neglecting partial and nonlinear charging may lead to infeasible or overly expensive solutions

To tackle large-scale instances of the eVRP with nonlinear charging function (eVRP-NL), in Chapter 4 we proposed an iterated local search (ILS) enhanced with heuristic concentration (HC). At each iteration, our ILS applies a variable neighborhood descent (VND) procedure with three different local search operators and stores the routes of the local optimum in a pool of routes. When the ILS finishes, the HC component assembles the final solution by recombining routes from the pool. One original feature of our approach is the use in the VND component of a local search operator that aims at improving the charging decisions of each route (where and how much to charge). To achieve this goal, the operator solves a new optimization problem: the fixed-route vehicle-charging problem (FRVCP). Our implementation solves the FRVCP using a heuristic and a commercial solver. To test our approach, we proposed a 120-instance testbed for the e-VRP-PNL . We discussed experiments assessing the performance of the proposed approach and showing the relevance of making optimal charging decisions.

Finally, in Chapter 5, we introduced the technician routing problem with conventional and electric vehicles (TRP-CEV), which is inspired in the real operation of ENEDIS . This problem consists in routing a set of technicians to serve a set of geographically scattered requests within their time windows. This problem considers the working schedules, the lunch breaks, and skills of the technicians. The technicians use a fixed fleet composed of conventional and electric vehicles. The EVs have a limited autonomy, and there are incompatibility constraints between EVs and CSs. In the TRP-CEV decisions include the vehicle-

to-technician assignment, the sequencing of the routes followed by the technicians , and the battery charging program for the EVs (where and how much to charge) considering that the charging functions are nonlinear. The objective function of TRP-CEV seeks to minimize the total cost, defined as the sum of the travel costs, battery charging fixed costs, the parking cost at CSs, and the fixed cost of using each technician. To tackle the TRP-CEV, we proposed a two-phase parallel matheuristic (PMa). In the first phase, the PMa builds a pool of feasible TRP-CEV routes solving in parallel a set of sub-problems. In the second phase the PMa assembles a TRP-CEV solution by solving an extended set covering formulation. To test our approach, we ran it on different benchmarks. The experiments showed that our PMa has competitive results. Furthermore, we applied (tested) the approach on a set of real instances based on the ENEDIS operation. We analysed the obtained solutions by our PMa on those instances, in order to evaluate the quality and structure of the solution as a function of the percentage of EVs in the fleet.

As a general conclusion, in this thesis we have highlighted the importance to make optimal charging decision in the eVRPs. We have found that the optimal charging decisions are essential to have competitive results in terms of solution quality. Therefore, for all the eVRPs addressed in this thesis, we proposed different approaches for making optimal charging decisions. On the other hand, in a large part of this thesis, we included realistic assumptions about the battery charging process. Those assumptions are the partial charging and the non-linear charging function. The incorporation of those assumptions into the eVRPs and the proposed approaches to tackle these assumptions allow to model more realistic eVRP problems. Other notable conclusion for the context of the eVRP is that the more complicated eVRPs (i.e., where there are charging decisions) are in rural zones. In different experiments conducted during this thesis, we found that only in rural areas there are day-long routes, where the travelled distance is larger than the EV autonomy.

To further our research we plan to design an exact method for the TRP-CEV to have a better base of comparison for our results. Recently, researchers have proposed exact methods for problems directly related to the TRP-CEV. For instance, a branch-and-price method for the technician dispatching problem (Cortés et al. 2014), a branch-price-and-cut method for the eVRPs with time windows (Desaulniers et al. 2014) and a branch-and-price method for a VRP with lunch break (Liu et al. 2016). Taking advantage of recent studies, the idea behind this perspective is to integrate and extend those last advances to propose an exact method for the TRP-CEV.

Despite this thesis contributes to bringing the eVRP closer to reality in terms of the battery charging assumptions, there is a pending research about other assumptions of the eVRPs. We suggest that further research should be undertaken in the following assumptions: the capacity of the CSs, and the uncertain CS availability and energy consumption. In terms of the methods, an interesting perspective for the eVRPs context is to explore alternative approach for the fixed-route charging-problems that ensures the optimal solution with computing time faster. Despite we tried -without success- to find a polynomial time algorithm, we still think that it is possible to find a better approach than the proposed in this thesis. Considering that in the road network the consumption of energy, the travelled distance, and the travel time are not proportional, another interesting perspective is to model the eVRP as a multi-graph. The idea behind this perspective is that existing more than one edge between two nodes, where each edge represents attributes as the energy consumption, the travel time, or the travelled distance.

Appendix A

Appendices

A.1 Detailed results for Green VRP instances

Tables A.1 and A.2 show the results of our three MSH configurations on the small and large Green VRP instances. We compare our results to the best results obtained by the MCWS and DBCA heuristics of Erdoğan & Miller-Hooks (2012), the VNS/TS of Schneider et al. (2014), the AVNS of Schneider et al. (2015), and the 48A and SA of Felipe et al. (2014); they did not report result of the SA for the small instances. For each instance, we report the problem name and the BKS taken from Erdoğan & Miller-Hooks (2012), Schneider et al. (2014), Schneider et al. (2015), or Felipe et al. (2014) and updated with some new BKSs found by our MSH.

Erdoğan & Miller-Hooks (2012) reported the best distance (Best), and the number of vehicles of the best found solution (v) of multiple runs with different parameters for their MCWS and DBCA; they did not give the exact number of runs. Schneider et al. (2014) reported the best distance, the number of vehicles of the best found solution, and the average computing time (t in minutes) over ten runs of their VNS/TS, Schneider et al. (2015) reported the best distance, the average distance (Avg.), and the average computing time over ten runs of their AVNS. Finally, Felipe et al. (2014) reported the best distance, the number of vehicles of the best found solution, and the computing time over a single run of their 48A and SA. For all the algorithms, we provide the number of customers served (n). The last rows of the table summarize the average BKS gap, the cumulative number of vehicles, the number of times each method found the BKS, and the average running time. Values in bold indicate that a method found the BKS.

Table A.1 – Results of MSH on small instances of Erdoğan & Miller-Hooks (2012).

		MCWS/DBCA			VNS/TS				AVNS				48A				MSH(1k)					MSH(5k)					MSH(10k)				
Instance	**BKS**	**n**	**v**	**Best**	**n**	**v**	**Best**	**t**	**n**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**t**	**n**	**v**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**Avg.**	**t**
20c3sU1	1797.49	20	6	1797.51	20	6	**1797.49**	0.69	20	**1797.49**	1797.49	0.16	20	6	1805.41	0.03	20	6	**1797.49**	1797.88	0.01	20	6	**1797.49**	1797.49	0.04	20	6	**1797.49**	1797.49	0.08
20c3sU2	1574.77	20	6	1613.53	20	6	**1574.77**	0.64	20	**1574.78**	1574.78	0.15	20	6	**1574.78**	0.02	20	6	**1574.78**	1574.78	0.01	20	6	**1574.78**	1574.78	0.04	20	6	**1574.78**	1574.78	0.07
20c3sU3	1704.48	20	7	1964.57	20	6	**1704.48**	0.64	20	**1704.48**	1704.48	0.13	20	6	**1704.48**	0.02	20	6	**1704.48**	1706.08	0.01	20	6	**1704.48**	1704.48	0.04	20	6	**1704.48**	1704.48	0.07
20c3sU4	1482.00	20	6	1487.15	20	5	**1482.00**	0.65	20	**1482.00**	1482.00	0.17	20	5	**1482.00**	0.03	20	5	**1482.00**	1482.00	0.01	20	5	**1482.00**	1482.00	0.04	20	5	**1482.00**	1482.00	0.07
20c3sU5	1689.37	20	5	1752.73	20	6	**1689.37**	0.67	20	**1689.37**	1689.37	0.18	20	6	**1689.37**	0.04	20	6	**1689.37**	1689.37	0.01	20	6	**1689.37**	1689.37	0.04	20	6	**1689.37**	1689.37	0.07
20c3sU6	1618.65	20	6	1668.16	20	6	**1618.65**	0.67	20	**1618.65**	1618.65	0.15	20	6	**1618.65**	0.03	20	6	**1618.65**	1618.65	0.01	20	6	**1618.65**	1618.65	0.04	20	6	**1618.65**	1618.65	0.07
20c3sU7	1713.66	20	6	1730.45	20	6	**1713.66**	0.64	20	**1713.66**	1713.66	0.19	20	6	**1713.67**	0.03	20	6	**1713.66**	1714.95	0.01	20	6	**1713.66**	1714.03	0.04	20	6	**1713.66**	1713.87	0.07
20c3sU8	1706.50	20	6	1718.67	20	6	**1706.50**	0.67	20	**1706.50**	1706.50	0.16	20	6	1722.78	0.03	20	6	**1706.50**	1706.50	0.01	20	6	**1706.50**	1706.50	0.04	20	6	**1706.50**	1706.50	0.07
20c3sU9	1708.81	20	6	1714.43	20	6	**1708.81**	0.66	20	**1708.82**	1708.82	0.19	20	6	**1708.82**	0.04	20	6	1710.90	1711.14	0.01	20	6	**1708.82**	1710.09	0.04	20	6	**1708.82**	1709.65	0.07
20c3sU10	1181.31	20	5	1309.52	20	4	**1181.31**	0.64	20	**1181.31**	1181.31	0.23	20	4	**1181.31**	0.02	20	4	**1181.31**	1181.31	0.01	20	4	**1181.31**	1181.31	0.04	20	4	**1181.31**	1181.31	0.07
20c3sC1	1173.57	20	5	1300.62	20	4	**1173.57**	0.62	20	**1173.57**	1173.57	0.38	20	4	1178.97	0.03	20	4	**1173.57**	1173.57	0.01	20	4	**1173.57**	1173.57	0.04	20	4	**1173.57**	1173.57	0.07
20c3sC2	1539.97	19	5	1553.53	19	5	**1539.97**	0.58	19	**1539.97**	1539.97	0.21	19	5	**1539.97**	0.02	19	5	**1539.97**	1539.97	0.01	19	5	**1539.97**	1539.97	0.04	19	5	**1539.97**	1539.97	0.08
20c3sC3	880.20	12	4	1083.12	12	3	**880.20**	0.25	12	**880.20**	880.20	0.15	12	3	**880.20**	0.01	12	3	**880.20**	880.20	0.01	12	3	**880.20**	880.20	0.03	12	3	**880.20**	880.20	0.04
20c3sC4	1059.35	18	5	1091.78	18	4	**1059.35**	0.53	18	**1059.35**	1077.71	0.23	18	4	**1059.35**	0.02	18	4	**1059.35**	1088.54	0.01	18	4	**1059.35**	1059.94	0.04	18	4	**1059.35**	1059.94	0.06
20c3sC5	2156.01	19	7	2190.68	19	7	**2156.01**	0.60	19	**2156.01**	2156.01	0.14	19	7	**2156.01**	0.02	19	7	**2156.01**	2157.31	0.01	19	7	**2156.01**	2156.56	0.05	19	7	**2156.01**	2156.04	0.10
20c3sC6	2758.17	17	9	2883.71	17	8	**2758.17**	0.71	17	**2758.17**	2758.17	0.14	17	8	**2758.17**	0.02	17	8	**2758.17**	2758.17	0.01	17	8	**2758.17**	2758.17	0.04	17	8	**2758.17**	2758.17	0.08
20c3sC7	1393.99	6	5	1701.40	6	4	**1393.99**	0.18	6	**1393.99**	1393.99	0.04	6	4	**1393.99**	0.00	6	4	**1393.99**	1393.99	0.01	6	4	**1393.99**	1393.99	0.03	6	4	**1393.99**	1393.99	0.06
20c3sC8	3139.72	18	10	3319.74	18	9	**3139.72**	0.62	18	**3139.72**	3139.72	0.08	18	9	**3139.72**	0.02	18	9	**3139.72**	3139.72	0.01	18	9	**3139.72**	3139.72	0.05	18	9	**3139.72**	3139.72	0.12
20c3sC9	1799.94	19	6	1811.05	19	6	**1799.94**	0.60	19	**1799.94**	1799.94	0.16	19	6	**1799.94**	0.02	19	6	**1799.94**	1799.94	0.01	19	6	**1799.94**	1799.94	0.05	19	6	**1799.94**	1799.94	0.10
20c3sC10	2583.42	15	8	2644.11	15	8	**2583.42**	0.45	15	**2583.42**	2600.39	0.09	15	8	**2583.42**	0.02	15	8	**2583.42**	2583.42	0.01	15	8	**2583.42**	2583.42	0.04	15	8	**2583.42**	2583.42	0.07
S1_2i6s	1578.12	20	6	1614.15	20	6	**1578.12**	0.71	20	**1578.12**	1578.12	0.16	20	6	**1578.12**	0.03	20	6	**1578.12**	1578.12	0.01	20	6	**1578.12**	1578.12	0.04	20	6	**1578.12**	1578.12	0.07
S1_4i6s	1397.27	20	5	1541.46	20	5	**1397.27**	0.75	20	**1397.27**	1397.27	0.16	20	5	1413.97	0.03	20	5	**1397.27**	1397.27	0.01	20	5	**1397.27**	1397.27	0.04	20	5	**1397.27**	1397.27	0.07
S1_6i6s	1560.49	20	6	1616.20	20	5	**1560.49**	0.73	20	**1560.49**	1560.49	0.20	20	6	1571.30	0.03	20	5	**1560.49**	1563.95	0.01	20	5	**1560.49**	1561.19	0.04	20	5	**1560.49**	1560.49	0.07
S1_8i6s	1692.32	20	6	1882.54	20	6	**1692.32**	0.74	20	**1692.32**	1692.32	0.17	20	6	**1692.33**	0.03	20	6	**1692.32**	1692.32	0.01	20	6	**1692.32**	1692.32	0.04	20	6	**1692.32**	1692.32	0.07
S1_10i6s	1173.48	20	5	1309.52	20	4	**1173.48**	0.71	20	**1173.48**	1173.48	0.24	20	4	**1173.48**	0.03	20	4	**1173.48**	1173.48	0.01	20	4	**1173.48**	1173.48	0.04	20	4	**1173.48**	1173.48	0.07
S2_2i6s	1633.10	20	6	1645.80	20	6	**1633.10**	0.75	20	**1633.10**	1633.10	0.19	20	6	1645.80	0.03	20	6	**1633.10**	1633.10	0.01	20	6	**1633.10**	1633.10	0.05	20	6	**1633.10**	1633.10	0.09
S2_4i6s	1505.07	19	6	1505.07	19	5	1532.96	0.88	19	**1505.07**	1505.07	0.14	19	6	**1505.07**	0.02	19	6	**1505.07**	1505.07	0.02	19	6	**1505.07**	1505.07	0.05	19	6	**1505.07**	1505.07	0.09
S2_6i6s	2431.33	20	10	3115.10	20	7	**2431.33**	0.78	20	**2431.33**	2431.33	0.13	20	8	2660.49	0.04	20	7	**2431.33**	2439.92	0.02	20	7	**2431.33**	2431.33	0.04	20	7	**2431.33**	2431.33	0.07
S2_8i6s	2158.35	16	9	2722.55	16	7	**2158.35**	0.57	16	**2158.35**	2158.35	0.09	16	7	2175.66	0.02	16	7	**2158.35**	2158.35	0.01	16	7	**2158.35**	2158.35	0.03	16	7	**2158.35**	2158.35	0.06
S2_10i6s	1585.46	16	6	1995.62	17	6	1958.46	0.61	16	**1585.46**	1585.46	0.15	16	5	**1585.46**	0.02	16	5	**1585.46**	1585.46	0.01	16	5	**1585.46**	1585.46	0.04	16	5	**1585.46**	1585.46	0.06
S1_4i2s	1582.20	20	6	1582.20	20	6	**1582.21**	0.63	20	**1582.21**	1582.21	0.13	20	6	1598.91	0.03	20	6	**1582.21**	1582.21	0.01	20	6	**1582.21**	1582.21	0.04	20	6	**1582.21**	1582.21	0.07
S1_4i4s	1460.09	20	6	1580.52	20	5	**1460.09**	0.68	20	**1460.09**	1460.09	0.16	20	5	1483.19	0.03	20	5	**1460.09**	1460.09	0.01	20	5	**1460.09**	1460.09	0.04	20	5	**1460.09**	1460.09	0.07
S1_4i6s	1397.27	20	5	1541.46	20	5	**1397.27**	0.75	20	**1397.27**	1397.27	0.16	20	5	1413.97	0.03	20	5	**1397.27**	1397.27	0.01	20	5	**1397.27**	1397.27	0.04	20	5	**1397.27**	1397.27	0.07
S1_4i8s	1397.27	20	6	1561.29	20	6	**1397.27**	0.82	20	**1397.27**	1397.27	0.17	20	6	**1397.27**	0.03	20	5	**1397.27**	1397.27	0.02	20	5	**1397.27**	1397.27	0.05	20	5	**1397.27**	1397.27	0.07
S1_4i10s	1396.02	20	5	1529.73	20	5	**1396.02**	0.85	20	**1396.02**	1396.02	0.23	20	5	**1396.02**	0.03	20	5	**1396.02**	1396.02	0.02	20	5	**1396.02**	1396.02	0.05	20	5	**1396.02**	1396.02	0.07
S2_4i2s	1059.35	18	5	1117.32	18	4	**1059.35**	0.51	18	**1059.35**	1069.42	0.23	18	4	**1059.35**	0.02	18	4	**1059.35**	1097.64	0.01	18	4	**1059.35**	1059.94	0.04	18	4	**1059.35**	1059.94	0.06
S2_4i4s	1446.08	19	6	1522.72	19	5	**1446.08**	0.60	19	**1446.08**	1449.17	0.21	19	5	**1446.08**	0.02	19	5	1476.91	1476.91	0.02	19	5	**1446.08**	1452.84	0.05	19	5	**1446.08**	1446.08	0.09
S2_4i6s	1434.14	20	6	1730.47	20	5	**1434.14**	0.69	20	**1434.14**	1445.35	0.20	20	5	**1434.14**	0.02	20	5	1440.10	1462.48	0.02	20	5	**1434.14**	1438.43	0.05	20	5	**1434.14**	1435.95	0.08
S2_4i8s	1434.14	20	6	1786.21	20	5	**1434.14**	0.75	20	**1434.14**	1434.14	0.20	20	5	**1434.14**	0.02	20	5	1440.10	1462.48	0.02	20	5	**1434.14**	1439.03	0.05	20	5	**1434.14**	1435.95	0.08
S2_4i10s	1434.13	20	6	1729.51	20	5	**1434.13**	0.78	20	**1434.13**	1455.31	0.24	20	5	**1434.13**	0.02	20	5	1440.09	1462.48	0.02	20	5	**1434.13**	1435.94	0.06	20	5	**1434.13**	1435.94	0.09
Avg. Gap above BKS				8.72%			0.63%			0.00%	0.15%				0.46%				0.09%	0.38%				0.00%	0.04%				0.00%	0.01%	
NBKS				2			38			40					29				35					40					40		
Cum. Number of Veh.			245			223								225				222					222				222			
Avg. Time (min)								0.65				0.17				0.02					0.01					0.04				0.07

Table A.2 – Results of MSH on large instances of Erdoğan & Miller-Hooks (2012).

		MCWS/DBCA			VNS/TS				AVNS				48A				SA			
Instance	**BKS**	**n**	**v**	**Best**	**n**	**v**	**Best**	**t**	**n**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**t**	**n**	**v**	**Best**	**t**
111c_21s	4770.47	109	20	5626.64	109	17	4797.15	21.76	109	**4770.47**	4791.53	1.78	109	18	4960.60	21.74	109	18	5062.06	12.35
111c_22s	4774.65	109	20	5610.57	109	17	4802.16	23.56	109	4776.81	4797.31	1.94	109	18	4914.20	21.23	109	18	5029.17	12.30
111c_24s	4767.14	109	20	5412.48	109	17	4786.96	21.90	109	**4767.14**	4790.84	2.16	109	18	4952.90	21.27	109	18	5010.59	12.13
111c_26s	4767.14	109	20	5408.38	109	17	4778.62	25.12	109	**4767.14**	4782.60	2.04	109	18	4934.11	21.25	109	18	5092.06	12.07
111c_28s	4765.52	109	20	5331.93	109	17	4799.15	24.17	109	**4765.52**	4781.26	1.73	109	18	4971.93	21.29	109	18	5038.84	11.99
200c_21s	8839.62	190	35	10413.59	192	35	8963.46	76.65	192	8886.00	8970.14	3.61	192	32	9276.63	76.41	192	32	9206.28	73.84
250c_21s	10482.52	235	41	11886.61	237	39	10800.18	120.90	237	10487.15	10531.20	3.67	237	39	11007.98	120.67	237	38	10885.71	108.10
300c_21s	12367.60	281	49	14229.92	283	46	12594.77	182.23	283	12374.49	12514.78	4.94	283	46	12869.17	184.27	283	45	12827.35	302.04
350c_21s	14073.34	329	57	16460.30	329	51	14323.02	232.03	329	14103.66	14271.56	7.11	329	54	14954.83	227.30	329	52	14828.63	332.38
400c_21s	16660.20	378	67	19099.04	378	61	16850.21	305.12	378	16697.21	16839.23	12.70	378	61	17351.92	302.03	378	60	17327.27	384.68
450c_21s	18241.48	424	75	21854.17	424	68	18521.23	525.52	424	18310.60	18512.47	13.19	424	68	19215.38	514.68	424	67	19085.91	456.26
500c_21s	20496.50	471	84	24517.08	471	76	21170.90	356.01	471	20609.67	20874.50	19.51	471	76	21636.59	352.16	471	75	21475.71	154.45
Avg. Gap above BKS				15.97%			1.38%			0.17%	0.92%				4.50%				4.97%	
NBKS				0			0			4					0				0	
Cum. Number of Veh.			508			461								466				459		
Avg. Time (min)								159.58				6.20				157.03				156.05

	MSH(1k)					MSH(5k)					MSH(10k)				
Instance	**n**	**v**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**Avg.**	**t**	**n**	**v**	**Best**	**Avg.**	**t**
111c_21s	109	17	4798.52	4861.35	0.91	109	17	4780.01	4793.07	2.40	109	17	4777.91	4781.85	4.94
111c_22s	109	17	4795.26	4858.09	1.04	109	17	4776.75	4789.81	2.13	109	17	**4774.65**	4778.80	4.69
111c_24s	109	17	4795.19	4855.89	1.16	109	17	4776.68	4789.77	2.74	109	17	4773.67	4778.62	5.64
111c_26s	109	17	4795.19	4855.89	0.79	109	17	4776.68	4789.77	2.56	109	17	4773.67	4778.62	5.23
111c_28s	109	17	4793.57	4854.86	0.93	109	17	4775.06	4788.22	2.80	109	17	4772.46	4777.03	5.54
200c_21s	192	32	9005.58	9039.97	2.96	192	31	8894.56	8923.18	19.80	192	31	**8839.62**	8879.98	19.96
250c_21s	237	39	10702.76	10755.30	13.43	237	38	10534.52	10579.12	27.59	237	37	**10482.52**	10518.32	21.58
300c_21s	283	45	12663.49	12719.25	44.28	283	44	12444.48	12548.70	37.43	283	44	**12367.60**	12421.75	47.53
350c_21s	329	52	14431.27	14470.40	53.01	329	50	14146.67	14253.55	50.96	329	50	**14073.34**	14226.03	63.01
400c_21s	378	60	16873.51	17143.75	64.29	378	59	16745.24	17128.76	67.91	378	59	**16660.20**	17119.89	71.70
450c_21s	424	66	18569.58	18831.00	72.11	424	65	18351.72	19009.22	76.35	424	65	**18241.48**	18902.03	80.75
500c_21s	471	75	20960.18	21678.33	80.11	471	73	20610.36	21297.07	84.95	471	73	**20496.50**	20997.04	89.95
Avg. Gap above BKS			1.42%	2.64%				0.40%	1.48%				0.05%	1.02%	
NBKS			0					0					8		
Cum. Number of Veh.	454						445					444			
Avg. Time (min)					27.92					31.47					35.04

A.2 Detailed results of the eVRP-NL

Tables A.3 and A.4 show the results of our two ILS versions (i.e, ILS(S)+HC and MILP(H)+HC) for the small and large eVRP-NL instances. In Table A.3, we compare our results with the optimal solutions found by Gurobi using the MILP formulation. In Table A.4, we compare our results with the best results obtained with Gurobi. For each instance, we report the problem name[1], and the best solution (BKS) taken from the results of Gurobi, ILS(H)+HC or ILS(S)+HC.

For the results obtained with Gurobi, we report the best solution (Best), and the gap with respect to the BKS (G) [2]. For the results obtained with the ILS(H)+HC and ILS(S)+HC, we report the best solution, the average solution (Avg.), and the average computing time (t in seconds) over ten runs. For the ILS(S)+HC, we also report the average computing time using the preprocessing procedure (t^*). For the two ILS+HC versions, we provide the gap of the average solution and best solution with reference to the BKS. The last rows of the table summarize the average and maximum BKS gap, the number of times each method found the BKS, and the average and maximum running time. Values in bold indicate that a method found the BKS.

Table A.3 – Results of ILS(H)+HC and ILS(S)+HC on the 20 small instances

		Gurobi		ILS(H)+HC					ILS(S)+HC					
Instance	BKS	Best	G(%)	Best	G(%)	Avg.	G(%)	t (s)	Best	G(%)	Avg.	G(%)	t (s)	t *(s)
tc2c10s2cf0	21.77	**21.77**	0.00	22.49	3.31	22.51	3.40	0.61	**21.77**	0.00	21.77	0.00	10.54	8.53
tc0c10s2cf1	19.75	**19.75**	0.00	**19.75**	0.00	20.12	1.87	0.60	**19.75**	0.00	20.12	1.87	4.59	3.86
tc1c10s2cf2	9.03	**9.03**	0.00	**9.03**	0.00	9.04	0.11	0.57	**9.03**	0.00	9.07	0.44	3.71	2.43
tc1c10s2cf3	16.37	**16.37**	0.00	**16.37**	0.00	16.37	0.00	0.83	**16.37**	0.00	16.37	0.00	7.58	5.63
tc1c10s2cf4	16.10	**16.10**	0.00	**16.10**	0.00	16.13	0.19	0.66	**16.10**	0.00	16.10	0.00	6.67	4.79
tc2c10s2ct0	12.45	**12.45**	0.00	**12.45**	0.00	12.53	0.64	0.49	**12.45**	0.00	12.45	0.00	8.36	5.38
tc0c10s2ct1	12.30	**12.30**	0.00	**12.30**	0.00	12.35	0.41	0.62	**12.30**	0.00	12.34	0.33	6.08	3.99
tc1c10s2ct2	10.75	**10.75**	0.00	10.76	0.09	10.77	0.19	0.54	**10.75**	0.00	10.75	0.00	6.51	4.21
tc1c10s2ct3	13.17	**13.17**	0.00	**13.17**	0.00	14.16	7.52	0.84	**13.17**	0.00	13.18	0.08	9.85	7.56
tc1c10s3ct4	13.21	**13.21**	0.00	**13.21**	0.00	13.25	0.30	0.68	**13.21**	0.00	13.21	0.00	11.06	6.01
tc2c10s3cf0	21.77	**21.77**	0.00	22.49	3.31	22.51	3.40	0.61	**21.77**	0.00	21.77	0.00	11.86	8.90
tc0c10s3cf1	19.75	**19.75**	0.00	**19.75**	0.00	20.12	1.87	0.63	**19.75**	0.00	20.12	1.87	6.46	4.41
tc1c10s3cf2	9.03	**9.03**	0.00	9.03	0.00	9.04	0.11	0.56	**9.03**	0.00	9.06	0.33	4.11	2.36
tc1c10s3cf3	16.37	**16.37**	0.00	**16.37**	0.00	16.37	0.00	0.82	**16.37**	0.00	16.37	0.00	9.50	6.06
tc1c10s3cf4	14.90	**14.90**	0.00	**14.90**	0.00	14.94	0.27	0.62	**14.90**	0.00	14.90	0.00	11.27	6.72
tc2c10s3ct0	11.51	**11.51**	0.00	11.54	0.26	11.65	1.22	0.51	**11.51**	0.00	11.54	0.26	11.18	6.81
tc0c10s3ct1	10.80	**10.80**	0.00	**10.80**	0.00	10.81	0.09	0.60	**10.80**	0.00	10.80	0.00	8.91	4.83
tc1c10s3ct2	9.20	**9.20**	0.00	10.76	16.96	10.77	17.07	0.54	**9.20**	0.00	9.34	1.52	9.66	5.33
tc1c10s3ct3	13.02	**13.02**	0.00	**13.02**	0.00	13.12	0.77	0.64	**13.02**	0.00	13.02	0.00	15.72	9.77
tc1c10s2ct4	13.83	**13.83**	0.00	**13.83**	0.00	13.83	0.00	0.77	**13.83**	0.00	13.83	0.00	7.08	4.84
Avg. Gap			0.00		1.20		1.97			0.00		0.34		
Max. Gap			-		16.96		17.07			0.00		1.87		
Best		20		15					20					
Avg. Time								0.64					8.54	5.62
Max. Time								0.84					15.72	9.77

Table A.4 – Results of ILS(H)+HC and ILS(S)+HC on the 100 large instances

		Gurobi		ILS(H)+HC					ILS(S)+HC					
Instance	BKS	Best	G(%)	Best	G(%)	Avg.	G(%)	t (s)	Best	G(%)	Avg.	G(%)	t (s)	t *(s)
tc2c20s3cf0	24.68	24.73	0.20	**24.68**	0.00	24.71	0.12	2.14	**24.68**	0.00	24.68	0.00	22.12	13.86
tc1c20s3cf1	17.50	17.55	0.29	17.51	0.06	17.68	1.03	1.39	**17.50**	0.00	17.53	0.17	19.54	12.32
tc0c20s3cf2	27.60	28.54	3.41	27.61	0.04	27.65	0.18	3.14	**27.60**	0.00	27.66	0.22	16.06	11.77
tc1c20s3cf3	16.63	16.81	1.08	**16.63**	0.00	16.79	0.96	1.41	**16.63**	0.00	16.78	0.90	13.15	8.41
tc1c20s3cf4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	1.20	**17.00**	0.00	17.00	0.00	5.77	3.77

Continued on next page

[1] tcαcβsμcϵ#, where α is the type of the location of the customers (i.e., 0:randomize, 1:cluster, 2: mixture of both), β is the number of customers, μ is the number of the CSs, ϵ is 't' if we use a p-median heuristic to locate the CSs and 'f' otherwise, and # is the number of the instance for each combination of parameters (i.e., $\# = 0, 1, 2, 3, 4$)

[2] $G(\%) = (of - of_{BKS})/of_{BKS} \times 100$

Table A.4 – continued from previous page

		Gurobi		ILS(H)+HC					ILS(S)+HC					
Instance	BKS	Avg	G(%)	Best	G(%)	Avg.	G(%)	t (s)	Best	G(%)	Avg.	G(%)	t (s)	t *(s)
tc2c20s3ct0	25.79	**25.79**	0.00	**25.79**	0.00	25.80	0.04	2.38	**25.79**	0.00	25.79	0.00	23.31	14.66
tc1c20s3ct1	18.95	19.38	2.27	19.55	3.17	19.65	3.69	1.58	**18.95**	0.00	19.38	2.27	23.49	15.25
tc0c20s3ct2	17.08	17.11	0.18	**17.08**	0.00	17.18	0.59	1.73	**17.08**	0.00	17.13	0.29	12.49	8.49
tc1c20s3ct3	12.65	12.68	0.24	12.75	0.79	12.82	1.34	1.76	**12.65**	0.00	12.72	0.55	15.36	8.86
tc1c20s3ct4	16.21	**16.21**	0.00	16.25	0.25	16.31	0.62	1.26	**16.21**	0.00	16.25	0.25	9.74	5.16
tc2c20s4cf0	24.67	25.36	2.80	25.29	2.51	25.35	2.76	1.77	**24.67**	0.00	24.69	0.08	25.90	14.63
tc1c20s4cf1	16.39	16.40	0.06	17.16	4.70	17.47	6.59	1.53	**16.39**	0.00	16.40	0.06	27.19	13.47
tc0c20s4cf2	27.48	-	-	27.60	0.44	27.65	0.62	3.04	**27.48**	0.00	27.61	0.47	18.53	12.81
tc1c20s4cf3	16.56	16.80	1.45	16.80	1.45	16.84	1.69	1.44	**16.56**	0.00	16.80	1.45	14.77	8.69
tc1c20s4cf4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	1.17	**17.00**	0.00	17.00	0.00	7.61	4.17
tc2c20s4ct0	26.02	-	-	26.49	1.81	26.51	1.88	2.01	**26.02**	0.00	26.02	0.00	25.92	15.25
tc1c20s4ct1	18.25	**18.25**	0.00	19.51	6.90	19.65	7.67	1.58	**18.25**	0.00	18.32	0.38	27.11	16.14
tc0c20s4ct2	16.99	17.21	1.29	17.06	0.41	17.12	0.77	1.62	**16.99**	0.00	17.10	0.65	15.25	9.33
tc1c20s4ct3	14.43	**14.43**	0.00	14.56	0.90	14.58	1.04	1.41	**14.43**	0.00	14.50	0.49	14.30	7.99
tc1c20s4ct4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	1.49	**17.00**	0.00	17.00	0.00	11.74	6.08
tc0c40s5cf0	32.67	-	-	33.84	3.58	34.53	5.69	6.09	**32.67**	0.00	33.25	1.78	46.66	23.85
tc1c40s5cf1	65.16	-	-	65.32	0.25	66.64	2.27	11.90	**65.16**	0.00	66.03	1.34	65.41	44.01
tc2c40s5cf2	27.54	38.93	41.36	28.22	2.47	28.86	4.79	7.67	**27.54**	0.00	27.67	0.47	48.50	31.64
tc2c40s5cf3	19.74	21.04	6.59	20.44	3.55	20.82	5.47	4.70	**19.74**	0.00	20.18	2.23	30.49	16.85
tc0c40s5cf4	30.77	36.47	18.52	33.06	7.44	34.21	11.18	11.08	**30.77**	0.00	31.49	2.34	49.91	33.33
tc0c40s5ct0	28.72	-	-	29.22	1.74	29.78	3.69	8.55	**28.72**	0.00	29.35	2.19	41.76	24.50
tc1c40s5ct1	52.68	-	-	54.54	3.53	55.05	4.50	12.64	**52.68**	0.00	53.36	1.29	94.40	58.52
tc2c40s5ct2	26.91	-	-	26.99	0.30	27.15	0.89	8.18	**26.91**	0.00	27.02	0.41	38.38	22.85
tc2c40s5ct3	23.54	-	-	23.56	0.08	23.90	1.53	5.77	**23.54**	0.00	23.77	0.98	43.87	26.48
tc0c40s5ct4	28.63	-	-	29.72	3.81	30.84	7.72	10.49	**28.63**	0.00	28.72	0.31	45.55	32.55
tc0c40s8cf0	31.28	-	-	32.73	4.64	33.68	7.67	6.11	**31.28**	0.00	32.02	2.37	72.91	33.59
tc1c40s8cf1	40.75	-	-	45.86	12.54	50.71	24.44	12.11	**40.75**	0.00	42.33	3.88	108.49	69.99
tc2c40s8cf2	27.15	29.19	7.51	28.05	3.31	28.19	3.83	7.87	**27.15**	0.00	27.31	0.59	57.90	28.92
tc2c40s8cf3	19.66	22.01	11.95	19.86	1.02	20.17	2.59	5.41	**19.66**	0.00	20.24	2.95	45.15	19.46
tc0c40s8cf4	29.32	-	-	32.53	10.95	33.69	14.90	9.93	**29.32**	0.00	29.86	1.84	91.10	43.05
tc0c40s8ct0	26.35	30.29	14.95	27.65	4.93	28.32	7.48	6.01	**26.35**	0.00	26.89	2.05	71.70	28.54
tc1c40s8ct1	40.56	-	-	49.35	21.67	49.85	22.90	11.86	**40.56**	0.00	41.19	1.55	124.31	70.50
tc2c40s8ct2	26.33	-	-	26.82	1.86	27.07	2.81	7.12	**26.33**	0.00	26.71	1.44	58.68	25.64
tc2c40s8ct3	22.71	23.51	3.52	23.26	2.42	23.44	3.21	5.16	**22.71**	0.00	23.23	2.29	63.76	25.25
tc0c40s8ct4	29.20	-	-	29.82	2.12	31.68	8.49	10.93	**29.20**	0.00	29.27	0.24	84.36	47.46
tc0c80s8cf0	39.43	-	-	39.78	0.89	40.52	2.76	31.70	**39.43**	0.00	39.86	1.09	104.77	56.41
tc0c80s8cf1	45.23	-	-	46.48	2.76	47.33	4.64	76.55	**45.23**	0.00	45.73	1.11	183.74	121.27
tc1c80s8cf2	30.81	-	-	32.52	5.55	33.30	8.08	31.57	**30.81**	0.00	31.83	3.31	79.36	50.99
tc2c80s8cf3	32.44	-	-	32.53	0.28	32.90	1.42	43.83	**32.44**	0.00	32.60	0.49	95.72	64.05
tc2c80s8cf4	49.29	-	-	50.41	2.27	51.06	3.59	45.59	**49.29**	0.00	49.69	0.81	160.43	99.84
tc0c80s8ct0	41.90	-	-	42.18	0.67	42.89	2.36	27.68	**41.90**	0.00	42.76	2.05	99.20	54.35
tc0c80s8ct1	45.27	-	-	46.39	2.47	47.50	4.93	82.12	**45.27**	0.00	45.85	1.28	195.16	129.66
tc1c80s8ct2	31.74	-	-	32.59	2.68	32.90	3.65	37.77	**31.74**	0.00	32.36	1.95	93.21	59.73
tc2c80s8ct3	32.31	-	-	32.74	1.33	33.41	3.40	25.10	**32.31**	0.00	32.55	0.74	111.47	65.15
tc2c80s8ct4	44.83	-	-	49.08	9.48	50.31	12.22	43.42	**44.83**	0.00	46.61	3.97	178.48	111.24
tc0c80s12cf0	34.64	-	-	36.01	3.95	37.25	7.53	29.39	**34.64**	0.00	35.59	2.74	126.00	57.24
tc0c80s12cf1	42.90	-	-	43.81	2.12	45.51	6.08	35.17	**42.90**	0.00	44.07	2.73	157.55	74.58
tc1c80s12cf2	29.54	-	-	32.61	10.39	33.34	12.86	31.52	**29.54**	0.00	30.73	4.03	112.44	61.34
tc2c80s12cf3	31.97	-	-	34.10	6.66	35.13	9.88	25.63	**31.97**	0.00	32.70	2.28	159.17	75.64
tc2c80s12cf4	43.89	-	-	47.95	9.25	48.57	10.66	50.96	**43.89**	0.00	44.97	2.46	274.06	131.13
tc0c80s12ct0	39.31	-	-	39.97	1.68	40.48	2.98	32.01	**39.31**	0.00	39.83	1.32	159.79	65.54
tc0c80s12ct1	41.94	-	-	42.56	1.48	43.67	4.12	35.06	**41.94**	0.00	43.03	2.60	162.38	73.32
tc1c80s12ct2	29.52	-	-	31.11	5.39	32.33	9.52	29.45	**29.52**	0.00	30.66	3.86	122.60	58.85
tc2c80s12ct3	30.83	-	-	32.09	4.09	32.31	4.80	28.06	**30.83**	0.00	31.59	2.47	123.57	57.57
tc2c80s12ct4	42.40	-	-	47.16	11.23	48.40	14.15	44.51	**42.40**	0.00	42.82	0.99	276.16	134.33
tc1c160s16cf0	79.80	-	-	88.37	10.74	90.49	13.40	298.56	**79.80**	0.00	80.75	1.19	1139.49	765.69
tc2c160s16cf1	60.34	-	-	61.56	2.02	63.57	5.35	181.57	**60.34**	0.00	61.26	1.52	464.11	273.86
tc0c160s16cf2	61.20	-	-	63.85	4.33	65.42	6.90	224.18	**61.20**	0.00	62.99	2.92	600.43	365.10
tc1c160s16cf3	71.76	-	-	73.93	3.02	75.04	4.57	331.06	**71.76**	0.00	72.75	1.38	666.64	461.58
tc0c160s16cf4	82.92	-	-	98.16	18.38	101.13	21.96	536.94	**82.92**	0.00	83.84	1.11	1662.82	1213.20
tc1c160s16ct0	79.04	-	-	83.82	6.05	85.47	8.14	391.41	**79.04**	0.00	79.90	1.09	1012.72	643.27
tc2c160s16ct1	60.27	-	-	61.97	2.82	62.64	3.93	177.27	**60.27**	0.00	60.62	0.58	507.69	287.64
tc0c160s16ct2	60.13	-	-	64.10	6.60	64.50	7.27	204.82	**60.13**	0.00	62.80	4.44	587.52	341.86
tc1c160s16ct3	73.29	-	-	75.29	2.73	76.55	4.45	180.48	**73.29**	0.00	75.11	2.48	483.20	278.67
tc0c160s16ct4	82.37	-	-	95.78	16.28	97.20	18.00	433.62	**82.37**	0.00	83.08	0.86	1413.91	944.60
tc1c160s24cf0	78.60	-	-	85.59	8.89	87.66	11.53	346.79	**78.60**	0.00	79.30	0.89	1343.54	741.12
tc2c160s24cf1	59.82	-	-	61.30	2.47	63.62	6.35	182.55	**59.82**	0.00	61.14	2.21	653.44	304.66
tc0c160s24ct2	59.25	-	-	62.93	6.21	63.31	6.85	206.85	**59.25**	0.00	60.19	1.59	861.85	409.80
tc1c160s24ct3	68.72	-	-	71.78	4.45	74.54	8.47	196.47	**68.72**	0.00	69.98	1.83	756.39	358.35
tc0c160s24cf4	81.44	-	-	95.47	17.23	99.35	21.99	508.19	**81.44**	0.00	82.13	0.85	1984.26	1209.32
tc1c160s24ct0	78.21	-	-	83.38	6.61	84.84	8.48	284.88	**78.21**	0.00	79.35	1.46	1183.70	577.83

Continued on next page

Table A.4 – continued from previous page

		Gurobi		ILS(H)+HC					ILS(S)+HC					
Instance	BKS	Avg	G(%)	Best	G(%)	Avg.	G(%)	t (s)	Best	G(%)	Avg.	G(%)	t (s)	t *(s)
tc2c160s24ct1	59.13	-	-	60.84	2.89	62.49	5.68	192.18	**59.13**	0.00	59.72	1.00	748.95	340.40
tc0c160s24cf2	59.27	-	-	62.63	5.67	64.12	8.18	210.13	**59.27**	0.00	60.92	2.78	845.72	403.33
tc1c160s24cf3	68.56	-	-	72.83	6.23	75.18	9.66	240.33	**68.56**	0.00	69.57	1.47	883.61	483.10
tc0c160s24ct4	80.96	-	-	90.55	11.85	93.83	15.90	453.34	**80.96**	0.00	82.11	1.42	1736.76	956.94
tc2c320s24cf0	182.52	-	-	195.23	6.96	210.45	15.30	3762.57	**182.52**	0.00	186.94	2.42	7855.89	6566.41
tc2c320s24cf1	95.51	-	-	97.39	1.97	100.07	4.77	1003.08	**95.51**	0.00	96.42	0.95	1927.61	1456.16
tc1c320s24cf2	152.23	-	-	177.71	16.74	185.68	21.97	3162.07	**152.23**	0.00	153.99	1.16	8370.48	7105.63
tc1c320s24cf3	117.48	-	-	124.23	5.75	126.08	7.32	2089.09	**117.48**	0.00	118.36	0.75	3737.73	3065.82
tc2c320s24cf4	122.88	-	-	134.30	9.29	136.17	10.82	2177.20	**122.88**	0.00	124.68	1.46	4961.50	3681.14
tc2c320s24ct0	181.50	-	-	208.32	14.78	212.18	16.90	4434.40	**181.50**	0.00	186.23	2.61	8606.62	7204.02
tc2c320s24ct1	94.73	-	-	96.69	2.07	99.71	5.26	942.21	**94.73**	0.00	96.49	1.86	1737.70	1259.26
tc1c320s24ct2	148.77	-	-	173.82	16.84	182.34	22.57	3617.64	**148.77**	0.00	154.13	3.60	8231.61	6853.35
tc1c320s24ct3	116.64	-	-	122.75	5.24	125.71	7.78	1984.42	**116.64**	0.00	119.17	2.17	3783.98	3273.79
tc2c320s24ct4	122.02	-	-	131.87	8.07	133.68	9.56	3074.58	**122.02**	0.00	123.85	1.50	5447.73	4273.94
tc2c320s38cf0	177.01	-	-	202.48	14.39	207.83	17.41	4007.13	**177.01**	0.00	182.31	2.99	9150.09	6733.82
tc2c320s38cf1	94.29	-	-	97.55	3.46	99.54	5.57	1082.91	**94.29**	0.00	95.07	0.83	2443.29	1601.78
tc1c320s38cf2	141.68	-	-	173.71	22.61	181.84	28.35	3208.78	**141.68**	0.00	147.08	3.81	9490.84	7235.62
tc1c320s38cf3	116.33	-	-	122.49	5.30	125.30	7.71	2024.26	**116.33**	0.00	117.74	1.21	4600.98	3113.71
tc2c320s38cf4	122.32	-	-	128.72	5.23	131.01	7.10	1814.78	**122.32**	0.00	123.47	0.94	4138.21	2660.68
tc2c320s38ct0	191.09	-	-	205.08	7.32	208.44	9.08	4766.36	**191.09**	0.00	192.15	0.55	10335.56	7636.50
tc2c320s38ct1	94.53	-	-	97.44	3.08	98.62	4.33	938.85	**94.53**	0.00	95.29	0.80	2284.16	1408.88
tc1c320s38ct2	141.14	-	-	172.99	22.57	181.62	28.68	3660.78	**141.14**	0.00	145.09	2.80	9264.46	6974.34
tc1c320s38ct3	116.07	-	-	122.91	5.89	126.17	8.70	1993.12	**116.07**	0.00	117.71	1.41	4559.66	3062.95
tc2c320s38ct4	121.66	-	-	127.40	4.72	130.35	7.14	1634.65	**121.66**	0.00	123.15	1.22	4265.37	2784.91
Avg. Gap			4.71		5.28		7.51			0.00		1.51		
Max. Gap			41.36		22.61		28.68			0.00		4.44		
Found solution		25		100					100					
Best		7		7			3		100					
Avg. Time								581.52					1393.38	1018.33
Max. Time								4766.36					10335.56	7636.50

A.3 Detailed results for the TRP-CEV

Tables A.5 shows the results of our PMa for the small TRP-CEV instances. We compare our results with the optimal solutions found by Gurobi using the MILP formulation presented in Section 3.3.2. For the results obtained with the PMa, we report the best solution, the average solution (Avg.), and the average computing time (t in seconds) over ten runs. The last rows of the table summarize the average gap, the number of times PMa found the BKS, and the average running time. Values in bold indicate that a method found the BKS. Tables A.6 shows the results of our PMa for the large TRP-CEV instances. For each instance and percentage of EVs in the fleet, we report the best solution, the average solution (Avg.), and the average computing time (t in seconds) over ten runs. The last rows of the table summarize the average gap, and the average running time.

Table A.5 – Results of PMa on small instances of TRP-CEV

Instance	Gurobi	PMa		
	Best	Best	Avg	Time (s)
1-5	805.30	805.30	805.30	0.01
2-5	806.31	806.31	806.31	0.02
3-5	801.26	801.26	801.26	0.02
4-5	802.52	802.51	802.51	0.02
5-5	801.63	801.63	801.63	0.02
1-10	810.14	810.14	810.14	1.32
2-10	1625.13	1625.13	1625.13	2.75
3-10	807.61	807.61	807.61	1.30
4-10	802.10	802.10	802.10	0.02
5-10	803.01	803.01	803.01	0.03
Avg. Gap		0.00	0.00	
Best		10		
Avg. Time				0.55

Table A.6 – Results of PMa on large instances of TRP-CEV

% of EVs in the fleet	0%			20%			40%			60%			80%			100%		
Instance	**Best**	**Avg.**	**t(min)**	**Best**	**Avg.**	**t(min)**	**Best**	**Avg.**	**t(min)**	**Best**	**Avg.**	**t(min)**	**Best**	**Avg.**	**t(min)**	**Best**	**Avg.**	**t(min)**
rural_18	7309.13	7347.27	6.13	7062.45	7104.56	10.36	6856.54	6905.22	10.79	6791.60	6817.33	13.60	6639.04	6685.71	N.R.	N.R.	N.R.	N.R.
rural_19	4568.71	4577.93	10.07	4337.86	4354.07	17.17	4243.67	4249.34	8.72	4243.14	4249.58	10.79	4243.14	4247.26	N.R.	N.R.	N.R.	N.R.
rural_20	6533.76	6552.38	9.74	6263.10	6277.22	19.66	6066.01	6079.87	21.25	5976.84	6013.09	18.53	5885.27	5934.11	N.R.	N.R.	N.R.	N.R.
rural_21	8465.34	8508.08	18.73	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.	N.R.
rural_22	6611.81	6700.23	20.53	6356.17	6475.38	20.17	6131.63	6181.34	20.36	6033.61	6068.19	19.41	5863.80	5871.67	N.R.	N.R.	N.R.	N.R.
semi_urbain_18	4362.85	4388.31	20.02	4158.41	4260.54	17.93	4026.93	4599.33	19.62	4028.21	4745.61	21.18	4024.50	4585.72	18.70	4024.97	4345.65	20.14
semi_urbain_19	3395.19	3400.45	19.51	3278.80	3285.17	19.24	3212.82	3213.21	19.34	3212.06	3212.36	19.98	3212.09	3212.39	20.85	3212.08	3212.28	20.28
semi_urbain_20	6898.40	7598.07	18.17	7503.12	7522.85	20.03	6582.82	7288.81	19.26	7275.06	7285.98	18.11	7233.03	7235.50	18.55	7231.85	7234.76	19.63
semi_urbain_21	6899.58	7537.84	22.18	7490.38	7522.74	19.38	6598.19	7264.17	20.31	7265.63	7274.67	19.14	6436.39	7078.40	19.05	7232.37	7234.52	18.63
semi_urbain_22	6096.59	6132.66	19.95	5874.59	5890.09	20.20	5683.24	6260.91	18.19	5631.99	6125.93	21.51	5634.75	6114.44	18.59	5630.73	6032.78	20.85
urbain_18	3355.42	3362.49	19.93	3331.72	3359.44	23.86	3211.44	3227.53	20.73	3210.76	3211.37	19.11	3210.76	3211.37	21.05	3210.76	3211.37	18.00
urbain_19	4104.48	4111.86	19.12	4090.84	4111.68	19.31	4055.55	4614.59	20.06	4031.57	4457.53	10.44	4816.51	4824.00	17.35	4816.74	4818.88	20.99
urbain_21	3342.50	3358.59	13.70	3256.36	3260.79	16.41	3211.39	3211.95	16.27	3210.53	3211.52	15.09	3210.53	3211.53	14.29	3210.53	3211.35	13.36
urbain_22	3356.74	3361.97	17.35	3356.74	3361.97	11.83	3356.74	3361.97	17.19	3356.74	3361.97	18.47	3214.70	3853.71	18.44	3214.70	3853.35	15.23
Avg. Gap		1.76						4.73			3.03			4.18			3.90	
Avg. Time			16.80			18.12			17.85			17.34			18.54			18.57

A.4 Detailed results for the E-FSMFTW

Tables A.7 and A.8 show the results of our PMa on the small and large E-FSMFTW instances. In Table A.7, we compare the results of our PMa and the ALNS by Hiermann et al. (2016) with the optimal solution found by the BnP by Hiermann et al. (2016). In Table A.8, we compare the results obtained by the PMa and ALNS with the BKSs taken from Hiermann et al. (2016) [3] and updated with some new BKSs found by our PMa.

Hiermann et al. (2016) reported the best solution (Best) only for the small instances, the average cost (Avg.), and the average computing time (t in minutes) over ten runs of their AVNS. The last rows of the table summarize the average BKS gap, the number of times each method found the BKS, and the average running time. Values in bold indicate that a method found the BKS.

[3] Hiermann et al. (2016) report the BKSs obtained using BnP or any run of their experiments

Table A.7 – Results of PMa on small instances of Hiermann et al. (2016).

Type	A							B							C						
Instance	BnP	ALNS			PMa			BnP	ALNS			PMa			BnP	ALNS			PMa		
	Best	Best	Avg.	t (min)	Best	Avg.	t (min)	Best	Best	Avg.	t (min)	Best	Avg.	t (min)	Best	Best	Avg.	t (min)	Best	Avg.	t (min)
c101c5	857.75	**857.75**	857.75	0.28	**857.75**	857.75	0.03	377.75	**377.75**	377.75	0.23	**377.75**	377.75	0.02	317.75	**317.75**	317.75	0.17	**317.75**	317.75	0.02
c103c5	476.05	**476.05**	476.05	0.21	**476.05**	476.05	0.02	236.05	**236.05**	236.05	0.15	**236.05**	236.05	0.02	206.05	**206.05**	206.05	0.15	**206.05**	206.05	0.02
c206c5	1261.88	**1261.88**	1262.19	0.22	**1261.88**	1261.88	0.02	461.88	**461.88**	461.88	0.16	**461.88**	461.88	0.02	361.88	**361.88**	361.88	0.16	**361.88**	361.88	0.02
c208c5	1164.34	**1164.34**	1164.34	0.18	**1164.34**	1164.34	0.02	364.34	**364.34**	364.34	0.11	**364.34**	364.34	0.02	264.34	**264.34**	264.34	0.15	**264.34**	264.34	0.02
r104c5	327.25	**327.25**	327.25	0.21	**327.25**	327.25	0.03	175.25	**175.25**	175.25	0.12	**175.25**	175.25	0.02	156.25	**156.25**	156.25	0.14	**156.25**	156.25	0.02
r105c5	346.08	**346.08**	346.08	0.19	**346.08**	346.08	0.02	194.08	**194.08**	194.08	0.13	**194.08**	194.08	0.02	175.08	**175.08**	175.08	0.13	**175.08**	175.08	0.02
r202c5	609.18	**609.18**	609.18	0.16	610.54	610.54	0.03	249.18	**249.18**	249.18	0.11	250.54	250.54	0.02	204.18	**204.18**	204.18	0.11	205.54	205.54	0.02
r203c5	645.63	**645.63**	645.63	0.20	**645.63**	645.63	0.03	285.63	**285.63**	285.63	0.11	**285.63**	285.63	0.02	240.63	**240.63**	240.63	0.13	**240.63**	240.63	0.02
rc105c5	511.96	**511.96**	511.96	0.14	**511.96**	511.96	0.02	295.96	**295.96**	295.96	0.15	**295.96**	295.96	0.02	268.96	**268.96**	268.96	0.13	**268.96**	268.96	0.02
rc108c5	532.04	**532.04**	532.04	0.18	**532.04**	532.04	0.02	313.93	**313.93**	313.93	0.17	**313.93**	313.93	0.02	283.93	**283.93**	283.93	0.17	**283.93**	283.93	0.02
rc204c5	496.74	**496.74**	496.74	0.12	498.74	498.74	0.02	255.55	**255.55**	255.79	0.17	258.74	258.74	0.02	220.55	**220.55**	220.86	0.16	228.74	228.74	0.02
rc208c5	328.89	**328.89**	328.89	0.15	**328.89**	328.89	0.02	208.89	**208.89**	208.89	0.14	**208.89**	208.89	0.02	193.89	**193.89**	193.89	0.13	**193.89**	193.89	0.02
c101c10	1302.15	**1302.15**	1302.15	0.24	1308.48	1308.48	0.03	582.15	**582.15**	582.15	0.25	588.48	588.48	0.03	492.15	**492.15**	492.15	0.27	498.48	498.48	0.04
c104c10	902.71	**902.71**	902.71	0.30	**902.71**	902.71	0.05	422.71	**422.71**	422.71	0.31	**422.71**	422.71	0.05	362.71	**362.71**	362.71	0.28	**362.71**	362.71	0.05
c202c10	1304.32	**1304.32**	1493.84	0.25	1312.91	1312.91	0.04	504.32	**504.32**	548.61	0.24	512.91	512.91	0.04	404.32	**404.32**	413.84	0.24	412.91	412.91	0.04
c205c10	2631.97	**2631.97**	2631.97	0.24	2661.17	2661.17	0.04	711.97	**711.97**	711.97	0.24	741.17	741.17	0.04	471.97	**471.97**	471.97	0.17	501.17	501.17	0.04
r102c10	595.34	**595.34**	595.34	0.23	**595.34**	595.34	0.05	325.19	**325.19**	325.19	0.24	**325.19**	325.19	0.05	287.19	**287.19**	287.19	0.20	**287.19**	287.19	0.05
r103c10	478.07	**478.07**	478.23	0.24	**478.07**	483.59	0.05	262.07	**262.07**	262.07	0.29	**262.07**	262.07	0.05	235.07	**235.07**	235.07	0.34	**235.07**	235.07	0.05
r201c10	1138.38	**1138.38**	1140.45	0.27	**1138.38**	1138.38	0.05	418.38	**418.38**	418.38	0.28	**418.38**	418.38	0.05	328.38	**328.38**	328.38	0.29	**328.38**	328.38	0.05
r203c10	952.16	**952.16**	956.50	0.23	956.59	956.59	0.06	392.16	**392.16**	438.29	0.23	396.59	396.59	0.06	322.16	**322.16**	350.29	0.22	326.59	326.59	0.06
rc102c10	1100.27	**1100.27**	1100.27	0.27	**1100.27**	1100.27	0.02	572.27	**572.27**	572.27	0.23	**572.27**	572.27	0.03	498.51	**498.51**	498.51	0.22	**498.51**	498.51	0.02
rc108c10	693.27	**693.27**	693.27	0.35	**693.27**	693.27	0.04	422.12	**422.12**	422.12	0.26	424.09	424.09	0.04	386.12	**386.12**	386.12	0.25	388.09	388.09	0.04
rc201c10	684.63	**684.63**	684.88	0.30	686.32	686.32	0.04	429.17	**429.17**	429.17	0.45	434.83	434.83	0.04	384.17	**384.17**	384.17	0.32	389.83	389.83	0.04
rc205c10	1064.59	**1064.59**	1067.44	0.29	1066.14	1066.14	0.05	504.59	**504.59**	504.59	0.38	506.14	506.14	0.05	429.69	**429.69**	429.69	0.27	**429.69**	429.69	0.05
c103c15	1291.03	**1291.03**	1291.46	0.50	**1291.03**	1291.03	0.11	571.03	**571.03**	571.78	0.55	**571.03**	571.03	0.10	481.03	**481.03**	481.64	0.43	**481.03**	481.03	0.10
c106c15	1253.59	**1253.59**	1253.59	0.49	**1253.59**	1253.59	0.08	533.59	**533.59**	538.28	0.55	**533.59**	533.59	0.08	415.13	**415.13**	440.52	0.48	**415.13**	415.13	0.07
c202c15	2403.35	**2403.35**	2403.35	0.64	**2403.35**	2403.35	0.10	803.35	**803.35**	803.35	0.61	**803.35**	803.35	0.10	603.35	**603.35**	607.82	0.61	**603.35**	603.35	0.10
c208c15	2325.89	**2325.89**	2325.89	0.53	**2325.89**	2325.89	0.11	725.89	**725.89**	725.89	0.47	**725.89**	725.89	0.11	525.89	**525.89**	526.29	0.38	**525.89**	525.89	0.11
r102c15	850.58	**850.58**	854.99	0.50	**850.58**	850.58	0.08	511.55	**511.55**	514.84	0.56	**511.55**	511.55	0.07	465.94	**465.94**	467.00	0.51	466.55	466.55	0.07
r105c15	760.13	**760.13**	762.48	0.41	**760.13**	760.13	0.08	436.89	**436.89**	436.89	0.43	**436.89**	436.89	0.08	387.89	**387.89**	387.94	0.38	**387.89**	387.89	0.08
r202c15	1311.24	**1311.24**	1316.84	0.92	**1311.24**	1311.24	0.15	591.24	**591.24**	591.24	0.74	**591.24**	591.24	0.15	495.64	**495.64**	495.64	0.55	498.05	498.05	0.15
r209c15	1033.50	**1033.50**	1033.50	0.64	**1033.50**	1033.50	0.15	473.50	**473.50**	473.50	0.63	**473.50**	473.50	0.15	403.50	**403.50**	403.50	0.55	**403.50**	403.50	0.15
rc103c15	840.97	**840.97**	840.98	0.52	**840.97**	843.01	0.08	499.67	**499.67**	499.67	0.43	**499.67**	499.67	0.07	448.67	**448.67**	448.67	0.43	**448.67**	448.67	0.07
rc108c15	1013.70	**1013.70**	1033.14	0.47	**1013.70**	1013.70	0.09	514.78	**514.78**	514.78	0.44	**514.78**	514.78	0.08	445.25	**445.25**	445.95	0.41	**445.25**	445.25	0.08
rc202c15	1101.61	**1101.61**	1101.61	0.56	**1101.61**	1101.61	0.13	541.61	**541.61**	541.61	0.51	**541.61**	541.61	0.13	471.61	**471.61**	471.61	0.44	**471.61**	471.61	0.13
rc204c15	810.90	**810.90**	810.90	0.54	**810.90**	810.90	0.21	410.90	**410.90**	410.90	0.59	**410.90**	410.90	0.21	360.90	**360.90**	360.90	0.52	**360.90**	360.90	0.20
Avg. Gap		0.00	0.52		0.10				0.00	0.62		0.33				0.00	0.52		0.50		
Best		36			28				36			27				36			26		
Avg. Time				0.34			0.06				0.32			0.06				0.29			0.06

Table A.8 – Results of PMa on large instances of Hiermann et al. (2016).

Type	A						B						C					
Instance	BKS	ALNS		PMa			BKS	ALNS		PMa			BKS	ALNS		PMa		
		Avg.	t (min)	Best	Avg.	t (min)		Avg.	t (min)	Best	Avg.	t (min)		Avg.	t (min)	Best	Avg.	t (min)
c101	7162.01	7190.21	17.53	**7162.01**	7165.15	25.88	2495.00	2505.73	14.43	**2495.00**	2502.09	25.56	1809.93	1816.06	14.08	1810.12	1820.45	8.84
c102	7139.45	7162.24	17.95	**7139.45**	7142.80	26.02	2445.99	2450.73	14.41	2448.76	2451.39	25.86	1759.73	1766.14	14.36	1763.97	1770.44	9.20
c103	7121.12	7149.31	18.30	**7121.12**	7124.26	31.22	2430.72	2452.40	15.03	**2430.72**	2437.99	31.16	1746.80	1759.20	15.09	**1746.80**	1757.68	20.76
c104	7099.88	7110.43	17.75	**7099.88**	7103.79	32.93	2404.73	2428.95	15.07	**2404.73**	2410.12	32.85	1719.67	1735.86	15.60	1721.53	1727.72	29.04
c105	7140.31	7182.56	17.60	**7140.31**	7152.21	27.24	2472.93	2475.95	14.45	2473.79	2476.46	27.16	1783.25	1785.43	14.51	1785.31	1790.59	10.66
c106	7136.90	7168.94	17.69	**7136.90**	7141.66	27.28	2462.54	2468.13	14.71	2462.90	2466.25	27.20	1774.77	1777.67	14.64	1775.08	1780.39	11.10
c107	7139.12	7171.04	17.42	**7139.12**	7145.54	30.49	2458.37	2461.32	14.60	**2458.37**	2459.31	30.25	1764.02	1768.33	14.57	1765.45	1775.74	14.68
c108	7133.30	7153.77	17.53	**7133.30**	7142.39	30.22	2450.17	2463.02	14.33	2452.58	2455.74	30.17	1761.41	1769.76	14.80	1768.05	1772.78	20.58
c109	7120.33	7132.19	17.32	7123.15	7132.66	33.32	2431.87	2452.57	15.07	**2431.87**	2434.02	33.22	1740.18	1749.07	15.30	1744.09	1749.05	29.92
c201	5736.35	5757.53	39.28	**5736.35**	5741.28	19.36	1730.41	1739.26	35.76	**1730.41**	1730.50	14.14	1210.41	1213.63	31.30	**1210.41**	1210.50	13.50
c202	5740.43	5765.52	42.22	**5740.43**	5753.56	21.50	1729.73	1745.24	38.14	**1729.73**	1730.12	17.01	1209.73	1220.97	34.17	**1209.73**	1210.12	16.58
c203	5726.08	5751.99	47.69	5744.25	5783.33	28.27	1716.29	1742.76	39.38	1730.26	1743.23	23.73	1212.34	1227.69	33.07	1216.18	1226.22	30.01
c204	5705.82	5727.18	50.91	5724.75	5772.72	41.49	1699.07	1709.43	37.02	1724.75	1753.55	36.43	1179.25	1199.37	32.15	1204.68	1232.00	35.62
c205	5694.58	5725.41	46.16	**5694.58**	5701.83	24.43	1694.58	1715.09	37.83	**1694.58**	1701.83	17.33	1188.92	1195.24	31.64	1192.54	1195.35	16.70
c206	5689.08	5714.39	31.20	**5689.08**	5696.99	26.40	1689.08	1712.38	36.78	**1689.08**	1696.99	20.69	1183.42	1192.30	31.01	1188.81	1191.74	20.16
c207	5696.54	5713.44	32.58	**5696.54**	5716.37	27.61	1694.61	1710.70	35.05	1696.54	1715.61	22.19	1183.42	1190.37	31.44	1187.49	1190.65	21.52
c208	5682.60	5707.65	50.23	**5682.60**	5700.58	30.48	1681.47	1707.11	36.46	1682.60	1701.29	24.49	1181.47	1192.96	30.82	1182.60	1192.42	23.78
r101	4366.21	4465.51	14.47	**4366.21**	4379.64	21.41	2249.14	2281.28	13.68	2253.04	2253.93	10.71	1954.00	1977.89	14.36	1959.04	1959.07	9.45
r102	4176.82	4270.92	14.97	**4176.82**	4180.61	13.11	2047.89	2095.87	14.45	2053.42	2057.97	8.01	1757.13	1791.03	14.67	1761.48	1767.00	7.54
r103	4043.29	4130.86	16.43	**4043.29**	4049.01	15.53	1892.84	1927.52	15.11	**1892.84**	1897.06	13.13	1587.32	1618.81	15.52	**1587.32**	1593.13	12.54
r104	3969.12	4025.60	15.27	**3969.12**	3970.77	21.45	1747.65	1775.33	15.28	1749.48	1761.84	22.78	1424.30	1448.31	16.20	1427.76	1437.37	19.66
r105	4138.01	4215.34	15.37	**4138.01**	4140.08	12.94	1997.75	2030.12	15.08	1998.80	2007.01	9.17	1699.34	1728.12	14.68	**1699.34**	1705.56	8.25
r106	4068.15	4155.24	15.54	**4068.15**	4075.44	16.46	1925.83	1963.88	14.51	**1925.83**	1929.13	12.02	1604.55	1635.42	14.67	1609.78	1612.20	8.55
r107	4003.84	4093.59	15.30	**4003.84**	4008.71	24.07	1818.03	1844.20	15.50	**1818.03**	1820.75	22.10	1490.04	1514.01	15.99	1493.12	1499.00	17.35
r108	3955.48	4025.75	15.77	**3955.48**	3961.93	23.96	1720.78	1753.09	16.27	**1720.78**	1733.87	23.31	1397.86	1417.39	16.42	**1397.86**	1404.05	18.13
r109	4016.33	4110.98	15.58	**4016.33**	4024.49	19.92	1863.48	1904.12	15.37	**1863.48**	1867.32	25.36	1550.40	1580.14	15.65	1551.69	1554.22	16.98
r110	3969.02	4045.96	15.73	**3969.02**	3975.95	26.51	1752.40	1793.48	15.54	**1752.40**	1755.92	12.89	1420.13	1471.66	15.64	**1420.13**	1429.38	12.11
r111	3972.84	4048.42	15.93	**3972.84**	3976.91	25.10	1765.08	1808.36	15.66	**1765.08**	1775.04	22.45	1438.81	1479.75	16.16	1439.02	1443.94	12.62
r112	3942.68	4023.01	15.80	**3942.68**	3947.93	21.85	1714.93	1746.02	15.77	**1714.93**	1722.34	26.75	1388.22	1403.82	16.10	**1388.22**	1394.19	25.74
r201	3413.93	3432.83	42.20	3515.45	3543.81	17.91	1594.58	1618.25	31.21	1697.28	1714.30	40.11	1366.63	1378.77	29.69	1474.75	1480.55	39.50
r202	3270.49	3295.26	44.95	3366.13	3397.33	18.23	1468.05	1479.42	29.61	1541.45	1551.73	17.87	1236.97	1249.65	29.13	1316.45	1328.35	42.06
r203	3136.47	3169.97	49.40	3224.34	3252.78	23.86	1340.00	1354.24	30.70	1424.34	1438.89	23.68	1104.85	1124.07	30.23	1183.71	1205.32	56.20
r204	3008.01	3026.09	46.32	3098.94	3114.40	26.84	1203.89	1211.63	27.35	1278.59	1286.75	27.33	977.72	983.97	26.81	1032.48	1050.51	65.96
r205	3234.26	3261.16	40.89	3326.29	3356.01	21.28	1430.70	1455.08	30.16	1524.53	1534.24	20.82	1217.77	1232.63	29.15	1293.54	1308.90	47.77
r206	3172.50	3194.12	47.73	3261.05	3296.06	26.25	1361.69	1376.34	31.35	1461.05	1470.36	27.07	1136.83	1155.47	30.95	1230.67	1239.18	60.69
r207	3079.39	3099.52	46.87	3159.76	3175.13	23.25	1256.22	1268.66	28.18	1359.76	1364.12	23.60	1031.22	1057.22	26.31	1110.93	1136.77	58.58
r208	3010.51	3026.57	51.26	3088.29	3104.77	33.89	1198.39	1208.89	29.02	1281.92	1289.03	34.43	971.15	984.87	28.21	1032.20	1049.20	85.72
r209	3142.72	3161.57	45.06	3224.28	3255.62	25.29	1333.33	1345.50	30.30	1381.34	1407.39	25.18	1099.24	1117.68	29.62	1156.34	1170.58	57.01
r210	3110.90	3143.79	45.94	3181.95	3235.09	25.02	1314.16	1324.07	30.07	1359.10	1385.87	25.44	1087.21	1100.27	29.88	1134.10	1143.99	57.33
r211	3041.93	3079.24	44.29	3133.26	3147.32	20.83	1231.38	1244.73	26.30	1319.15	1326.35	21.33	1006.38	1026.07	25.93	1087.95	1098.37	51.81
rc101	5247.39	5346.49	14.13	**5247.39**	5255.03	13.21	2504.72	2560.33	13.71	**2504.72**	2508.65	9.42	2121.02	2153.24	13.80	**2121.02**	2123.26	8.96
rc102	5114.15	5180.03	14.63	**5114.15**	5118.29	24.81	2330.50	2359.92	14.35	2334.08	2335.72	9.68	1947.54	1972.85	14.68	**1947.54**	1950.16	9.40
rc103	4916.98	5007.37	14.53	**4916.98**	4924.38	18.60	2105.84	2136.78	14.14	2108.82	2113.31	10.78	1726.85	1764.22	14.54	**1726.85**	1736.49	10.72
rc104	4801.06	4862.65	16.03	**4801.06**	4815.22	24.28	1979.16	2002.33	15.56	**1979.16**	1984.66	14.39	1595.44	1614.09	15.80	1596.18	1605.91	15.20
rc105	5060.96	5117.09	14.48	**5060.96**	5068.85	14.74	2252.05	2287.95	13.87	**2252.05**	2264.49	9.81	1875.91	1900.42	14.14	**1875.91**	1880.03	9.93

Continued on next page

Table A.8 – continued from previous page

Type	A						B						C					
Instance	BKS	ALNS		PMa			BKS	ALNS		PMa			BKS	ALNS		PMa		
		Avg.	t (min)	Best	Avg.	t (min)		Avg.	t (min)	Best	Avg.	t (min)		Avg.	t (min)	Best	Avg.	t (min)
rc106	4985.03	5102.46	14.69	**4985.03**	5001.29	14.12	2187.30	2232.05	14.50	**2187.30**	2194.98	10.21	1811.86	1844.99	15.18	**1811.86**	1814.49	9.96
rc107	4835.75	4913.90	15.24	**4835.75**	4853.96	15.67	2034.35	2050.20	15.43	**2034.35**	2039.77	15.20	1639.84	1675.58	15.31	1645.79	1659.27	15.60
rc108	4798.48	4862.41	15.64	**4798.48**	4819.12	23.59	1962.87	1995.41	14.99	1969.75	1975.91	16.03	1578.51	1601.47	14.98	1583.70	1588.87	14.15
rc201	4346.25	4361.17	14.27	4371.85	4388.86	32.88	1899.99	1931.42	15.69	1976.91	2013.94	16.64	1589.99	1617.52	15.64	1664.90	1682.78	14.79
rc202	4273.74	4295.27	14.59	4288.62	4313.51	31.02	1807.30	1825.07	16.19	1876.72	1921.19	19.55	1485.13	1497.99	16.72	1541.00	1587.09	15.11
rc203	4152.94	4186.28	15.98	4191.76	4227.27	36.69	1642.43	1660.93	19.00	1717.98	1749.83	22.51	1310.37	1333.25	19.35	1345.08	1363.51	48.90
rc204	4113.49	4127.11	19.18	4136.86	4160.55	46.62	1521.80	1543.04	22.13	1597.07	1638.89	34.12	1183.16	1193.93	22.69	1212.07	1224.22	77.59
rc205	4246.52	4273.59	14.86	4274.33	4294.99	33.33	1753.79	1774.22	19.03	1832.28	1892.25	21.98	1424.75	1440.00	20.95	1500.24	1508.13	39.75
rc206	4237.75	4270.25	15.09	4260.22	4289.85	33.95	1751.75	1767.75	17.79	1819.18	1883.33	23.71	1431.21	1439.17	18.11	1473.86	1493.94	43.46
rc207	4177.23	4199.60	16.20	4215.09	4242.41	36.42	1616.96	1640.23	19.07	1694.69	1782.89	27.96	1277.71	1299.14	21.30	1344.75	1364.75	49.95
rc208	4097.04	4122.12	18.13	4142.05	4173.32	42.75	1497.95	1520.76	22.03	1578.86	1660.49	32.61	1161.57	1171.52	22.91	1202.33	1216.16	64.83
Avg. Gap		1.01		0.66	1.04			1.24		1.91	2.82			1.29		1.98	2.72	
Best				34						22						13		
Avg. Time			25.68			25.57			21.47			21.96			20.83			28.08

Bibliography

Andelmin, J. & Bartolini, E. (2016), A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation, Technical report, AALTO. 12

Automobile-propre (2015), 'La recharge des voitures électriques', http://www.automobile-propre.com/dossiers/voitures-electriques/recharge-voitures-electriques/Les_modes_de_recharge. Accessed: 2016-09-12. 77

Avella, P., Boccia, M. & Sforza, A. (2004), 'Solving a fuel delivery problem by heuristic and exact approaches', *European Journal of Operational Research* **152**(1), 170–179. 11

Baldacci, R., Battarra, M. & Vigo, D. (2008), Routing a heterogeneous fleet of vehicles, *in* 'The vehicle routing problem: latest advances and new challenges', Springer, pp. 3–27. 66

Bansal, P. (2015), 'Charging of electric vehicles: Technology and policy implications', *Journal of Science Policy & Governance* **6**(1). 38

Beasley, J. E. (1983), 'Route first-cluster second methods for vehicle routing', *Omega* **11**(4), 403–408. 20

Becker, T. A., Sidhu, I. & Tenderich, B. (2009), Electric vehicles in the United States: a new model with forecasts to 2030, Technical report, DOE/EIA-0573(2007) Center for Entrepreneurship and Technology, University of California, Berkeley. 37

Berman, B. & Gartner, J. (2013), 'Selecting electric vehicles for fleets'. url = http://www.navigantresearch.com/wp-content/uploads/2013/02/RB-SEVF-13-Executive-Summary.pdf, Last Accessed= 16/04/2015. 11

Bostel, N., Dejax, P., Guez, P. & Tricoire, F. (2008), Multiperiod planning and routing on a rolling horizon for field force optimization logistics, *in* 'The Vehicle Routing Problem: Latest Advances and New Challenges', Springer, pp. 503–525. 66

Bräysy, O. & Gendreau, M. (2005), 'Vehicle routing problem with time windows, part II: Metaheuristics', *Transportation science* **39**(1), 119–139. 28

Bruglieri, M., Colorni, A. & Luè, A. (2014), 'The vehicle relocation problem for the one-way electric vehicle sharing: An application to the Milan case', *Procedia - Social and Behavioral Sciences* **111**, 18 – 27. 36

Bruglieri, M., Pezzella, F., Pisacane, O. & Suraci, S. (2015), 'A variable neighborhood search branching for the electric vehicle routing problem with time windows', *Electronic Notes in Discrete Mathematics* **47**, 221–228. 36

Bulk, J. V. D. (2009), 'Costs of the electric car'. url = http://www.olino.org/us/articles/2009/02/17/costs-of-the-electric-car, Last Accessed= 23/06/2016. 11

Bullard, J. A. & Kiernan, A. (1922), *Plane and Spherical Trigonometry: With Stereographic Projections*, DC Heath. 27

Castillo-Salazar, J. A., Landa-Silva, D. & Qu, R. (2016), 'Workforce scheduling and routing problems: literature survey and computational study', *Annals of Operations Research* **239**(1), 39–67. 66

Cheng, E. & Rich, J. L. (1998), A home health care routing and scheduling problem, Technical report, Rice University. 66

Conrad, R. G. & Figliozzi, M. A. (2011), The recharging vehicle routing problem, *in* T. Doolen & E. V. Aken, eds, 'Proceedings of the 2011 Industrial Engineering Research Conference', Reno, NV, USA. 12, 20, 41

Cortés, C. E., Gendreau, M., Rousseau, L. M., Souyris, S. & Weintraub, A. (2014), 'Branch-and-price and constraint programming for solving a real-life technician dispatching problem', *European Journal of Operational Research* **238**(1), 300 – 312. 88

Crainic, T. G. (2008), Parallel solution methods for vehicle routing problems, *in* 'The Vehicle Routing Problem: Latest Advances and New Challenges', Springer, pp. 171–198. 70

Crevier, B., Cordeau, J.-F. & Laporte, G. (2007), 'The multi-depot vehicle routing problem with inter-depot routes', *European Journal of Operational Research* **176**(2), 756–773. 18, 19

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

De Cauwer, C., Van Mierlo, J. & Coosemans, T. (2015), 'Energy consumption prediction for electric vehicles based on real-world data', *Energies* **8**(8), 8573–8593. 77

Desaulniers, G., Errico, F., Irnich, S. & Schneider, M. (2014), Exact algorithms for electric vehicle-routing problems with time windows, Technical report, GERAD, G-2014-110. 12, 36, 37, 88

Doerner, K. F., Hartl, R. F. & Lucka, M. (2005), 'A parallel version of the d-ant algorithm for the vehicle routing problem', *Parallel Numerics* **5**, 109–118. 70

Doerner, K. F. & Schmid, V. (2010), *Survey: Matheuristics for Rich Vehicle Routing Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 206–221. 70

Dohn, A., Kolind, E. & Clausen, J. (2009), 'The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach', *Computers & Operations Research* **36**(4), 1145 – 1157. 66

Energy Information Administration (2008), Emissions of greenhouse gases in the united states 2007, Technical report, Center for Entrepreneurship and Technology, University of California, Berkeley. 11

ERDF (2014), 'ERDF étoffe sa flotte de véhicules électriques en Ile-de-France'. url = goo.gl/jpYml9, Last Accessed= 23/06/2016. 11, 65, 66

Erdoğan, S. & Miller-Hooks, E. (2012), 'A green vehicle routing problem', *Transportation Research Part E: Logistics and Transportation Review* **48**(1), 100–114. 7, 12, 17, 18, 19, 22, 27, 28, 29, 41, 89, 90, 91

Felipe, A., Ortuno, M. T., Righini, G. & Tirado, G. (2014), 'A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges', *Transportation Research Part E: Logistics and Transportation Review* **71**, 111 – 128. 12, 18, 19, 25, 27, 28, 36, 49, 54, 89

Feo, T. A. & Resende, M. G. (1995), 'Greedy randomized adaptive search procedures', *Journal of global optimization* **6**(2), 109–133. 71

Fontecha, J., Duque, D., Akhavan-Tabatabaei, R., Rodriguez, J. & Medaglia, A. (2015), Combining maintenance and routing models: the case of a water & sewage utility., *in* 'Optimization Workshop 2015 (NOW 2015)', La Rochelle, France. 18

Foster, B. A. & Ryan, D. M. (1976), 'An integer programming approach to the vehicle scheduling problem', *Operational Research Quarterly* pp. 367–384. 20

Goeke, D., Hof, J. & Schneider, M. (2015), Adaptive variable neighborhood search for the battery swap station location-routing problem with capacitated electric vehicles, Technical report, Technical Report BISOR-01. 12

Goeke, D. & Schneider, M. (2015), 'Routing a mixed fleet of electric and conventional vehicles', *European Journal of Operational Research* **245**(1), 81 – 99. 13, 41, 65, 66

Gómez, A., Mariño, R., Akhavan-Tabatabaei, R., Medaglia, A. L. & Mendoza, J. E. (2015), 'On modeling stochastic travel and service times in vehicle routing', *Transportation Science* **50**(2), 627–641. 18

Gonçalves, F., Cardoso, S. R., Relvas, S. & Barbosa-Póvoa, A. (2011), Optimization of a distribution network using electric vehicles: A vrp problem, *in* 'Proceedings of the IO2011-15 Congresso da associação Portuguesa de Investigação Operacional, Coimbra, Portugal', pp. 18–20. 19

Hall, R. & Partyka, J. (2008), 'On the road to mobility–2008 survey of vehicle routing software spotlights integration with portable phones', *OR/MS Today by Institute for Operations Research and the Management Sciences* . 12

H&E (2016), 'Endesa introducirá 3.000 coches eléctricos en su flota para 2020'. url = http://www.hibridosyelectricos.com/articulo/sector/endesa-introducira-3-000-coches-electricos-flota-2020.html, Last Accessed= 23/06/2016. 11, 65, 66

Heineken (2013), 'Europe's largest electric truck will drive down emissions', http://sustainabilityreport.heineken.com/Reducing-CO2-emissions/Case-studies/Europes-largest-electric-truck-will-drive-down-emissions/index.htm. Last accessed 13/04/2016. 12

Hiermann, G., Puchinger, J., Ropke, S. & Hartl, R. F. (2016), 'The electric fleet size and mix vehicle routing problem with time windows and recharging stations', *European Journal of Operational Research* **252**(3), 995 –1018. 7, 13, 37, 41, 50, 66, 79, 80, 97, 98, 99

Hõimoja, H., Rufer, A., Dziechciaruk, G. & Vezzini, A. (2012), An ultrafast EV charging station demonstrator, *in* 'Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2012 International Symposium on', IEEE, pp. 1390–1395. 9, 35

Juan, A. A., Goentzel, J. & Bektaş, T. (2014), 'Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations?', *Applied Soft Computing* **21**, 84–94. 20

Keskin, M. & Čatay, B. (2016), 'Partial recharge strategies for the electric vehicle routing problem with time windows', *Transportation Research Part C: Emerging Technologies* **65**, 111 – 127. 12, 36, 37

Kleindorfer, P. R., Neboian, A., Roset, A. & Spinler, S. (2012), 'Fleet renewal with electric vehicles at La Poste', *Interfaces* **42**(5), 465–477. 11

Koç, Ç., Bektaş, T., Jabali, O. & Laporte, G. (2016), 'Thirty years of heterogeneous vehicle routing', *European Journal of Operational Research* **249**(1), 1–21. 66

Koč, C. & Karaoglan, I. (2016), 'The green vehicle routing problem: A heuristic based exact solution approach', *Applied Soft Computing* **39**, 154 – 164. 12

Kovacs, A. A., Parragh, S. N., Doerner, K. F. & Hartl, R. F. (2012), 'Adaptive large neighborhood search for service technician routing and scheduling problems', *Journal of Scheduling* **15**(5), 579–600. 66

Lahyani, R., Khemakhem, M. & Semet, F. (2015), 'Rich vehicle routing problems: From a taxonomy to a definition', *European Journal of Operational Research* **241**(1), 1–14. 65

Laporte, G., Nobert, Y. & Desrochers, M. (1985), 'Optimal routing under capacity and distance restrictions', *Operations Research* **33**(5), 1050–1073. 18

Lenstra, J. K. & Kan, A. (1981), 'Complexity of vehicle routing and scheduling problems', *Networks* **11**(2), 221–227. 17

Liao, C.-S., Lu, S.-H. & Shen, Z.-J. M. (2016), 'The electric vehicle touring problem', *Transportation Research Part B: Methodological* **86**, 163 – 180. 38, 50

Liu, R., Yuan, B. & Jiang, Z. (2016), 'Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements', *International Journal of Production Research* p. DOI:10.1080/00207543.2016.1213917. 66, 88

Loaiza (2014), 'Convenio Sura-Celsia, por el bien del medio ambiente', `http://www.sura.com/blogs/autos/convenio-sura-celsia-por-el-medio-ambiente.aspx`. Last accessed 13/04/2016. 12

Lourenço, H. R., Martin, O. C. & Stützle, T. (2010), Iterated local search: Framework and applications, *in* 'Handbook of Metaheuristics', Springer, pp. 363–397. 47, 49

Lozano, L., D. D. & Medaglia, A. L. (2013*a*), 'The pulse algorithm: A java implementation (jpulse)'. **URL:** *http://dspace.uniandes.edu.co:9090/xmlui/handle/1992/1162* 26

Lozano, L. & Medaglia, A. L. (2013*b*), 'On an exact method for the constrained shortest path problem', *Computers & Operations Research* **40**(1), 378–384. 26

Mendoza, J. E., Guéret, C., Hoskins, M., Lobit, H., Pillac, V., Vidal, T. & Vigo, D. (2014), VRP-REP: a vehicle routing community repository, *in* 'Third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog). Oslo (Norway)'. 56, 77

Mendoza, J. E. & Villegas, J. G. (2013), 'A multi-space sampling heuristic for the vehicle routing problem with stochastic demands', *Optimization Letters* **7**(7), 1503–1516. 18, 20, 21

Mladenović, N. & Hansen, P. (1997), 'Variable neighborhood search', *Computers & Operations Research* **24**(11), 1097–1100. 47, 72

Montoya, A., Guéret, C., Mendoza, J. E. & Villegas, J. G. (2015), 'A multi-space sampling heuristic for the green vehicle routing problem', *Transportation Research Part C: Emerging Technologies* **70**, 113 – 128. 12, 50

Muter, I., Cordeau, J.-F. & Laporte, G. (2014), 'A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes', *Transportation Science* **48**(3), 425–441. 19

Pedersen, M. B., Madsen, O. B. & Nielsen, O. A. (2005), Optimization models and solution methods for intermodal transportation, PhD thesis, Technical University of Denmark. 11

Pelletier, S., Jabali, O. & Laporte, G. (2014), Battery electric vehicles for goods distribution: a survey of vehicle technology, market penetration, incentives and practices, Technical report, CIRRELT-2014-43. 20

Pelletier, S., Jabali, O. & Laporte, G. (2016), 'Goods distribution with electric vehicles: Review and research perspectives', *Transportation Science* **50**(1), 3–22. 11

Pelletier, S., Jabali, O., Laporte, G. & Veneroni, M. (2015), Goods distribution with electric vehicles: Battery degradation and behaviour modeling, Technical report, CIRRELT-2015-43. 35

Peugeot (2015), 'Peugeot ion', http://www.peugeot.co.uk/media/peugeot-ion-prices-and-specifications-brochure.pdf. Accessed: 2016-09-12. 77

Pillac, V., Guéret, C. & Medaglia, A. L. (2013), 'A parallel matheuristic for the technician routing and scheduling problem', *Optimization Letters* **7**(7), 1525–1535. 66, 73

Post & Parcel (2014), 'CTT group invests 5M of euros in green fleet', http://postandparcel.info/60290/uncategorized/ctt-group-invests-e5m-in-green-fleet/. Last accessed 13/04/2016. 11

Prins, C. (2004), 'A simple and effective evolutionary algorithm for the vehicle routing problem', *Computers & Operations Research* **31**(12), 1985 – 2002. 22, 49

Prins, C., Lacomme, P. & Prodhon, C. (2014), 'Order-first split-second methods for vehicle routing problems: A review', *Transportation Research Part C: Emerging Technologies* **40**, 179–200. 20

Priselac, M. (2013), 'A frigid fleet: Coca-Cola unveils first electric refrigerated trucks.'. **URL:** *http://www.coca-colacompany.com/stories/a-frigid-fleet-coca-cola-unveils-first-electric-refrigerated-trucks* 12

Quest, C. (2014), 'Carte des bornes de recharge pour véhicules électriques', https://www.data.gouv.fr/fr/reuses/carte-des-bornes-de-recharge-pour-vehicules-electriques. Accessed: 2016-09-12. 77

Ramirez (2015), 'Bimbo inagura el segundo centro de ventas ecologico en mexico', http://www.milenio.com/negocios/Centro_de_Ventas_Ecologico-bimbo-Centro_de_Ventas_Ecologico_bimbo_0_633536930.html. Last accessed 13/04/2016. 12

Reimann, M., Doerner, K. & Hartl, R. F. (2004), 'D-ants: Savings based ants divide and conquer the vehicle routing problem', *Computers & Operations Research* **31**(4), 563–591. 70

Renault (2014), 'Renault ZE', http://www.renault.re/ze/brochure/KANGOOZE.pdf. Accessed: 2016-09-12. 77

Resende, M. G. (2008), 'Metaheuristic hybridization with greedy randomized adaptive search procedures', *Tutorials in Operations Research* pp. 295–319. 72

Rosenkrantz, D., Stearns, R. & Lewis, P. (1974), Approximate algorithms for the traveling salesperson problem, *in* 'Switching and Automata Theory, 1974., IEEE Conference Record of 15th Annual Symposium on', pp. 33–42. 47

Rosing, K. & ReVelle, C. (1997), 'Heuristic concentration: Two stage solution construction', *European Journal of Operational Research* **97**(1), 75 – 86. 47

Rousseau, L.-M., Gendreau, M. & Pesant, G. (2002), 'Using constraint-based operators to solve the vehicle routing problem with time windows', *Journal of heuristics* **8**(1), 43–58. 72

Sassi, O., Cherif-Khettaf, W. & Oulamara, A. (2015), Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles, *in* H. A. Le Thi, T. Pham Dinh & N. T. Nguyen, eds,

'Modelling, Computation and Optimization in Information Systems and Management Sciences', Vol. 359 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, pp. 57–68. 13, 66

Sassi, O., Cherif, W. R. & Oulamara, A. (2014), Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs, Technical report, hal-01083966. 19, 36, 41

Savelsbergh, M. W. (1992), 'The vehicle routing problem with time windows: Minimizing route duration', *ORSA journal on computing* **4**(2), 146–154. 72

Schiffer, M. & Walther, G. (2015), The electric location routing problem with time windows and partial recharging, Technical report, RWTH Aachen University. 36

Schneider, M., Stenger, A. & Goeke, D. (2014), 'The electric vehicle-routing problem with time windows and recharging stations', *Transportation Science* **48**(4), 500–520. 12, 17, 18, 19, 25, 27, 28, 37, 41, 89

Schneider, M., Stenger, A. & Hof, J. (2015), 'An adaptive VNS algorithm for vehicle routing problems with intermediate stops', *OR Spectrum* **37**(2), 353–387. 12, 18, 19, 25, 27, 28, 89

Solomon, M. M. (1987), 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research* **35**(2), 254–265. 72

Suzuki, Y. (2014), 'A variable-reduction technique for the fixed-route vehicle-refueling problem', *Computers & Industrial Engineering* **67**, 204–215. 50

Sweda, T. M., Dolinskaya, I. S. & Klabjan, D. (2014), 'Optimal recharging policies for electric vehicles', *Transportation Science, forthcoming* p. DOI:10.1287/trsc.2015.0638. 50

Taillard, É. (1993), 'Parallel iterative search methods for vehicle routing problems', *Networks* **23**(8), 661–673. 70

Taillard, É. D. (1999), 'A heuristic column generation method for the heterogeneous fleet VRP', *RAIRO-Operations Research* **33**(1), 1–14. 70

Tarantilis, C. D., Zachariadis, E. E. & Kiranoudis, C. T. (2008), 'A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities', *INFORMS Journal on Computing* **20**(1), 154–168. 18

Thompson, P. M. & Psaraftis, H. N. (1993), 'Cyclic transfer algorithm for multivehicle routing and scheduling problems', *Operations research* **41**(5), 935–946. 19

Toth, P. & Vigo, D. (2001), An overview of vehicle routing problems, *in* 'The vehicle routing problem', Society for Industrial and Applied Mathematics, pp. 1–26. 12

Toth, P. & Vigo, D. (2014), *Vehicle routing: problems, methods, and applications*, Vol. 18, Siam. 12

Tseng, Y.-y., Yue, W. L., Taylor, M. A. et al. (2005), The role of transportation in logistics chain, Eastern Asia Society for Transportation Studies. 11

Uhrig, M., Weiß, L., Suriyah, M. & Leibfried, T. (2015), E-mobility in car parks–guidelines for charging infrastructure expansion planning and operation based on stochastic simulations, *in* 'the 28th International Electric Vehicle Symposium and Exhibition, KINTEX, Korea'. 36, 37, 41, 56, 77

UPS (2008), 'UPS hybrid electric vehicle fleet.', `http://www.pressroom.ups.com/HEV`. Last accessed 05/14/2014. 11

Vidal, T., Maculan, N., Ochi, L. S. & Vaz Penna, P. H. (2015), 'Large neighborhoods with implicit customer selection for vehicle routing problems with profits', *Transportation Science* **50**(2), 720–734. 29

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L. & Velasco, N. (2010), 'Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots', *Engineering Applications of Artificial Intelligence* **23**(5), 780–794. 25

Wang, H., Liu, Y., Fu, H. & Li, G. (2013), 'Estimation of state of charge of batteries for electric vehicles', *International Journal of Control and Automation* **6**(2), 185–194. 36

Xu, J. & Chiu, S. Y. (2001), 'Effective heuristic procedures for a field technician scheduling problem', *Journal of Heuristics* **7**(5), 495–509. 66

Zamorano, E. & Stolletz, R. (2016), 'Branch-and-price approaches for the multiperiod technician routing and scheduling problem', *European Journal of Operational Research* p. DOI:10.1016/j.ejor.2016.06.058. 66

Zündorf, T. (2014), Electric vehicle routing with realistic recharging models, Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany. 36, 38

l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

université angers

Thèse de Doctorat

Alejandro MONTOYA

Problèmes de tournées de véhicules électriques : modèles et méthodes de résolution

Electric Vehicle Routing Problems: models and solution approaches

Résumé

Etant donné leur faible impact environnemental, l'utilisation des véhicules électriques dans les activités de service a beaucoup augmenté depuis quelques années. Cependant, leur déploiement est freiné par des contraintes techniques telles qu'une autonomie limitée et de longs temps de charge des batteries. La prise en compte de ces contraintes a mené à l'apparition de nouveaux problèmes de tournées de véhicules pour lesquels, en plus d'organiser les tournées, il faut décider où et de combien charger les batteries. Dans cette thèse nous nous intéressons à ces problèmes au travers de quatre études. La première concerne le développement d'une metaheuristique en deux phases simple mais performante pour résoudre un problème particulier appelé "Green VRP". Dans la seconde, nous nous concentrons sur la modélisation d'un aspect essentiel dans ces problèmes : le processus de chargement des batteries. Nous étudions différentes stratégies pour modéliser ce processus et montrons l'importance de considérer la nature non linéaire des fonctions de chargement. Dans la troisième étude nous proposons une recherche locale itérative pour résoudre des problèmes avec des fonctions de chargement non linéaires. Nous introduisons un voisinage dédié aux décisions de chargement basé sur un nouveau problème de chargement sur une tournée fixée. Dans la dernière étude, nous traitons un problème réel de tournées de techniciens avec des véhicules classiques et électriques. Ce problème est résolu par une metaheuristique qui décompose le problème en plusieurs sous-problèmes plus simples résolus en parallèle, puis qui assemble des parties des solutions trouvées pour construire la solution finale.

Mots clés

Problèmes de tournées de véhicules électriques ; Problèmes de tournées de techniciens ; Green vehicle routing problems ; Problème de chargement sur une tournée fixée ; Fonctions de chargement non linéaires ; Méta-heuristique ; Matheuristique, Recherche locale itérative

Abstract

Electric vehicles (EVs) are one of the most promising technologies to reduce the greenhouse gas emissions. For this reason, the use of EVs in service operations has dramatically increased in recent years. Despite their environmental benefits, EVs still face technical constraints such as short autonomy and long charging times. Taking into account these constraints when planning EV operations leads to a new breed of vehicle routing problems (VRPs), known as electric VRPs (eVRPs). In addition, to the standard routing decisions, eVRPs are concerned with charging decisions: where and how much to charge. In this Ph. D thesis, we address eVRPs through 4 different studies. In the first study, we tackle the green vehicle routing problem. To solve the problem, we propose a simple, yet effective, two-phase matheuristic. In the second study, we focus a key modelling aspects in eVRPs: the battery charging process. We study different strategies to model this process and show the importance of considering the nonlinear nature of the battery charging functions. In the third study, we propose an iterated local search to tackle eVRP with non-linear charging functions. We introduce a particular local search operator for the charging decisions based on a new fixed-route charging problem. The fourth and last study considers a real technician routing problem with conventional and electric vehicles (TRP-CEV). To tackle this problem, we propose a parallel matheuristic that decomposes the problem into a set of easier-to-solve subproblems that are solved in parallel processors. Then the approach uses parts of the solutions found to the subproblems to assemble final solution to the TRP-CEV.

Key Words

Electric vehicle routing problems; Technician routing problems; Green vehicle routing problems; Fixed-route vehicle charging problems; Non linear charging functions; Metaheuristic; Matheuristic; Iterated local search