The Vehicle Scheduling Problem of Electric Buses

Maria Leonor Bento Ayres Pereira

Master's Dissertation

Supervisor: Prof. Dr. Maria Teresa Galvão Dias

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Integrated Master in Mechanical Engineering

2019-07-01

Abstract

Despite its evident advantages, *electric vehicles* face two major limitations: its lower autonomy and long charging time. Therefore, its adoption for public transportation forces the study of new approaches in the scheduling method in order to adapt the common solutions to its new constraints.

The present paper aims to answer this problem with a multi-depot and charging station model. Developed for a homogeneous fleet, the mathematical formulation allows the determination of an optimized vehicle scheduling, minimizing the total cost involved, such as vehicle purchase, energy consumption and operational costs.

As for its experimental phase, *CPLEX Optimization Studio* software is adopted, using *Optimization Programming Language* (OPL), requiring some coding adaptions. Effective for small instances, the algorithm becomes limited for larger inputs, increasing substantially its computational time.

Real data, provided by STCP, is tested, leading to additional script modifications. Alterations based on the adaption of the code to this specific scenario help improve its efficiency, decreasing the waiting time.

A comparison between electric and non-electric vehicles scheduling is carried out, contrasting its percentage of operational time. Due to the large amount of time it takes to charge its battery and the visits to charging stations, a decline of the vehicle useful time is observed and compensated by the purchased of a higher number of vehicles.

Both initial stated limitations of electric vehicles are also tested. Results demonstrate that the influence of the charging time is higher than the level of autonomy of a vehicle, proving a greater responsibility of the first one in the deterioration that electric vehicles brings to the scheduling problem. Therefore, it is concluded that a decrease of the charging time is more rewarding than an improvement of the vehicle's autonomy.

Resumo

Apesar das vantagens, os *veículos elétricos* apresentam duas principais limitações: baixa autonomia e elevado tempo de carregamento. Assim, a adoção dos mesmos para transporte público implica recorrer a novas abordagens para o método de agendamento dos veículos, visando a adaptação das soluções correntes às novas restrições.

O presente trabalho tem como objetivo responder a este problema, com recurso a um modelo com múltipos depósitos e estações de carregamento. Desenvolvido para uma frota homogénea, a formulação matemática descrita permite a determinação de um agendamento de veículos otimizado, minimizando o custo total envolvido, tais como a aquisição dos veículos, a energia consumida e os custos operacionais.

Na fase experimental, o software utilizado foi o *CPLEX Optimization Studio* e a linguagem foi *Optimization Programming Language* (OPL), exigindo adaptações para o código. Eficiente para exemplos pequenos, o algoritmo torna-se limitado para maiores cenários, aumentando consideravelmente o tempo computacional.

Dados reais, fornecidos pela STCP, são testados, introduzindo novas modificações ao algoritmo. Estas alterações, baseadas na adaptação do código para este cenário específico, permitem melhor a eficiência do mesmo, diminuindo consideravelmente o tempo computacional.

Uma comparação entre veículos elétricos e a gás é levada a cabo, contrastando a sua percentagem de tempo operacional. Devido ao grande tempo de carregamento e às visitas às estações de carregamento, constata-se um declínio no tempo útil do veículo, compensado com a aquisição de um maior número de veículos.

As duas limitações apresentadas inicialmente são também testadas. Os resultados demonstram que a influência do tempo de carregamento é maior do que a autonomia dos veículos, provando uma maior responsibilidade no que toca à deterioração que os veículos elétricos acarretam. Assim, conclui-se que a diminuição do tempo de carregamento é mais benéfico que uma melhoria na autonomia dos veículos elétricos.

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Teresa Galvão Dias, for all the knowledge she passed on and the availability, promptitude and clarity with which she answered all my doubts. But foremost by the vow of confidence and space she gave me, letting me follow my own ideas.

I would also like to thank *Caetanobus - Fabricação de Carroçarias SA* for the succinct and detailed explanation about electric buses.

To Marina, my official editor on the other side of the ocean, for having read and reread this thesis more often than anyone else, commenting and correcting each detail carefully. Your patience and willingness to help others make me proud to be called your sister.

To José, who always have been an example to me, for inspiring me to do my best in everything I do and enjoying it while I do it. Thank you for always believing in my capacities more than I do.

To Rosário, for always solving my problems promptly, making them seem small, whether it was about electric vehicles or any other subject. You are a real life saver.

To all my friends, the old ones and the recent ones, for listening to all my dramas over and over again until they seem insignificantly and for the laughs that followed it. I could not have done it without you.

Finally, to my Family, the perfect *trio-ish*, for always being present in the backstage of all the stages of my life. In particular, to my Parents and brothers Filipa, Xavier and Carmo, for putting up with me on the good and bad days, making it always feel good to return home. I can firmly say I have the best ones with me. I could never thank you enough.

Leonor

"Learn the most you can - and have fun doing it."

To my brother José.

Contents

1 Introduction **1**
1.1 Problem Overview . . . 1
1.1.1 The Charging Problem . . . 1
1.2 Context . . . 2
1.3 Transit Scheduling . . . 3
1.3.1 Vehicle Scheduling Problem . . . 3
1.4 Structure of the Document . . . 4

2 Literature Review **5**
2.1 Vehicle Scheduling Problem . . . 5
2.1.1 Single Depot Vehicle Scheduling Problem . . . 5
2.1.2 Multiple Depot Vehicle Scheduling Problem . . . 6
2.1.3 Electric Vehicle Scheduling Problem . . . 6
2.2 Solutions . . . 7
2.2.1 Genetic Algorithm . . . 7
2.2.2 Tabu Search . . . 8
2.2.3 Variable Neighborhood Search . . . 9
2.2.4 Simulated Annealing . . . 9
2.2.5 Ant Colony Algorithm . . . 10
2.2.6 Branch-and-Bound Algorithm . . . 11
2.2.7 Branch-and-Cut Algorithm . . . 11
2.2.8 Branch-and-Price Algorithm . . . 11
2.2.9 Column Generation . . . 11
2.2.10 Lagrangian Relaxation . . . 12
2.2.11 Large Neighborhood Search . . . 12

3 Electric Vehicle Scheduling Formulation **13**
3.1 Formal problem description . . . 13
3.2 Assumptions . . . 16
3.3 Variables . . . 16
3.3.1 Sets and Indexes . . . 16
3.3.2 Parameters . . . 16
3.3.3 Auxiliary variables . . . 17
3.3.4 Decision variables . . . 17
3.4 Objective Function . . . 18
3.5 Constraints . . . 18

4 Solution Approach **23**
4.1 Software Application 23
4.1.1 *IBM ILOG CPLEX Optimization Studio* software 23
4.1.2 Optimization Programming Language 23
4.2 Programming 23
4.3 Data preparation 28

5 Results **29**
5.1 STCP 29
5.2 Provided Data 29
5.2.1 Description of the Provided Data 29
5.2.2 Current Solution 30
5.3 Experiments 33
5.3.1 Data Preparation 33
5.3.2 Input Parameters 35
5.3.3 Settings and Preprocessing 37
5.3.4 Results 37
5.3.5 Discussion 39
5.3.6 Performance 43

6 Conclusions and Future Work **45**

A Detailed Information Regarding the Service Trips **49**

B CPLEX results **53**

C Evolution of the energy level per vehicle **61**

Acronyms and Symbols

AGA	Multiple Ant Colony Algorithm
B&B	Branch-and-Bound
DH	Deadheading
EV	Electric Vehicles
eVSP	Electric Vehicle Scheduling Problem
GA	Genetic Algorithm
GGA	Grouping Genetic ALgorithm
LNS	Large Neighborhood Search
LS	Local Search
MDVSP	Multiple Depot Vehicle Scheduling
NP-hard	Non-deterministic polynomial-time hard
OPL	Optimization Programming Language
RMP	Restricted Master Problem
SDVSP	Single Depot Vehicle Scheduling
TCO	Total Cost of Ownership
TS	Tabu Search
VNS	Variable Neighborhood Search
VSP	Vehicle Scheduling Problem
VSPRFTC	Vehicle Scheduling Problem with Route and Fueling Time Constraints

List of Figures

1.1 Representative graph of the problem 4

3.1 Representative graph of the problem 13
3.2 Representation of charging events at a single charging station 15
3.3 Exemplification of the parameter p in a graph 15

4.1 Possible scenarios between two service trips 24

5.1 Schedule of *Vehicle 1* by STCP 31
5.2 Schedule of *Vehicle 2* by STCP 31
5.3 Schedule of *Vehicle 3* by STCP 32
5.4 Schedule of *Vehicle 4* by STCP 32
5.5 Schedule of *Vehicle 5* by STCP 32
5.6 Schedule of *Vehicle 6* by STCP 32
5.7 Schedule of *Vehicle 1* of *Scenario 1* 39
5.8 Schedule of *Vehicle 2* of *Scenario 1* 39
5.9 Schedule of *Vehicle 3* of *Scenario 1* 39
5.10 Schedule of *Vehicle 6* of *Scenario 1* 40
5.11 Schedule of *Vehicle 1* of *Scenario 2* 40
5.12 Schedule of *Vehicle 2* of *Scenario 2* 40
5.13 Schedule of *Vehicle 3* of *Scenario 2* 40

C.1 Evolution of the energy level on the *Scenario 1* 61
C.2 Evolution of the energy level on the *Scenario 2* 62

List of Tables

5.1 Data size provided by STCP 29
5.2 Information about the nodes provided by STCP 30
5.3 Service trips grouped in 6 different trajectories 30
5.4 Current solution of STCP by service trips 31
5.5 Current solution of STCP by trajectories 33
5.6 Average duration of each trajectory (in minutes) 33
5.7 Average distance of each trajectory (in km) 34
5.8 Average energy consumption of each trajectory (in kWh) 34
5.9 Average duration of each DH trip between trajectories (in minutes) 35
5.10 Average energy consumption of each DH trip between trajectories (in kWh) . . . 35
5.11 Adapted data size 35
5.12 Input parameters regarding the data size 36
5.13 Input Parameters regarding Costs 36
5.14 Input parameters regarding the formulation 37
5.15 CPLEX Solution for *Scenario 1* 38
5.16 CPLEX Solution for *Scenario 2* 38
5.17 Objective function values and computational time for both scenarios 38
5.18 Time per task on the STCP solution for *Scenario 1* (in minutes) 41
5.19 Time per task on the CPLEX solution for *Scenario 1* (in minutes) 41
5.20 Time per task on the STCP solution for *Scenario 2* (in minutes) 42
5.21 Time per task on the CPLEX solution for *Scenario 2* (in minutes) 42
5.22 Results of the tests to the influence of the two limitations of EV 43
5.23 CPLEX results of the *Scenario 1* with $k = 4$ 43
5.24 CPLEX solution of the *Scenario 1* with $k = 4$ 44

A.1 Detailed Information Regarding the Service Trips 49

B.1 Values of x_{ij}^{kp} for *Scenario 1* 53
B.2 Values of E_i^{kp} for *Scenario 1* 55
B.3 Values of x_{ij}^{kp} for *Scenario 2* 56
B.4 Values of E_i^{kp} for *Scenario 2* 58

Chapter 1

Introduction

1.1 Problem Overview

Air pollution represents a serious threat to our health and planet, being road transport the most blamed cause regarding urban pollution (V. Franco, 2013). Therefore, the growing urgency to reduce or even eliminate emissions and save resources led to the development of *electric vehicles* (EV). These vehicles are powered by regenerative energy sources and have the advantageous capacity of recover some of their kinetic and potential energy during deceleration phases. This last characteristic allows them to recover around 40% of their consumption, according to the Portuguese company *Caetanobus - Fabricação de Carroçarias SA*.

Given that the replacement of vehicles requires a large investment, the need to low operational costs represents an important goal. Costs reduction is only possible through the optimization of the vehicles' assignment, leading to a minimization of the fleet size, energy consumption and operational costs.

Despite their advantages over other types of vehicles, EV face two major limitations: its short range and its long charging time. These restrictions force the usual modelling approaches to change, adapting it to the new constraints. Because of its lower autonomy, some electric vehicles need to be exposed to multiple battery charges instead of the usual one time per day refuel, in the case of diesel vehicles. These charges take place at designated charging stations which may be located at the depots or specialized areas. Due to its long charging time, it has to be inserted in the schedule of each vehicle.

1.1.1 The Charging Problem

There are two alternatives on how to recharge an EV's battery: either through replacement of the wasted one for a charged one, which was loaded previously and outside the bus, or by charging it inside the vehicle.

The first permits the charging of the battery during off-peak hours, allowing it to do it slower, which increases the battery lifespan, and cheaper, due to the lower cost of energy on this periods.

Furthermore, it offers shorter charging times for the vehicles as the battery is already charged at its maximum at the time of the swap (Zheng et al. (2014)).

However, even though battery replacement solves the charging time issue, it entails some disadvantages. For the swapping batteries method to be feasible, standardization between manufacturers has to occur which limits the innovation and flexibility. Also, swapping stations lead to higher investment as it has to storage all spare batteries and charge them. China is one of the countries where this method was adopted, with better results for fleets than for personal vehicles (Z. Wan (2015)).

As for battery charging, there exist multiple solutions for urban buses. In some cities, the charging process is assigned at the beginning and ending of each service trip (rapid charging, 5-10 minutes), allowing vehicles to charge the amount of energy needed for the next service trip, or even at each stop of the route (ultra-rapid charging, 20 seconds). These options are called *opportunity charging*. On the other hand, the *overnight charging* is longer (around 3 hours), in exchange for higher autonomy. These vehicles charge mainly overnight at depots or other charging stations (Transport&Environment (2018)).

Each of these approaches answers different problems. Also, despite the existence of different battery capacities, the higher the capacity, the heavier the battery, leading to a higher consumption. Furthermore, the higher the number of batteries, the smaller the space for people accommodation. Therefore, the trade-off make it difficult to label a solution as the best one.

In this paper, it is discussed the overnight charging approach since it is the one adopted in Portugal.

In order to increase batteries lifespan, the charges and discharges can not be completed, avoiding extreme operational points. These numbers depend on the manufacture, but normally it cannot go lower than 20% or higher than 80%, according to *Caetanobus - Fabricação de Carroçarias SA*.

The existence of multiple depots, allied to the need to allocate each vehicle to one and just one depot, complicates the problem in a way that many previous authors solved already. Nevertheless, the possibility of charging in different depots during the day increases the complexity of the problem and has not been deeply studied.

1.2 Context

In Porto, there are currently 15 buses being experimented in the city with 15 chargers at one single depot. Due to the not optimized location of these chargers, the vehicles are forced to spend most of their operational time charging, being able to do only a small fraction of the morning and afternoon schedules.

Furthermore, the scheduling does not take into consideration the possibility of purchase an heterogeneous fleet. Instead of finding the most suitable bus type for each chain of service trips, it adopted a singular model that satisfies every specification, leading to a less optimal solution.

The present work aims to develop a solution for the complex trade-off that results in a minimization of costs, helping at the time of decision, presenting a new approach to the problem. The heterogeneous fleet issue is not, however, contemplated.

1.3 Transit Scheduling

Transit Scheduling is developed in four connected phases. It starts by designing routes, answering to the Vehicle Routing Problem, followed by the creation of timetables. The third phase consists on allocating vehicles to trips, named Vehicle Scheduling, and it is proceeded by the assignment of drivers (Crew Scheduling) (Bishop, 2006). Ideally, the four steps would be worked out together to ensure the highest level of optimization. However, that would lead to a too complex problem, causing the researchers to normally opt for one of them. This paper will focus on the third phase.

1.3.1 Vehicle Scheduling Problem

The *Vehicle Scheduling Problem* (VSP) was first discussed in 1959 by Dantzig and Ramers (C. Xu, 2010), aiming to plan all the regular operation in passenger service, named *service trips*, of a given timetable.

Each service trip is characterized by its starting and ending locations, called *terminus*, and its arrival and departure times. Providing the travel times and distances between each one of these terminus, the VSP finds an assignment of the service trips to vehicles, ensuring that:

- Each service trip is covered by one and only one vehicle;
- Each vehicle has a feasible chain of trips;
- The total cost is minimized (S. Bunte, 2009).

The final result is the allocation of a created schedule to each different vehicle. Thus, a *schedule* can be defined as a set of service trips that a certain vehicle has to secure and an allocated depot to that same vehicle in an arranged sequence. In order to obey the basic rules, the first and last element have to be the allocated depot to that vehicle and the vehicle has to ensure at least one service trip. For a schedule to be considered *feasible*, the vehicle has to be able to accomplish it without running out of fuel and to obey to the service trips demands such as starting and ending times (J. Adler, 2017).

In order to minimize the number of vehicles required, empty trips may be planned between an ending location and the starting one of the following trip. These empty trips also occur between a depot and a service trip and are called *deadheading trips* (DH trips). The insertion of a DH trip assumes that the cost of including one more vehicle is significantly higher than any additional operational cost that this auxiliary trip may entail (Bishop, 2006).

In the Figure 1.1, a visual representation of these concepts is described. The blue nodes represent each terminus of a service trip, being i and i' its starting and ending point, respectively. Each red dashed arrow symbolizes the trajectory of a service trip i and the full arrows between two nodes represent a deadheading trip connecting both locations. The yellow and brown color distinguish each schedule of both vehicles.

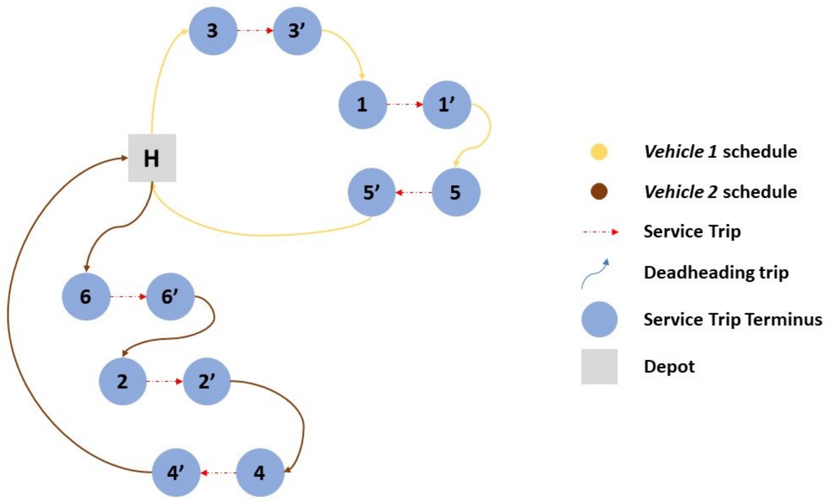

Figure 1.1: Representative graph of the problem

The allocation of service trips to a schedule may force the vehicle to wait at a determined location before it starts the next task. This waiting time is called *idle time* and has to be taken into account when minimizing the operational costs.

1.4 Structure of the Document

The present paper is organized in 6 chapters. After the initial chapter of introduction, Chapter 2 contextualizes the present work in the current state of the art, describing the models and solutions commonly applied. Chapter 3 presents a mathematical formulation of the problem. It describes the variables, parameters, the objective function and the constraints to which it is subjected. In Chapter 4, a detailed process of the development of the script is defined, characterizing its features and adaptations. Chapter 5 converts the theory into practice, dealing with real data and analyzing results. Finally, some highlights are stated in Chapter 6 where the main outcomes are described and future work is suggested.

Chapter 2

Literature Review

In this chapter, it is exposed all the development and knowledge achieved until today convenient for the better understanding of the following topics. Therefore, it is characterized the concept of *Vehicle Scheduling Problem* in all its extensions. Additionally, some major models and formulations usually used to describe the solution, as well as some methods to solve it, are described.

2.1 Vehicle Scheduling Problem

Over the years, numerous extensions for the VSP have been developed with additional requirements, such as multiple depots, heterogeneous fleet, small variations on the departure times, among others (S. Kulkarni, 2018).

2.1.1 Single Depot Vehicle Scheduling Problem

The simplest case of a VSP is the *Single Depot Vehicle Scheduling Problem* (SDVSP) which has, as the name itself suggests, only one depot where all vehicles should start and end its schedules. The solution for a SDVSP can be reached in a polynomial time through many developed algorithms, modelled as the Minimal Decomposition Model, the Assignment Model, the Transportation Model and the Network Flow Model (S. Bunte, 2009).

Xu (2010) presented one solution to this problem, based on a *Genetic Algorithm*. Its model suits homogeneous fleets scenarios with only one depot and a maximum number of vehicles, establishing a bi-objective function. Through a minimization of the number of the vehicles needed and the distance they have to run, this formulation has the additional feature of contemplating the vehicle capacity. Even though this is not relevant for the present study, its principles are similar and help to understand the basic of vehicle scheduling (C. Xu, 2010).

2.1.2 Multiple Depot Vehicle Scheduling Problem

Another case of a VSP is the one with different locations for starting and ending vehicles' chain of trips, denominated *Multiple Depot Vehicle Scheduling Problem* (MDVSP). It requires a new constraint that ensures that all vehicles return to its starting depot at the end of its task. The existence of multiple depots transforms it into a NP-hard problem, just like the possibility of a heterogeneous fleet. For the MDVSP, there are three major modelling approaches: single commodity, multi-commodity ("in which vehicles from different depots are considered as different commodities") and set partitioning model (S. Bunte, 2009).

For small to medium sized instance, exact solutions approaches can be used. However, for larger instances, and since the MDVSP is a NP-hard, it is normally applied heuristics (S. Kulkarni, 2018).

A. Pepin (2008) compared five different heuristics to solve the multiple depot VSP - two as an integer multi-commodity formulation and the others as a set partitioning type formulation. The fist formulation describes the problem as a time-space network, being each service trip symbolized by a node and each possible deadheading trip by an arc. Its goal is therefore focused on the minimization of the costs regarding the arcs assignment. The second model represents each possible schedule as a decision variable, minimizing, once again, its total cost. In this study, it is proposed a *Branch-and-Cut* method, based on a time-space network, using *CPLEX MIP* solver, for the first formulation, as well as a *Lagrangian* heuristic relying on a Lagrangian relaxation. Another common heuristic was also recommended for the second type of formulation denominated *Truncated Column Generation* which decomposes the problem into a restriction master problem (RMP) and a subproblem per depot. A metaheuristic *Large Neighborhood Search* (LNS) was also proposed which destroys a part of the current solution and reoptimizes it again, in order to find a better solution. Finally, it presented a solution based on a *Tabu Search* metaheuristic, one of the most famous local search technique. These procedures analyze the neighbours of each solution, searching for an improved one. However, *Tabu Search* decreases the tendency to be stuck in a local optimal solution by allowing worse moves whenever an improved one does not exist and by discouraging coming back to already observed solutions (A. Pepin, 2008).

2.1.3 Electric Vehicle Scheduling Problem

The subject approached in this paper is an extension of the VSP for electric vehicles - *Electric Vehicle Scheduling Problem* (eVSP). The goal changes from searching for the minimum or faster path to optimizing the consumption and charging instances due to the major limitation of an EV which is its lower driving range.

In the eVSP, a *schedule* also includes the necessary fuel stations, along with the service trips and the depot, which can not be visited two in a row. In a *feasible* schedule, the vehicle has to be able to visit the entire sequence without running out of battery (J. Adler, 2017).

H. Wang (2007) formulated the eVSP as a VSP with route and fueling time constraints (VSPRFTC), solving it through a *Multiple Ant Colony Algorithm* (AGA). This model minimizes a multiple objective function that contemplates the number of vehicles required and the deadhead time.

Z. Chao (2013) developed an approach in which battery replacement is applied, additionally calculating the optimal number of standby batteries. The problem was formulated as a multi-objective model and solved through a variation of the *Genetic Algorithm*.

T. Paul (2014) studied the possibility of a mixed fleet composed by electric and diesel vehicles, using a *k-Greedy Algorithm*. Here, each type of vehicles is fixed, being that higher priority is given to the electric ones. The difficulty of this study relies in the estimation of the charging amount as a consequence of the remaining battery in an electric vehicle.

Another solution for the MDVSP is developed by J. Adler (2017), applying a *Branch-and-Price algorithm*. It focus not only on time constraints, but also on the charging problem. Contrary to many previous authors, it studies the possibility of refueling in the middle of a schedule, between service trips, not forcing each vehicle to charge at its depot, but allowing it to do it at any other. It uses a combination of the *Branch-and-Bound algorithm* with *Column Generation* called *Branch-and-Price*. Each main variable represents a feasible schedule (set of trips), instead of a single trip. However, it does not consider the possibility of having a limited number of chargers at a charging station which raises the issue of guaranteeing that a limited number of vehicles can charge at the same time. Its initial assumption of instantaneous charges is too optimist since it is one of the biggest limitation of an eVSP. It also proposes a heuristic for faster results.

M. Rogge (2018) modelled this problem relying on a *Grouping Genetic Algorithm* (GGA) using a multi-objective model, minimizing the *Total Cost of Ownership* (TCO) and the number of chargers needed. TCO was defined as the sum of the initial vehicles' investment, operational costs, energy consumption costs and charging infrastructures investment. This approach has its significant interest due to the new concept of chargers optimization, defining it as an output instead of an input. However, it does not consider the possibility of multiple depots.

2.2 Solutions

In this section, the main solutions to the models previously described are explained with more detail.

2.2.1 Genetic Algorithm

Traditional optimization methods can be divided into two categories - direct or gradient-based. The first one uses only the Objective Function and the constraints, while the second one also takes advantages of its derivatives. However, these traditional processes have some limitations such as the tendency to be stuck in a suboptimal solution and the dependence of the efficiency in many factors like the initial chosen solution or the specific type of problem.

Genetic Algorithm (GA) is a search and optimization tool, first introduced by John Holland, developed to answer these barriers. It starts by transforming parameters values into binary strings. The length of the strings define the parameters precision and it can be different for each one. Also, it is allowed to take positive or negative values. Unlike the classical optimization methods, GA starts with a set of solution called *population* instead of the random initial solution. Every solution of the population is then evaluated by assigning it a *fitness* value, normally the Objective Function value, until the termination criteria is reached. If this specification is not fulfilled, three different operators are applied to alter the population.

- *Reproduction Operator*: good solutions, normally the ones with fitness values above average, are identified and copied, replacing the eliminated bad solutions.
- *Crossover Operator*: whenever the crossover operator stops being capable of creating new solutions, reproduction operator starts its task. It chooses two random solution from the new population (after reproduction phase), called *parent solutions*, and divides them at a arbitrary spot. Then, it swaps the same part of each one, creating two new, and hopefully better, solutions, named *children solutions*. To avoid loosing all good strings from the reproduction, not all solutions are subjected to crossover, being directly copied to the new population.
- *Mutation Operator*: in order to guarantee the existence of diversity in the population, this phase changes a string locally to create a better solution.

Genetic Algorithm is built on the idea that created bad solutions will be deleted by the reproduction operator, while good ones will be highlighted (Deb, 1999).

2.2.2 Tabu Search

A *Local Search* (LS) procedure starts from a initial feasible solution and improves it through series of iterative modifications (or moves). These moves change the current solution slightly at each iteration and stop whenever a local optimum is reached. Because it stops at local optimum, the final result obtained is normally a poor and mediocre result.

Developed by Fred Glover in 1986, *Tabu Search* (TS) is an improved local search method. It avoids LS to be trapped in a local optima by permitting non-improving moves whenever it finds a final solution. The use of *tabu lists* allows TS to have a memory which disables the possibility of cycling to already visited solutions. Tabu lists are a short-term memories with limited capacity (usually fixed) that record the recent activity of the search.

Tabu Search is based in two concepts. *Search Space* is the set of all possible solutions that can be analyzed during the process - not always necessarily or advisable limited to feasible solutions. *Neighborhood Structure* is a subset of the Search Space which includes all the possible solutions generated from the current one through viable moves.

Tabus can sometimes be too restricted, forbidding modifications with no risk of cycling or cause the stagnation of the search. *Aspiration criteria* cancels tabus whenever necessary, for example, if it results in a solution with better objective values (D. Henderson, 2003).

2.2.3 Variable Neighborhood Search

Based on Local Search approaches, *Variable Neighborhood Search* (VNS) is a method with constant change of neighborhood within the local search. Unlike the common LS methods, VNS seeks increasingly farther neighborhood of the current solution, instead of tracing a trajectory. It then changes to an *mandatory improved* solution, allowing it to keep convenient attributes of the current solution (for example, the ones which are already at their best objective value). This can be used to find auspicious neighborhood solutions, followed by the search for its local optima, employing local search routine (P. Hansen, 2001).

2.2.4 Simulated Annealing

Simulated Annealing was inspired by the annealing process of crystalline solids, a physical method that consists of heating a solid to then let it cool slowly until it reaches the most regular crystal lattice configuration, without defects and in the low energy ground state. In order to avoid frozen defects, the cooling stage has to be sufficiently slow to allow the solid to reach *thermal equilibrium* for each temperature *T*, preventing the development of an amorphous structure. In that phase, the probability of having an energy level *E* is given by *Boltzmann distribution*, defining the possibility of reaching equilibrium for each temperature *T*:

$$Pr\{\mathrm{E} = E\} = \frac{1}{Z(T)}\, exp(-\frac{E}{k_B T})$$

Where *Z(T)* is the *Normalization Factor*, defined as a partition function, and k_B is the *Boltzmann constant*. The component $exp(-\frac{E}{k_B T})$ represents the *Boltzmann Factor*.

In order to simulate the development of the solid reaching the thermal equilibrium for a fixed value of temperature *T*, a Monte Carlo method was proposed where the current particle arrangement of a solid is perturbed by a small arbitrary disturbance, causing a small displacement of a random particle. If the structure's energy suffers a decrease, that is, if the variation between the current state and the slightly disturbed one, $\triangle E$, is negative, then the new one is accepted and the procedure proceeds with that one. Otherwise, that is, if $\triangle E \geq 0$, the probability of acceptance of the new state is calculated:

$$exp(-\frac{\triangle E}{k_B T})$$

.

The acceptance principle is called *Metropolis criterion* and leads to an approximation to the thermal equilibrium.

This principle can be applied to solve a combinatorial optimization problem, generating set of configurations and replacing energy and temperature with the objective function and the control parameter, respectively (P. van Laarhoven, 1987).

In a Simulated Annealing algorithm, the objective function generates, at each iteration, a random value for a solution *j* in the neighborhood of the current solution *i*. When compared, solution *j* is always accepted, if it improves the objective function value, or partially accepted (only a fraction of it) otherwise, expecting to avoid getting trapped at a local optima. The probability of accepting a solution is always non-zero, taking the value of 1, in the first case, and

$$exp(-\frac{\triangle F}{t_k})$$

in the second case, being $\triangle F$ the variation of the objective function value and t_k the temperature parameter at each iteration k, which is normally non-increasing. The capability of the Simulated Annealing algorithm to escape local optimum relies on the possibility of taking *hill-climbing* moves which do not improve the objective function value. With the decreasing of the temperature parameter, these moves take place less frequently and the systems approaches equilibrium (D. Henderson, 2003).

2.2.5 Ant Colony Algorithm

Like other optimization solutions, *Ant Colony System* was inspired by natural processes and developed through the study of its behaviour. Its principle is based on the real ants routine of finding the shortest way between a food source and their nest, without resorting to visual means. By releasing pheromones on their path, ants leave a trail that is accentuated every time one passes by, persuading others to choose the same passage. When confronted with two choices of path, ants divide themselves arbitrarily by the two possibilities. Admitting an average constant speed, shorter routes lead to a higher number of visitors per average and to a consequently faster accumulation of pheromone. The increased level of pheromone affects at the time of decision, leading to the selection of the most accented one. By choosing the one with the higher amount of pheromone, they contribute again for an increase, starting a "snowball effect"; soon they begin to use all the same shorter path.

In the Ant Colony System, the algorithm generates a set of artificial agents, called *ants*, which seek for good solutions in parallel, communicating through an exchange of information. The pheromone deposited by ants in the edges of the graph assigns the algorithm a memory, being this the main key of the system. Because it is stored in the edges and not within the agents themselves, it allows them to communicate through a process called *stigmergy*, which defends that the trace left from a first act stimulates the conduct of the next one. As for the heuristic information, edge's length is also given.

At the beginning, each ant creates a route, adding shorter edges with higher level of pheromone. When finished, that is, when each agent has it own tour, a *global pheromone updating rule* is applied, leading to an evaporation of a fraction of the pheromone in all edges and to a posterior deposit of pheromone in all the edges constituent of the current routes in proportion to how short that trail is. Consequently, the trails that were not refreshed get less desirable and shorter ones receive higher amounts of pheromone. Finally, all this steps are repeated in an iterated process (M. Dorigo, 1997).

2.2.6 Branch-and-Bound Algorithm

Branch-and-Bound Algorithm (B&B) is a search technique that relies on the expectation of finding the solution for a big problem by solving a easier related one. It analyzes the space of feasible solutions and divides it in smaller subsets, *bounding* the bigger difficult problem. A lower bound, in the case of a minimization problem, or an upper bound, for the maximization problems, of the objective function are defined for each subset and the ones with worse values that a known feasible solution get cut out from the set of future partitionings. The division stops whenever the process finds a feasible solution better than any bound from each subset; meanwhile, it calculates various feasible solutions.

The current smaller problems that substitute the main one are represented in the tree by nodes, denominated *leaves*, connected to the bigger problem by branches coming out from the main node, being this the reason to the term "branching" in the process' name. Each node has a bound associated and gets classified as *active*, if its bound is better than any current feasible solution. Otherwise, it is defined as *terminated* and excluded from the set of future analysis (E. L. Lawler, 1966).

2.2.7 Branch-and-Cut Algorithm

Branch-and-Cut Algorithm is an exact approach that consists of a *cutting plane method* into a *branch-and-bound algorithm*, improving the linear relaxations of the integer problem of the last one.

Solving a problem applying only cutting planes is not normally efficient, resulting in a consequent need for merging it with a branch-and-bound procedure. Furthermore, it leads to an improvement of the B&B, speeding it up.

In the case of not being able to determinate the exact solution, the branch-and-cut algorithm provides a lower bound, allowing to know how far is the solution from the optimal (Mitchell, 1988).

2.2.8 Branch-and-Price Algorithm

Branch-and-Price Algorithm is a branch-and-bound approach that resorts to a column generation method to solve its linear relaxation (G. Desaulniers, 2011).

2.2.9 Column Generation

Column Generation is an efficient method for solving large linear problems, considering only a subset of variables at a time. This reduced problem is called *Restricted Master Problem* (RMP) to which new variables are added whenever needed.

It starts with a few variables in the master problem and solves it, searching for new variables to be added. The search for the new ones consists on solving a subproblem, finding its optimal, that is, the one with the most *negative reduced cost* and therefore with potential improvement to the objective function and does not belong to the RMP. The procedure stops when solutions to all subproblems are non-negative, reaching the optimal LP solution to the master problem (J. Desrosiers, 2005).

2.2.10 Lagrangian Relaxation

Lagrangian Relaxation allows to find bounds of given problems which, if exact, end up being the optimal solution. Instead of constraints, it attributes to each one a "price" that can be paid or received, depending on its signal. This method permits the penalization in case of violation of constraints and the benefit otherwise. It adds the constraints to the objective function, converting it into a *dual problem*. This method transforms the initial primal problem into a simpler one which gives a good, or sometimes exact, approximation to the optimal solution (Lemaréchal, 2001).

2.2.11 Large Neighborhood Search

Large Neighborhood Search (LNS) applies methods to destroy and repair solutions and consequently improve them gradually. It starts with a initial solution that is partly destroyed and then rebuild with a repair method. The destroy method usually chooses different parts of the incumbent solution at every iteration to generate a stochastic process. The neighborhood of a solution is therefore defined as the space of solutions that can be generated through a sequentially destroy and repair method.

An important aspect that must be taken into account about the destroy method is the *degree of the destruction*. If low, it may ruin the concept of a large neighborhood as it may not allow to fully reach the entire search space. On the other hand, a large degree, that is, the destruction of a large part of the incumbent solution, can lead to repeated steps which can be time consuming or result in worse quality solutions. As for the repair method, the choice relies on a optimal or heuristic process. The first one generates the best possible full solution from the destroyed solution while for the second one a good solution is enough. The optimal option is not always better since it always leads to improved or equal solutions which can make the process to be stuck in a local optimum solution.

LNS usually jumps from a feasible solution to an infeasible one: it destroys part of the current feasible solution, turning it into infeasible, and converts it into a feasible option through a repair method (D. Pisinger, 2010).

Chapter 3

Electric Vehicle Scheduling Formulation

This chapter describes the solution developed to answer the problem presented in Chapter 1, represented by a mathematical formulation with multiple depots and charging stations.

3.1 Formal problem description

In order to allow a better visual understanding, the problem is described as a graph, displayed on the Figure 3.1.

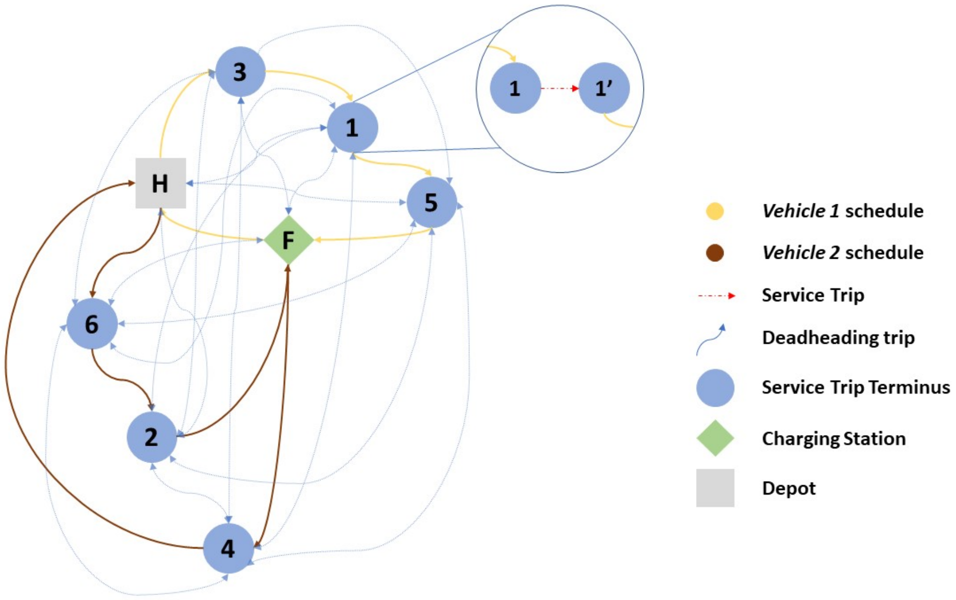

Figure 3.1: Representative graph of the problem

Since the information regarding what happens within the trajectory of the line is not important for the vehicle scheduling, each service trip is represented as a node. However, it is important to highlight that this *non-conventional node* does not symbolize any terminus, but the entire service trip itself, and its place on the graph relatively to other nodes is not related to any of its time or place information. Every depot and charging station is also displayed as different nodes. Since VSP only manage DH trips, the blue arcs represent each possible deadheading trip, being the yellow and brown arcs already allocated DH trips for the two hypothetical vehicles, *Vehicle 1* and *Vehicle 2*, respectively.

In the present formulation, the concept *event* refers to whatever is accomplished within a node. Therefore, its exist charging events, service trips events or even depot events.

In the solution exemplified by Figure 3.1, *Vehicle 1* goes from the depot to the starting location of *Service Trip 3*. After completing this service trip, the same vehicle goes from the end of it to the starting point of *Service Trip 1* to carry it out. The deadheading trip from the ending location of this trip to the starting point of *Service Trip 5* is then executed to give place to the fulfilment of this last one. Finally, *Vehicle 1* goes from the ending point of *Service Trip 5* to a charging station, for its charging event. After it charges to its maximum, it goes from this station to the depot. For the *Vehicle 2*, the process is the same, accomplishing *Service Trip 6* before *Service Trip 2* and *Service Trip 4* before it goes back to the depot. In between these two last service trips, this vehicle goes to a charging station to charge.

Even though in most cases the charging stations coincide with the depots, the possibility of having locations for charging only leads to distinguish them. A depot that is also a charging station will force, consequently, to a duplication of the nodes and the information.

For the developed formulation, an upper bound k for the number of vehicles is required, given that the vehicles that are not needed stay at the depot, not being taken into account in the costs. For each DH trip from a depot to a different node, a new vehicle is considered with a fixed cost that corresponds to its acquisition.

Both depots and charging stations have a maximum capacity. Depots' capacity is limited by the parking spots and the capacity of charging stations is limited by the number of chargers. In the first case, only the vehicles that leave the depot are considered, as those that stay at the parking spot will not, in principle, be used. Contrarily to this, the charging stations' capacity does not limit the number of vehicles that visit it in general, but only occasionally. In other words, the fulfillment of this capacity has to be constantly confirmed. Observing Figure 3.2, a scenario regarding the occupation of a charging station is exemplified. If the capacity of this charging station is 2, it has to be checked that, at every moment t, no more than two vehicles are charging. However, this does not mean that only two vehicles may stop at that station, but only two at a time.

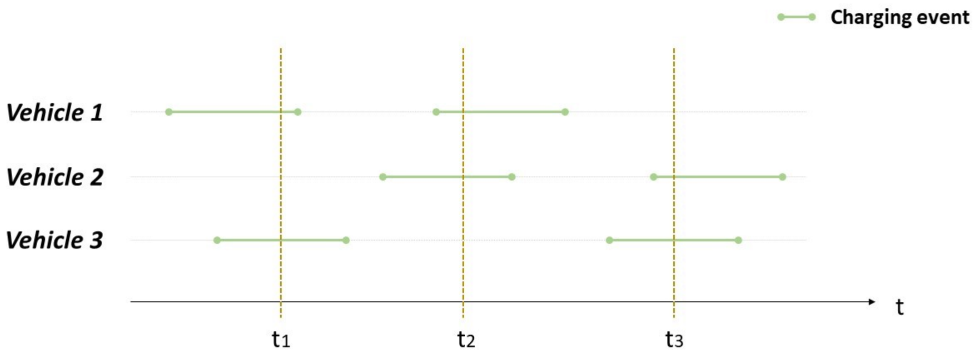

Figure 3.2: Representation of charging events at a single charging station

The duration l of a charging event covers the entire time required from the arrival to the station to its departure, including parking time and any logistic tasks.

Since the problem allows multiple visits to the same charging station by the same vehicle, it becomes necessary to establish a parameter p that defines the order of the vehicle deadheading trips, allowing the distinction of different visits to the same place by the same vehicle. Using the previous example, Figure 3.3 exemplifies how this parameter works.

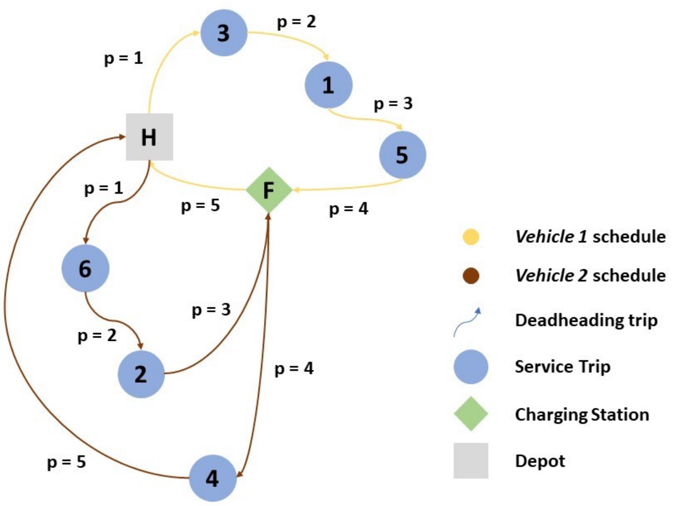

Figure 3.3: Exemplification of the parameter p in a graph

Contrary to charging or depot events, service trips have a starting time α that must be obeyed. However, it is not always feasible to arrive at the starting point exactly at the starting time, forcing the vehicle to wait an idle time u. Since between service trips, this idle period demands the presence of a driver, it has to be considered as an operational cost and therefore minimized.

3.2 Assumptions

In an initial phase, some assumptions are required in order to facilitate the development of the problem. Posteriorly, some of them might be eliminated, increasing the complexity of the formulation, or new ones added, reducing the computational time.

- All vehicles are the same type;
- Whenever a vehicle stops at a charging point, it only leaves when charged to the maximum allowed;
- Whenever a vehicle stops at a charging station, it charges a fixed amount of time (energy), regardless its battery level;
- A vehicle schedule starts and ends at the same depot, not being allowed the vehicle to visit any depot in between during its schedule;
- The battery losses whenever the vehicle is stopped are assumed to be zero;
- All schedules start and end in the same day.

In addition to these optional assumptions, some commonsense prohibitions are also settled, such as not allowing a vehicle to visit two charging stations in a row or to stay at the same place (except for depots).

3.3 Variables

This section explains the sets, parameters and variables needed to formulate the problem.

3.3.1 Sets and Indexes

In order to represent the problem as a graph, the following sets are necessary.

S = {1..s}	set of $s = h + f + n$ nodes
H = {1..h}	set of h depots
F = {h+1..h+f}	set of f charging stations
N = {h+f+1..h+f+n}	set of n service trips
K = {1..k}	set of k vehicles
P = {1..p}	set of p possible stops
P_0 = {1..p-1}	set of $p-1$ possible stops
W = {1,2}	set of W values for each charging event interval

3.3.2 Parameters

For the formulation to adjust each case, some input is necessary entering as a form of parameters.

c_k purchasing cost of each vehicle

c_t logistic costs per unit of time

c_e energy costs per unit of energy

E_0 maximum energy capacity of vehicle's battery

e_{ij} energy consumption of DH trip from the end of trip i to the beginning of trip j

e_i energy consumption of event i

α_i starting time of service trip i

d_i duration of event i

d_{ij} duration of DH trip from the end of event i to the beginning of event j

r_h capacity of depot h

q_f capacity of charging station f

M large constant

3.3.3 Auxiliary variables

For the level of energy to be monitored, an auxiliary continuous and positive group of variables is added to the formulation to estimate the amount left after every assignment. In order to record the starting time of each event, a second auxiliary group of variables is created.

E_i^{kp} energy level of vehicle k after completing event i before DH trip p

t_i^{kp} starting time of the event i by vehicle k before DH trip p

3.3.4 Decision variables

Two groups of binary decision variables are on the basis of the formulation. The first one, denoted, for simplification, by x, determines which DH trips lead to the best solution. The second one, Y, is related to every two charging intervals of different vehicles. If $Y = 1$, then the intervals are somehow intersected, which means that at least two chargers will be needed and those two vehicles cannot share the same charger.

$$x_{ij}^{kp} = \begin{cases} 1, & \text{if vehicle } k \text{ goes from node } i \text{ to node } j \text{ in its } p^{th} \text{ DH trip} \\ 0, & \text{otherwise} \end{cases}$$

$$Y_i^{kpk'p'fw} = \begin{cases} 1, & \text{if } t_i^{kp} \text{ (if } w = 1\text{) or } (t_i^{kp} + l) \text{ (if } w = 2\text{) belong to the time interval} \\ & [t_i^{k'p'}, t_i^{k'p'} + l] \\ 0, & \text{otherwise} \end{cases}$$

3.4 Objective Function

The objective function aims to minimize the total costs of the vehicle scheduling. Since we cannot minimize the travel time or the energy wasted with each service trip, only deadheading trips are taken into consideration in the calculation of its value.

$$min \sum_{k\in K} \sum_{i\in H, j\in N\cup F} x_{hj}^{k1} c_k + \sum_{k\in K, p\in P} \sum_{i,j\in S} d_{ij} x_{ij}^{kp} c_t + \sum_{k\in K, p\in P} \sum_{i,j\in N} (\alpha_j - \alpha_i - d_i - d_{ij}) x_{ij}^{kp} c_t + \\ + \sum_{k\in K, p\in P} \sum_{i,j\in S} e_{ij} x_{ij}^{kp} c_e \quad (3.1)$$

The total cost can be defined as the sum of the purchasing costs (investment on the acquisition of vehicles), operational costs (whatever cost it takes to operate the vehicle per unit of time, such as driver's payment) and energy costs (electricity costs).

Therefore, the first sum regards the cost of the vehicles purchased. It multiples the number of vehicles that leave a depot by a fixed cost assigned to each vehicle.

The second component calculates the operational costs, that is, the cost of the work time of a driver, regarding the deadheading trips. It multiples the total time wasted on DH trips by the logistic cost per unit of time. The third sum considers the additional operational costs caused by the idle time u between service trips, since only then is required the presence of a driver, multiplying the period by the cost per unit of time.

Finally, the last sum represents the total cost of the energy used on deadheading trips, being that c_e is the energy cost by unit of energy.

3.5 Constraints

The constraints to which the problem is restricted can be divided in the following main categories.

Vehicle Scheduling Constraints:

$$\sum_{k\in K, p\in P} \sum_{j\in S} x_{ij}^{kp} = 1, \ \forall i \in N \quad (3.2)$$

$$\sum_{k\in K, p\in P} \sum_{i\in S} x_{ij}^{kp} = 1, \ \forall j \in N \quad (3.3)$$

$$\sum_{h\in H, j\in S, p\in P} x_{hj}^{kp} = 1, \ \forall k \in K \quad (3.4)$$

$$\sum_{h\in H, j\in S} x_{hj}^{k1} = 1, \ \forall k \in K \quad (3.5)$$

$$\sum_{i,j \in S} x_{ij}^{kp} \leq 1, \; \forall p \in P, k \in K \tag{3.6}$$

$$\sum_{i \in S} x_{ij}^{k(p-1)} - \sum_{i \in S} x_{ij}^{kp} = 0, \; \forall k \in K, j \in N \cup F, p \in P : p > 1 \tag{3.7}$$

$$\sum_{i \in S} x_{ih}^{kp} - \sum_{i \in S} x_{hi}^{k1} = 0, \; \forall k \in K, h \in H \tag{3.8}$$

Equations 3.2 and 3.3 guarantee that a vehicle only ends and starts a given service trip once, respectively. Equation 3.4 allocates a depot for each vehicle, while equation 3.5 forces that depot trip to be the first one on the schedule. Equation 3.6 assures that there is not more than one deadheading p for each schedule of each vehicle. Equation 3.7 describes the flow conservation for service trips and charging stations nodes and equation 3.8 is its specification for depots.

Depot Constraints:

$$\sum_{k \in K} \sum_{j \in N \cup F} x_{hj}^{k1} \leq r_h, \; \forall h \in H \tag{3.9}$$

Equation 3.9 assures that each depot's capacity is not exceeded.

Time Constraints:

The next constraints regard the enforcement of the starting times of each service trip.

$$(t_i^{kp} + d_i + d_{ij}) x_{ij}^{kp} \leq t_j^{k(p+1)}, \; \forall i, j \in S, k \in K, p \in P_0 \tag{3.10}$$

$$t_i^{kp} = \alpha_i, \; \forall i \in N, k \in K, p \in P \tag{3.11}$$

$$t_f^{kp} x_{fi}^{kp} = (t_i^{k(p+1)} - d_{fi} - d_f) x_{fi}^{kp}, \; \forall i \in S, f \in F, k \in K, p \in P_0 \tag{3.12}$$

$$t_h^{kp} x_{ih}^{k(p-1)} = (t_i^{k(p-1)} + d_i + d_{ih}) x_{ih}^{k(p-1)}, \; \forall i \in S, h \in H, k \in K, p \in P : p > 1 \tag{3.13}$$

$$t_h^{k1} x_{hj}^{k1} = (t_j^{k2} - d_{hj}) x_{hj}^{k1}, \; \forall j \in S, h \in H, k \in K \tag{3.14}$$

Equation 3.10 is the time constraint that guarantees that each vehicle arrives in time to an event. The next time constraints record each starting time of an event into a variable t. Therefore, Equation 3.11 assigns the mandatory starting time of each service trip to the variable t. Equations 3.12, 3.13 and 3.14 calculate the starting time of a charging event, the entrance of a vehicle into a depot and its exit, respectively.

Energy Constraints:

$$0,2E_0 \leq E_i^{kp} \leq 0,8E_0,\ \forall i \in S, k \in K, \forall p \in P \tag{3.15}$$

$$E_f^{kp} = 0,8E_0,\ \forall k \in K, f \in F, \forall p \in P \tag{3.16}$$

$$E_j^{k(p+1)} \leq (E_i^{kp} - e_{ij} - e_j)x_{ij}^{kp} + 0,8E_0\,(1 - x_{ij}^{kp}),\ \forall j \in N, i \in S, k \in K, p \in P_0 \tag{3.17}$$

$$E_h^{k1} \leq (E_i^{kp} - e_{ij} - e_j)x_{ij}^{kp} + 0,8E_0\,(1 - x_{ij}^{kp}),\ \forall h \in H, i \in S, k \in K, p \in P \tag{3.18}$$

Equation 3.15 ensures that the battery level always stands within its limits, while equation 3.16 forces it to be at its maximum after charging. Both equations 3.17 and 3.18 update the battery level after leaving a service trip or a depot, respectively.

Charging Stations Constraints:

$$(\sum_{k' \in K: k' \neq k, p' \in P} Y_{kpk'p'}^{fw}) + 1 \leq q_f,\ \forall f \in F, k \in K, p \in P, w \in W \tag{3.19}$$

$$t_f^{kp} \in\ ([t_f^{k'p'}, t_f^{k'p'} + d_f] + (1 - Y_{kpk'p'}^{f1})[-M, +M]),\ \forall f \in F, \forall k, k' \in K, \forall p, p' \in P \tag{3.20}$$

$$(t_f^{kp} + d_f) \in\ ([t_f^{k'p'}, t_f^{k'p'} + d_f] + (1 - Y_{kpk'p'}^{f2})[-M, +M]),\ \forall f \in F, \forall k, k' \in K, \forall p, p' \in P \tag{3.21}$$

$$t_i^{kp} \in\ (]-\infty, t_i^{k'p'}[\cup]t_i^{k'p'} + d_f, \infty[+ Y_{kpk'p'}^{f1}[-M, +M]),\ \forall i \in F, \forall k, k' \in K, \forall p, p' \in P \tag{3.22}$$

$$(t_i^{kp} + d_f) \in\ (]-\infty, t_i^{k'p'}[\cup]t_i^{k'p'} + d_f, \infty[+ Y_{kpk'p'}^{f2}[-M, +M]),\ \forall i \in F, \forall k, k' \in K, \forall p, p' \in P \tag{3.23}$$

Equation 3.19 controls charging stations' capacity by ensuring that the number of charging events at the same time is lower or equal to the capacity. On the other hand, equations from 3.20 to 3.23 force the variables Y to be one or zero according to the position in time of the start time (equations 3.20 and 3.22) or finish time (equations 3.21 and 3.23) of a charging event relative to another. For a better clarification, observing the Figure 3.2 and selecting a charging event, if the starting time t of that selected charging event intersects another, then the variable Y that relates those two intervals (with $w = 1$ because it is the starting point) is 1. The same thing happens for the ending time $t + l$, but with $w = 2$. Then, Equation 3.19 counts all the charging intervals that intersects a specific charging interval and sums 1 to have into account the charger for the selected charging event.

General Constraints:

$$x_{ij}^{kp} \in \{0, 1\} \tag{3.24}$$

$$Y_i^{kpk'p'} \in \{0, 1\} \tag{3.25}$$

$$x_{h_1 h_2}^{kp} = 0, \ \forall h_1, h_2 \in H : h_1 \neq h_2, k \in K, p \in P \tag{3.26}$$

$$x_{f_1 f_2}^{kp} = 0, \ \forall f_1, f_2 \in F, k \in K, p \in P \tag{3.27}$$

Equations 3.24 and 3.25 characterize the binary nature of both decision variables. Equation 3.26 does not allow a vehicle to go from a depot to another, while equation 3.27 precludes trips between charging stations.

Chapter 4

Solution Approach

4.1 Software Application

The formulation is programmed in *Optimization Programming Language* (OPL), using *IBM ILOG CPLEX Optimization Studio* software, 12.9.0.0 version.

4.1.1 *IBM ILOG CPLEX Optimization Studio* software

Normally referred as simply *CPLEX*, this software uses mathematical and constraint programming to rapidly solve optimization models. It supplies several languages interfaces, such as Python and C, and supports *Optimization Programming Language* (OPL) as well as solvers like *CP Optimizer* or *CPLEX*.

CPLEX applies Branch-and-Bound to solve models, aiming for an optimal solution. Furthermore, when set with default parameters, it resorts to heuristics whenever it seems to be favorable, leading to integer solutions at nodes during branching. This speeds up the process when compared to branching alone, not deteriorating substantially the quality of the solution.

4.1.2 Optimization Programming Language

As stated previously, *IBM ILOG CPLEX Optimization Studio* allows the development of a combinatorial optimization model using OPL. This modeling language leads to a more compact and synthesised script, decreasing the effort needed to write it. It can be useful in many areas like planning, scheduling, sequencing, among others.

4.2 Programming

When developed into OPL, the formulation requires some adaptations to improve the efficiency of the script. Being the computational time a major limitation of optimization problems, this section focuses on the simplification of the model.

In order to avoid the creation of a variable t to record the beginning of each event, which consumes computational time, that parameter is eliminated. Not only it would be an additional variable, but its inclusion would lead to non-linear constraints that would have to be linearized, increasing the computational time. Therefore, every time that information is needed, it has to be calculated, leading to the development of multiple constraints to characterize the different possible cases described in the Figure 4.1.

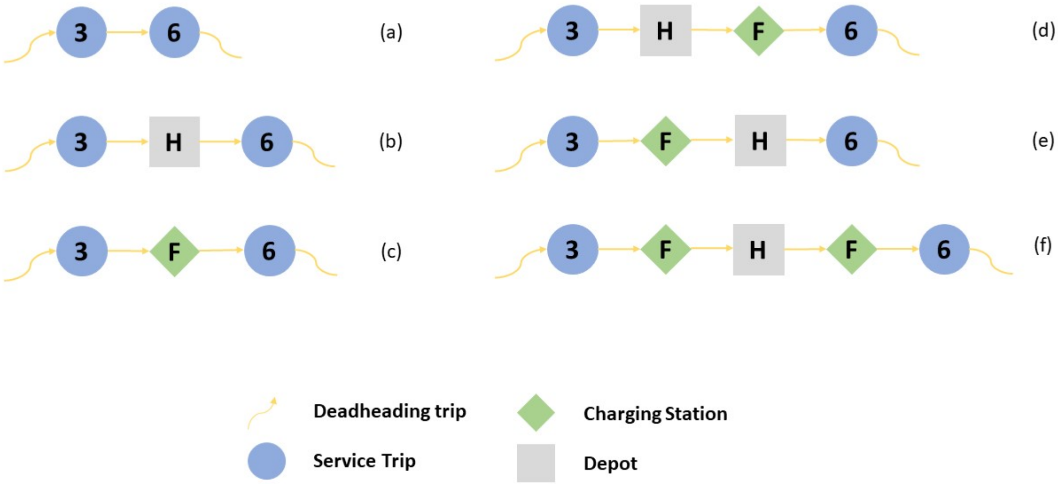

Figure 4.1: Possible scenarios between two service trips

A new parameter b is created to help distinguish starting times of events from different days, therefore having the value of the number of units of time a day has (24 if the input times are by hour or 1440 if they are by minute).

$$(\alpha_i + d_i + d_{ij})x_{ij}^{kp} \leq \alpha_j, \ \forall i, j \in N : i \neq j, k \in K, p \in P \tag{4.1}$$

$$(\alpha_i + d_i + d_{ig} + d_g + d_{gj})x_{ig}^{kp}x_{gj}^{k(p+1)} \leq \alpha_j, \ \forall i, j \in N : i \neq j, g \in F \cup H, k \in K, p \in P \tag{4.2}$$

$$(\alpha_i + d_i + d_{if} + d_f + d_{fh} + d_h + d_{hj})x_{if}^{kp}x_{fh}^{k(p+1)}x_{hj}^{k1} \leq \alpha_j + b, \quad \forall i, j \in N : i \neq j, h \in H, f \in F, k \in K, p \in P \tag{4.3}$$

$$(\alpha_i + d_i + d_{ih} + d_h + d_{hf} + d_f + d_{fj})x_{ih}^{kp}x_{hf}^{k1}x_{fj}^{k2} \leq \alpha_j + b,$$
$$\forall i,j \in N : i \neq j, h \in H, f \in F, k \in K, p \in P \quad (4.4)$$

$$(\alpha_i + d_i + d_{if_1} + d_{f_1} + d_{f_1h} + d_h + d_{hf_2} + d_{f_2} + d_{f_2j})x_{if_1}^{kp}x_{f_1h}^{k(p+1)}x_{hf_2}^{k1}x_{f_2j}^{k2} \leq \alpha_j + b,$$
$$\forall i,j \in N : i \neq j, h \in H, f_1, f_2 \in F, k \in K, p \in P \quad (4.5)$$

Equation 4.1 describes the case of two service trips in a row (Figure 4.1.(a)), while Equation 4.2 converts it to the case where there is an additional stop, either a depot (Figure 4.1.(b)) or a charging station (Figure 4.1.(c)), in between the two service trips. Equations 4.3 and 4.4 expand the previous constraints to the possibility of having two stops in between the service trips: a depot and then a charging station on the former (Figure 4.1.(d)), and the other way around on the later (Figure 4.1.(e)). Finally, Equation 4.5 exposes the only possibility of having three stops between the two service trips: a charging station before going to the depot, and another after it leaves the depot (Figure 4.1.(f)). This 5 new constraints replace Equation 3.10, while Equations 3.11 to 3.14 are eliminated.

When multiplying two variables, the equation becomes non-linear which leads to the possibility of non-convexity. Being this a limitation of CPLEX, it forces the equation to be altered.

There are three ways to deal with non-convex constraints: (1) linearization, (2) the use of constraint programming or (3) to resort to logical constraint within CPLEX. The first option leads to the appearance of several constraints for each equation to be linearized, which makes development and understanding more confusing. The second one forces the variables to be integer, which would lead to some limitations. Therefore, non linearity was solved using logical constraints.

Constraint 4.1 is converted into equation 4.6.

$$(x_{ij}^{kp} = 1) \Rightarrow (\alpha_i + d_i + d_{ij} \leq \alpha_j), \; \forall i,j \in N : i \neq j, k \in K, p \in P \quad (4.6)$$

Because the equation 4.2 describes the possibility of having either a charging station or a depot in between two service trips, when written in OPL this constraint gives rise to two new equations, 4.7 and 4.8, respectively.

$$(x_{if}^{kp} = 1 \wedge x_{fj}^{k(p+1)} = 1) \Rightarrow (\alpha_i + d_i + d_{if} + d_f + d_{fj} \leq \alpha_j), \; \forall i,j \in N : i \neq j, f \in F, k \in K, p \in P_0 \quad (4.7)$$

$$(x_{ih}^{kp} = 1 \wedge x_{hj}^{k1} = 1) \Rightarrow (\alpha_i + d_i + d_{ih} + d_{hj} \leq \alpha_j + b),\ \forall i,j \in N : i \neq j, h \in H, k \in K, p \in P \quad (4.8)$$

Equations 4.9, 4.10 and 4.11 are linear versions of equation 4.3, 4.4 and 4.5, respectively.

$$(x_{ih}^{kp} = 1 \wedge x_{hf}^{k1} = 1 \wedge x_{fj}^{k2} = 1) \Rightarrow (\alpha_i + d_i + d_{ih} + d_h + d_{hf} + d_f + d_{fj} \leq \alpha_j + b, \\ \forall i,j \in N : i \neq j, h \in H, f \in F, k \in K, p \in P \quad (4.9)$$

$$(x_{if}^{k(p-1)} = 1 \wedge x_{fh}^{kp} = 1 \wedge x_{hj}^{k1} = 1) \Rightarrow (\alpha_i + d_i + d_{if} + d_f + d_{fh} + d_h + d_{hj} \leq \alpha_j + b), \\ \forall i,j \in N : i \neq j, h \in H, f \in F, k \in K, p \in P : p > 1 \quad (4.10)$$

$$(x_{if_1}^{k(p-1)} = 1 \wedge x_{f_1h}^{kp} = 1 \wedge x_{hf_2}^{k1} = 1 \wedge x_{f_2j}^{k2} = 1) \Rightarrow \\ (\alpha_i + d_i + d_{if_1} + d_{f_1} + d_{f_1h} + d_h + d_{hf_2} + d_{f_2} + d_{f_2j} \leq \alpha_j + b), \\ \forall i,j \in N : i \neq j, h \in H, f_1, f_2 \in F, k \in K, p \in P : p > 1 \quad (4.11)$$

Regarding the energy constraints, the same approach is applied, resulting in the equation 4.12 for the constraint 3.17, and equation 4.13 for the constraint 3.18.

$$(x_{ij}^{kp} = 1) \Rightarrow (E_j^{k(p+1)} = E_i^{kp} - e_{ij} - e_j),\ \forall i \in S, j \in N : j \neq i, k \in K, p \in P_0 \quad (4.12)$$

$$(x_{ih}^{kp} = 1) \Rightarrow (E_h^{k1} = E_i^{kp} - e_{ih}),\ \forall i \in S, h \in H : h \neq i, k \in K, p \in P \quad (4.13)$$

Regarding the chargers, vehicles arriving to charging stations can either come from a depot or from the terminus of a service trips, since trips between two charging events are forbidden. However, once starting times of depots events are not recorded, this would force the appearance of several new constraints describing each possibility (from a service trip to a depot and then to a charging station or from a service trip to a charging station and only then to a depot and another charging station) in a combinatorial way for both vehicles arriving. This would lead to a very large computational time with a small and basic example of data. Therefore, in order to improve its performance and because in practice it does not happen, DH trips from depots directly to charging stations are banned, reducing considerably the computational time. Because CPLEX software does not allows strictly *less than*, a very small constant m was used in order to distinguish whenever the two sides of the inequation had the same value. Equation 3.20 is therefore replaced by equations 4.14, 4.15, 4.16 and 4.17, while equation 3.22 is replaced by equations 4.18 and 4.19.

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f1} = 1) \Rightarrow (\alpha_j + d_j + d_{jf} \leq \alpha_i + d_i + d_{if} - m),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.14)$$

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f1} = 1) \Rightarrow (\alpha_j + d_j + d_{jf} + d_f \geq \alpha_i + d_i + d_{if} + m),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.15)$$

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f2} = 1) \Rightarrow (\alpha_j + d_j + d_{jf} \leq \alpha_i + d_i + d_{if} + d_f - m),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.16)$$

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f2} = 1) \Rightarrow (\alpha_j + d_j + d_{jf} + d_f \geq \alpha_i + d_i + d_{if} + d_f + m),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.17)$$

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f1} = 0) \Rightarrow$$
$$(\alpha_i + d_i + d_{if} \geq \alpha_j + d_j + d_{jf} + d_f \ \vee \ \alpha_i + d_i + d_{if} \leq \alpha_j + d_j + d_{jf}),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.18)$$

$$(x_{if}^{kp} = 1 \wedge x_{jf}^{k'p'} = 1 \wedge Y_{kpk'p'}^{f2} = 0) \Rightarrow$$
$$(\alpha_i + d_i + d_{if} + d_f \leq \alpha_j + d_j + d_{jf} \ \vee \ \alpha_i + d_i + d_{if} + d_f \geq \alpha_j + d_j + d_{jf} + d_f),$$
$$\forall f \in F, k, k' \in K : k \neq k', p, p' \in P, i, j \in N : i \neq j \quad (4.19)$$

The elimination of the possibility of a trip from a depot to a charging station led to new alterations, such as the simplification of the depot constraint 3.9 which is then replaced by equation 4.20.

$$\sum_{k \in K} \sum_{j \in N} x_{hj}^{k1} \leq r_h, \ \forall h \in H \quad (4.20)$$

The objective function also gets influenced in the first sum, resulting in a new equation 4.21.

$$min \sum_{k} \sum_{i \in H, j \in N} x_{hj}^{k1} c_k + \sum_{k,p} \sum_{i,j} d_{ij} x_{ij}^{kp} c_t + \sum_{k,p} \sum_{i,j \in N} (\alpha_j - \alpha_i - d_i - d_{ij}) x_{ij}^{kp} c_t + \sum_{k,p} \sum_{i,j} e_{ij} x_{ij}^{kp} c_e \quad (4.21)$$

As for the time constraints, the reduction of feasible solutions leads to a consequent reduction of constraints, where constraints 4.4 and 4.5 get eliminated and, consequently, equations 4.10 and 4.11.

Finally, an obvious constraint has to be added as described in equation 4.22.

$$x_{hf}^{kp} = 0, \; \forall h \in H, f \in F, k \in K, p \in P \quad (4.22)$$

4.3 Data preparation

In order to run the algorithm, some data is necessary.

All the costs c_k, c_t and c_e are constant parameters given as inputs, as well as the maximum capacity of the vehicles' battery E_0.

As stated initially, the energy consumption of the DH trips, as well as its duration, enters the script as a squared matrix of $s \times s$, where the first h rows and columns are reserved for depots information, the next f for charging station and finally the service trips.

The energy consumption and duration of each event are two vectors of size s where, once again, the first h values refer to depot information, the next f for charging station and the last ones for service trips. For charging nodes or depots, the energy consumption took the value of zero in every experience done, as it represents despicable values in the real world, but it can take any other value. For the service trips, it is equated to the amount of energy needed for each trajectory. As for the duration, in the case of the charging stations, it takes a value equal to l and the last n values take the value of each trajectory's duration. As for the depots, it also took the value of zero in all the experiments, but it can take any other value if needed for mandatory tasks.

The starting times α are inputted as a vector of size n, since they refer to service trips only. Finally, the capacity r_h of the each depot and the number of chargers q_f of each charging station are also introduced as vectors of size h and f, respectively. As for the constant M, it takes an arbitrary and very high value.

Chapter 5

Results

In this chapter, real data is tested and its results analyzed. A brief presentation of the company that provided the information is given.

5.1 STCP

Sociedade de Transportes Colectivos do Porto, SA, often designated as STCP, is a Portuguese company born in 1946 aiming to supply an urban public transport service to the Porto metropolitan area. Its heterogeneous bus fleet is currently composed by standard, articulated or minis buses.

In recent years, STCP has focused on providing a cleaner service, acquiring electric and natural gas vehicles. Their 15 new electric buses currently circulate in three different lines of Porto city.

5.2 Provided Data

In this section, an interpretation and analysis of the data provided is described.

5.2.1 Description of the Provided Data

The data provided by STCP relates to several lines, describing the service trips that constitute it and which are periodically repeated in both directions. It regards starting and ending times (in minutes) of each service trip, as well as its starting and ending locations. It is also provided the distance (in meters) and duration (in minutes) of each deadheading trip between one depot and the terminus of each service trip. Finally, the current solution is also presented.

The data is summarize in Table 5.1.

Table 5.1: Data size provided by STCP

Node	Quantity
Depots	1
Service Trips	100

The earliest starting time of a service trip in the provided data is 346 minutes and the latest ending time is 1300 minutes, which is equivalent to 5h46 am and 9h40 pm, therefore confirming the previous assumption that a schedule must end in the same day as its starting time.

There are 7 nodes (locations) provided by STCP, of which only one is a depot (Node ID: 11), as observed in the Table 5.2.

Table 5.2: Information about the nodes provided by STCP

Node Name	ID	Depot
Aliados	38	No
ER. Francos	11	Yes
Fonte da Moura	3	No
Mota Pinto	800307	No
P. Alegre	25	No
Pr. Republica	72	No
Boavista	4	No

Despite the high number of service trips, they all regard the same lines repeated in both directions. Therefore, they can be grouped in 6 different groups, as presented in the Table 5.3. Even though a specific trajectory repeated in both directions is considered to be just one line, in the present approach it is discriminated into different trajectories to allow a easier comprehension. Summarizing the service trips in 6 trajectories helps to interpret the duration and energy consumption of each possible deadheading trip.

Table 5.3: Service trips grouped in 6 different trajectories

Trajectory ID	Starting Node	Ending Node	Quantity
1	25	38	42
2	38	25	44
3	800307	25	4
4	800307	38	3
5	25	3	6
6	38	800307	1

A detailed description of each service trip can be observed in the Table A.1 in Appendix A.

5.2.2 Current Solution

As stated previously, the current vehicle scheduling for the lines in study is provided. Once it is carried out by non-electric vehicles, not a single charging event occurs during the entire shifts.

This solution requires 6 vehicles for the 100 service trips, whose schedules are presented in the Table 5.4. For a better visual comprehension, the depot is marked as a highlighted **H** and each service trip is represented by its corresponding number on the Table A.1.

Table 5.4: Current solution of STCP by service trips

Vehicle	Sequence
1	**H** - 87 - 1 - 44 - 4 - 49 - 94 - **H**
2	**H** - 91 - 43 - 3 - 48 - 9 - 54 - 15 - 60 - **H** - 90 - 22 - 67 - 27 - 72 - 32 - 77 - 37 - 82 - 41 - 86 - 99 - **H**
3	**H** - 88 - 2 - 46 - 7 - 52 - 95 - **H** - 13 - 58 - 19 - 64 - 24 - 69 - 29 - 74 - 34 - 79 - 39 - 84 - 42 - 100 - **H**
4	**H** - 92 - 45 - 6 - 51 - 11 - 56 - 17 - 62 - 21 - 66 - 26 - 71 - 31 - 76 - 36 - 81 - 40 - 85 - 98 - **H**
5	**H** - 89 - 5 - 50 - 10 - 55 - 14 - 59 - 18 - 63 - 23 - 68 - 28 - 73 - 33 - 78 - 38 - 83 - 97 - **H**
6	**H** - 93 - 47 - 8 - 53 - 12 - 57 - 16 - 61 - 20 - 65 - 25 - 70 - 30 - 75 - 35 - 80 - 96 - **H**

Two of these schedules include a trip to the depot in the middle of the shift, which is not allowed in the developed formulation.

As in the mathematical solution presented in this paper, all schedules start and end at a depot allocated for that vehicle. Also, all service trips are fulfilled, obeying to its starting time and location.

Organizing the data by time in a graph, it is possible to observe the operating time of each vehicle, allowing to visualize its percentage of use. Examining the Figures 5.1 to 5.6, it is clear that *Vehicle 1* is the one with the lower operational time, spending most of its time at the depot.

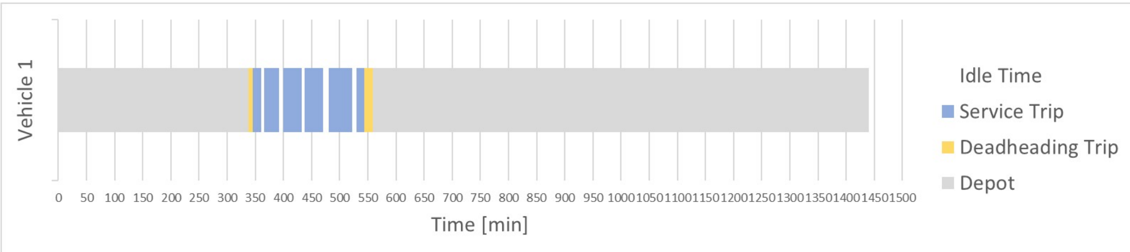

Figure 5.1: Schedule of *Vehicle 1* by STCP

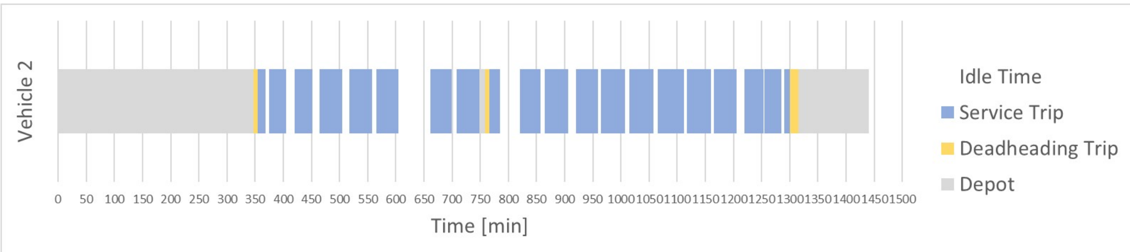

Figure 5.2: Schedule of *Vehicle 2* by STCP

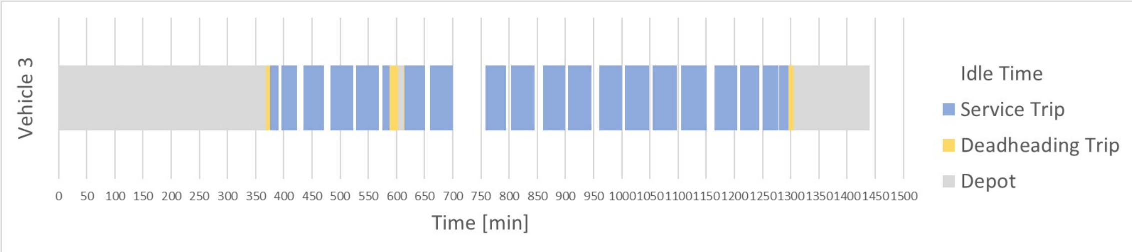

Figure 5.3: Schedule of *Vehicle 3* by STCP

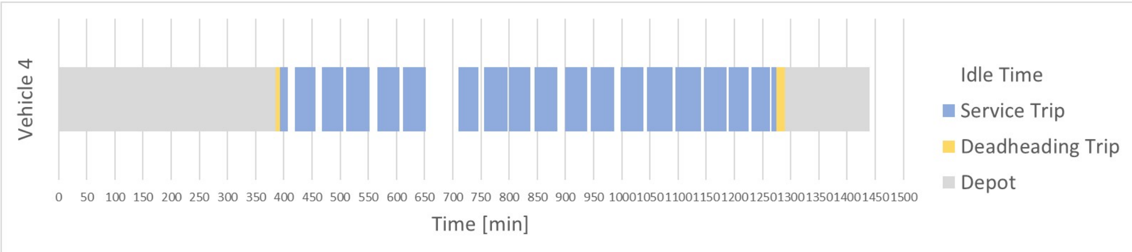

Figure 5.4: Schedule of *Vehicle 4* by STCP

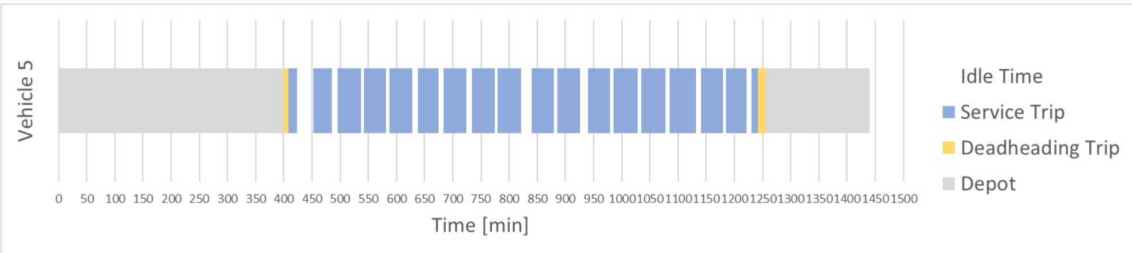

Figure 5.5: Schedule of *Vehicle 5* by STCP

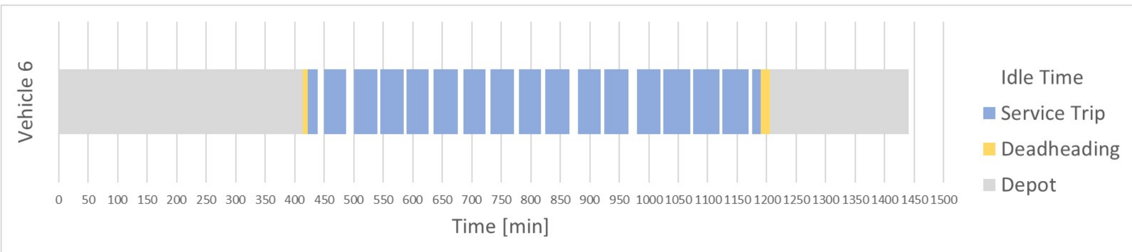

Figure 5.6: Schedule of *Vehicle 6* by STCP

Each blue segment regards different service trips, while the yellow sections refer to deadheading trips. At idle times, represented by blank spaces, and depots, represented by grey portions, the vehicles are parked.

In a different perspective, the same solution can be observed in the Table 5.5, but this time represented by the equivalent trajectory ID, allowing a better understanding of the course of each vehicle.

Table 5.5: Current solution of STCP by trajectories

Vehicle	Sequence
1	**H** - 3 - 1 - 2 - 1 - 2 - 5 - **H**
2	**H** - 4 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - **H** - 3 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 5 - **H**
3	**H** - 3 - 1 - 2 - 1 - 2 - 5 - **H** - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 6 - **H**
4	**H** - 4 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 5 - **H**
5	**H** - 3 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 5 - **H**
6	**H** - 4 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 1 - 2 - 5 - **H**

5.3 Experiments

In this section, computational experiments are described and its results analyzed in detail.

5.3.1 Data Preparation

The provided data is not adapted to the algorithm developed. Therefore, some preparation is required in order to run it.

Because the energy consumption for each service trip is not provided, an estimation is calculated. First, an average duration of each trajectory is determined based on the duration of each service trip, as observed in the Table 5.6.

Table 5.6: Average duration of each trajectory (in minutes)

Trajectory ID	Average duration [min]
1	39
2	41
3	15
4	14
5	12
6	16

According to STCP (2018), the average velocity of a commercial bus is around 16 km/h. Consequently, the calculation of the average distance of each trajectory becomes an easy problem. The results are stated in Table 5.7.

Table 5.7: Average distance of each trajectory (in km)

Trajectory ID	Average distance [km]
1	10.4
2	10.9
3	4.0
4	3.7
5	3.2
6	4.3

The average energy consumption is 0.9 kWh/km, according to the *Caetanobus - Fabricação de Carroçarias SA*, which allows the determination of that value for each trajectory, obtaining the results presented in the Table 5.8.

Table 5.8: Average energy consumption of each trajectory (in kWh)

Trajectory ID	Average energy consumption [kWh]
1	9.4
2	9.8
3	3.6
4	3.4
5	2.9
6	3.8

Those values are therefore applied as the energy consumption for each service trip according to its trajectory.

The duration of the DH trips between the depot and the beginning of each trajectory is given depending on the time of day, which is unsuitable to the algorithm proposed. Therefore, an average is calculated. Since the gap between the arrival of a vehicle to a depot and its exit is considerably large, the adjustment to the period of the day can be made posteriorly.

As for the energy consumption, only the distance between the depot and each trajectory is presented. Knowing the average consumption per kilometer, it is possible to calculate the energy consumption following the previously applied steps.

Furthermore, duration and energy consumption of the deadheading trips between terminus is not provided which leads to the establishment of values based on common sense. Between service trips where the first ends at the starting place of the second, this value is obviously zero. Also, for example, from a service trip of the *Trajectory 1* to another of the same type, since going back from *Node 38* to *Node 25* is the same as doing a service trip of the *Trajectory 2* type, this parameter takes a really high value in order to avoid this situation. The result can be observed in the followings tables where, as stated previously, each trajectory is represented by a node, being Table 5.9 relative to the duration and Table 5.10 to the energy consumption.

Table 5.9: Average duration of each DH trip between trajectories (in minutes)

Node	H	1	2	3	4	5	6
H	0	1	16	8	8	1	16
1	15	10^{99}	0	10^{99}	10^{99}	10^{99}	0
2	1	0	10^{99}	10^{99}	10^{99}	0	10^{99}
3	1	0	10^{99}	10^{99}	10^{99}	0	10^{99}
4	15	10^{99}	0	10^{99}	10^{99}	10^{99}	0
5	15	10^{99}	10^{99}	10^{99}	10^{99}	10^{99}	10^{99}
6	9	10^{99}	10^{99}	0	0	10^{99}	10^{99}

Table 5.10: Average energy consumption of each DH trip between trajectories (in kWh)

Node	H	1	2	3	4	5	6
H	0	4.5	5.085	2.7	2.7	4.5	5.085
1	5.085	10^{99}	0	10^{99}	10^{99}	10^{99}	0
2	4.5	0	10^{99}	10^{99}	10^{99}	0	10^{99}
3	4.5	0	10^{99}	10^{99}	10^{99}	0	10^{99}
4	5.085	10^{99}	0	10^{99}	10^{99}	10^{99}	0
5	4.5	10^{99}	10^{99}	10^{99}	10^{99}	10^{99}	10^{99}
6	2.7	10^{99}	10^{99}	0	0	10^{99}	10^{99}

In order to adapt the data to an electric vehicle scheduling problem, a charging station is required. Therefore, a station is added at the same location of the depot, not only because it is the most common approach, but also because it facilitates the determination of the data regarding the new node.

Finally, the entire data is prepared and ready to be used, with a total of 102 nodes grouping as stated in Table 5.11.

Table 5.11: Adapted data size

Node	Quantity
Depots	1
Charging Stations	1
Service Trips	100

5.3.2 Input Parameters

After the interpretation and adjustment of the data to the algorithm, there is the establishment of the input parameters.

Several attempts to run the algorithm inputting the 100 service trips are carried out. However, since CPLEX is not efficient for larger instances, it leads to either a software break down or to a manually interruption due to its long computational time, not reaching a final result on both situations. Therefore, smaller scenarios are generated and analyzed.

In order to establish a fair comparison, the selection of service trips is processed by STCP' schedules. This way it is possible to know the number of vehicles for both cases (with and without EV's), which would not happen if the service trips were chosen randomly.

Since the computational time increases substantially for a number of service trips higher than 50, the two scenarios are created assuring a lower number of these trips. Therefore, *Scenario 1* is constituted by the service trips belonging to the STCP' schedules of *Vehicle 1*, *Vehicle 2* and *Vehicle 3*, marked in orange in the Table A.1, while *Scenario 2* aggregates the service trips belonging to the STCP' schedules of *Vehicle 4* and *Vehicle 5*, in blue in the Table A.1. The schedule of *Vehicle 6* is not used since its union with each of the other two groups causes a set of service trips higher than 50 and its analysis alone is not of interested for the study.

For both scenarios, the input parameters are therefore settled as observed in the Table 5.12.

Table 5.12: Input parameters regarding the data size

Parameter	*Scenario 1*	*Scenario 2*
h	1	1
f	1	1
n	46	37

The d_{ij} and e_{ij} matrices are defined based on the Table 5.9 and Table 5.10, respectively, where the values are repeated the number of times a trajectory is executed in a service trip. Its construction is already described in the Section 4.3, resulting in 2 matrices 48x48 for *Scenario 1* (46 service trips, plus a depot and a charging station) and 2 matrices 39x39 for *Scenario 2* (37 service trips, plus a depot and a charging station).

As for the d_i and e_i vectors, which are represented by Table 5.6 and Table 5.8, respectively, the process is the same, resulting in two vectors of size 48 for the first scenario and two vectors of size 39 for the second.

The starting times α for each scenario is provided by Table A.1, imputed as a vector of size 46 for *Scenario 1* and another of size 37 for *Scenario 2*.

As for the costs parameters, the values for both scenarios are described in the Table 5.13. Being the average salary of drivers in Portugal around 1000 € per month, an approximated value for the operational cost c_t is calculated, assuming they work 40h a week and 4 weeks a month.

Table 5.13: Input Parameters regarding Costs

Parameter	Value
c_k [1]	500,000.00 [euros/vehicle]
c_t	0.11 [euros/min]
c_e	0.14 [euros/kWh]

The battery capacity E_0 is set in both cases as 200 kWh, the real value of the STCP electric vehicles according to *Caetanobus - Fabricação de Carroçarias SA*.

[1] Value based on the base price of the STCP tender (STCP (2017))

The k and p parameters are the most difficult to define. Because they are related to the output and the result is obviously unknown initially, there is not an optimal value for both. However, it is possible to determine an upper bound. In the worst case, each service trip would be completed by a different vehicle. This scenario would lead to a k equal to the number of service trips and a p equal to 3 (DH trip from depot to service trip, from service trip to charging station and back to the depot). In the best case, we would only need one vehicle ($k = 1$), but p could go until $2n+1$ (a charging event between each service trip and before returning to the depot). For that reason, k should be smaller or equal to n and p to $2n+1$.

Knowing the solution, in this study, k takes the value of twice the number of vehicles needed for the current solution of each scenario. The inputs that do not regard the graph are stated in the Table 5.14.

Table 5.14: Input parameters regarding the formulation

Parameter	*Scenario 1*	*Scenario 2*
k	6	4
p	20	20
r_h	10	10
q_f	10	10

5.3.3 Settings and Preprocessing

All the tests are performed with the standard settings of CPLEX software.

When running the algorithm, due to the big amount of data entering as an input, the computational time becomes a major limitation. Therefore, additional optimization of the script is made.

Since the general optimization of the code is already contemplated, the additional possible improvements only regard the adaptation of the formulation to this case.

When ran with a small instance (just 6 of the 100 nodes), the CPLEX tool *Profiler* allows to conclude that the most time-consuming constraints are the ones regarding the charging station capacity. Since this is not a current problem of STCP (remember that they have 15 chargers for 15 electric vehicles), for bigger instances these restrictions are relaxed, eliminating Equation 3.19 and Equations 4.14 to 4.19.

Also, because of the considerable gap verified previously between the earliest service trip and the end of the latest, the two constraints that assure that a vehicle arrives its depot before its first trip of the next day become unnecessary. This leads to the elimination of two more constrains, represented by Equation 4.8 and Equation 4.10.

5.3.4 Results

The results obtained by the developed formulation are present in the Table B.1 and Table B.3 for *Scenario 1* and *Scenario 2*, respectively. Since the decision variable x_{ij}^{kp} is boolean, only the ones that take the value of one are of interest, symbolizing the deadheading trips to comply. All the other possible combinations are not present due to its considerable extension.

For a better interpretation of the result, the service trips are organized by schedules, as observed in the Table 5.15 for *Scenario 1* and 5.16 for *Scenario 2*. Just as previously, the depots and charging stations are highlighted and represented by **H** and **F**, respectively.

Table 5.15: CPLEX Solution for *Scenario 1*

Vehicle	Sequence
1	**H** - 88 - 2 - 46 - 7 - 52 - 95 - **F** - 22 - 67 - **F** - 41 - 86 - 99 - **F** - **H**
2	**H** - 3 - 48 - 9 - 54 - 13 - 58 - **F** - 27 - 72 - 32 - 77 - 37 - 82 - **F** - **H**
3	**H** - 87 - 1 - 44 - 4 - 49 - 94 - **F** - 90 - *F* - 39 - 84 - 42 - 100 - **F** - **H**
4	**H** - **H**
5	**H** - **H**
6	**H** - 91 - 43 - **F** - 15 - 60 - 19 - 64 - 24 - 69 - 29 - 74 - 34 - 79 - **F** - **H**

Table 5.16: CPLEX Solution for *Scenario 2*

Vehicle	Sequence
1	**H** - 89 - **F** - 17 - 62 - 21 - 66 - 26 - 71 - 31 - 76 - **H**
2	**H** - 92 - 45 - 6 - 51 - 11 - 56 - **F** - 23 - 68 - 28 - 73 - 33 - 78 - 38 - 83 - 85 - **F** - **H**
3	**H** - 5 - 50 - 10 - 55 - 14 - 59 - 18 - 63 - **F** - 36 - 81 - 40 - 85 - 98 -**F** - **H**
4	**H** - **H**

In the *Scenario 1*, as observed in the Table 5.15, both *Vehicle 4* and *Vehicle 5* stay at the depot. Since only those leaving the depot are necessary, it can be concluded that 4 vehicles are required to conclude the 46 service trips: *Vehicle 1*, *Vehicle 2*, *Vehicle 3* and *Vehicle 6*. Similarly, in the *Scenario 2*, observed in the Table 5.16, *Vehicle 4* is not needed. From now on, only the ones needed will be referenced.

Obeying to the basics of the VSP, all schedules start and end at the allocated depot for that vehicle and each service trip completed according to its demands.

The resulting values of the objective function is stated in the Table 5.17, along with its computational time.

Table 5.17: Objective function values and computational time for both scenarios

Parameter	*Scenario 1*	*Scenario 2*
Objective Function	2,000,052.260 €	1,500,040.056 €
Computational Time	53min06s	22min56s

It is important to emphasize that an optimal value is not reached in either scenarios. Integrality tolerances help CPLEX discriminate small values with meaning in the model from those resulting of mathematical optimization or data rounding. Therefore, it defines the maximum deviation that an integer variable can take from the integer value. Since the present formulation works with significantly large data, this does not interferes with the quality of the solution, decreasing the computational time. At the default settings, this parameter takes the value of 1e-05.

The energy level for each node is also provided and described for each scenario in the Table B.2 and Table B.4.

5.3.5 Discussion

Once again, for a better visual comprehension, the solution is arranged by time for *Scenario 1*, as observed in the Figures 5.7, 5.8, 5.9 and 5.10.

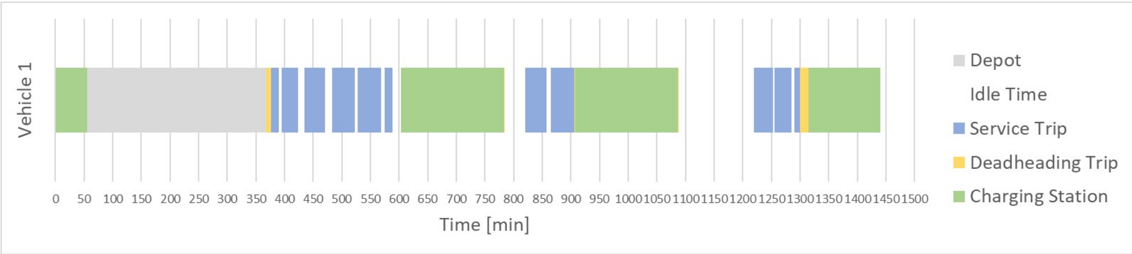

Figure 5.7: Schedule of *Vehicle 1* of *Scenario 1*

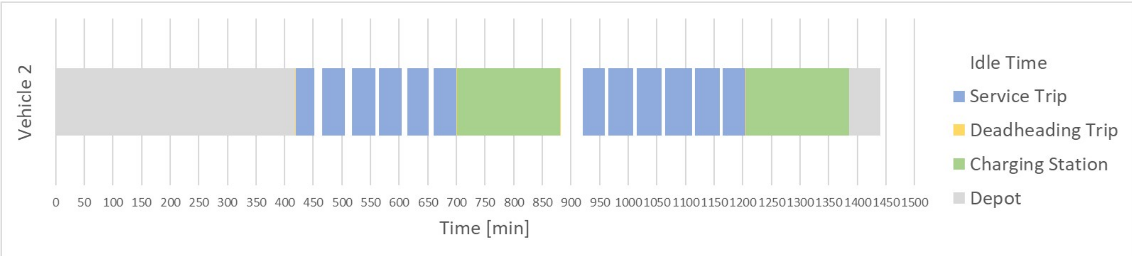

Figure 5.8: Schedule of *Vehicle 2* of *Scenario 1*

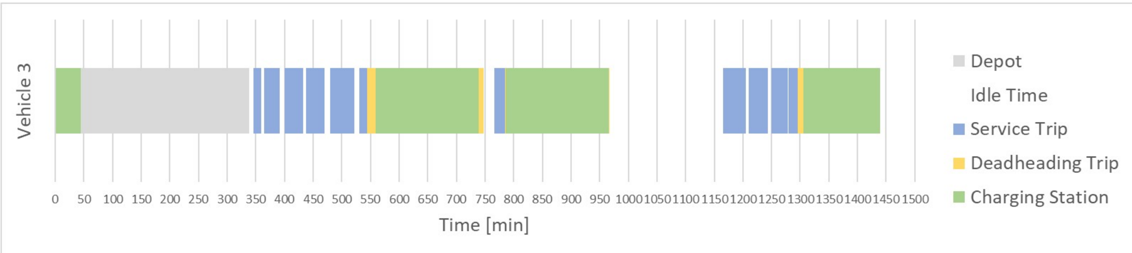

Figure 5.9: Schedule of *Vehicle 3* of *Scenario 1*

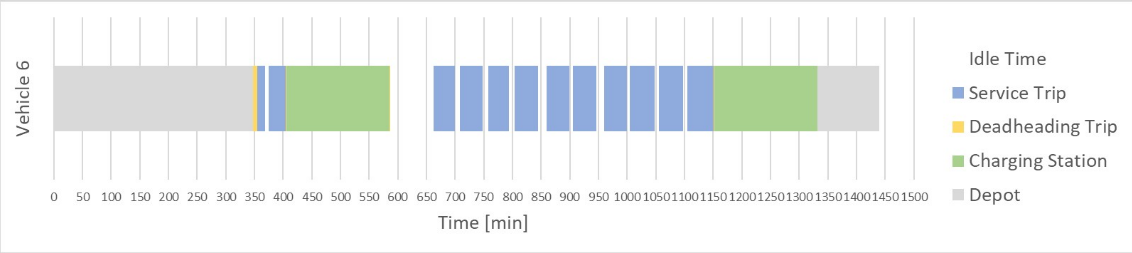

Figure 5.10: Schedule of *Vehicle 6* of *Scenario 1*

The same approach is applied for *Scenario 2*, as observed in the Figures 5.11, 5.12 and 5.13.

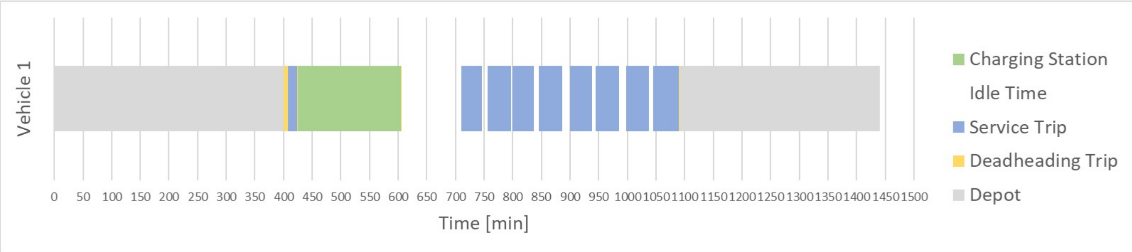

Figure 5.11: Schedule of *Vehicle 1* of *Scenario 2*

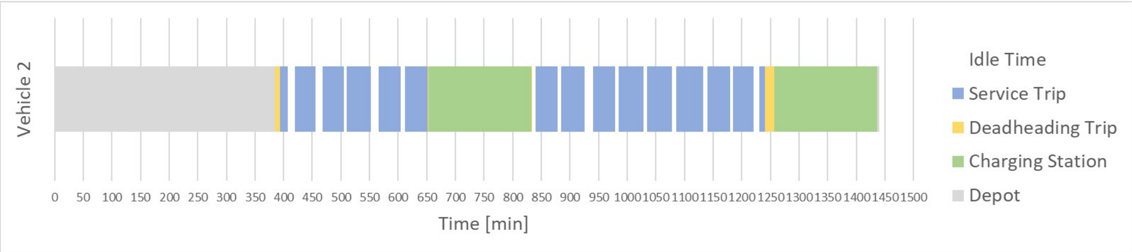

Figure 5.12: Schedule of *Vehicle 2* of *Scenario 2*

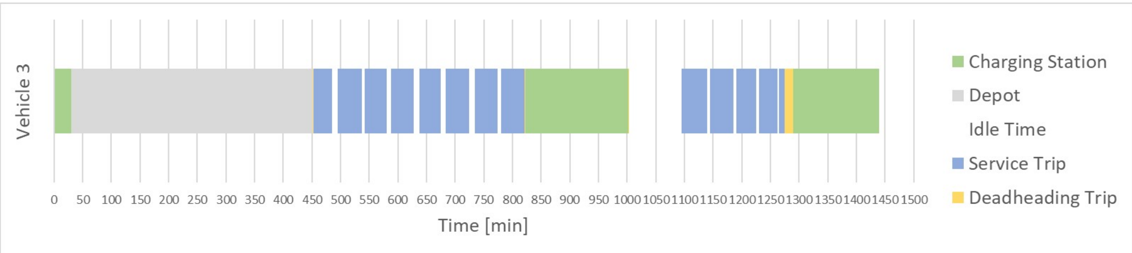

Figure 5.13: Schedule of *Vehicle 3* of *Scenario 2*

When compared to the current solution, the electric vehicle scheduling forces the purchase of one more vehicle in both cases. The need for additional vehicles meets the expectations once the lower autonomy of electric vehicles requires more empty trips to the charge, consuming operational time.

The analysis of the time spent at each task on both STCP and CPLEX solution for *Scenario 1* results in the Tables 5.18 and 5.19, respectively.

Table 5.18: Time per task on the STCP solution for *Scenario 1* (in minutes)

Task	*Vehicle 1*	*Vehicle 2*	*Vehicle 3*	Total	%
Service Trips	162	698	696	1556	36 %
Depot	1219	481	513	2213	51 %
DH Trips	23	32	33	88	2 %
Idle Time	36	229	198	463	11 %
Total	1440	1440	1440	4320	100 %

Table 5.19: Time per task on the CPLEX solution for *Scenario 1* (in minutes)

Task	*Vehicle 1*	*Vehicle 2*	*Vehicle 3*	*Vehicle 6*	Total	%
Service Trips	323	484	299	450	1556	27%
Depot	313	473	293	455	1534	27%
DH Trips	41	4	42	11	98	2%
Idle Time	223	119	266	164	772	13%
Charging Station	540	360	540	360	1800	31%
Total	1440	1440	1440	1440	5760	100%

As expected, the total time reserved for service trips is the same on both solutions, since it has a fixed and mandatory duration. However, its percentage relatively to the total operational time decreases. Being this the main function of the vehicles, it is concluded that the vehicles exploitation deteriorates with the replacement to electric ones.

Despite the additional vehicle, the time consumed by DH trips does not increase substantially, contrarily to the idle time. Observing the previous figures, it is clear that the larger periods of idle time is after a charging event, since the only ones minimized are between service trips. The deadheading trip between the charging station and the beginning of the next task is placed before the idle time to guarantee the fulfilment of the schedule, preventing delays and waiting at the starting place. However, this is a decision that it is up to the client to make, having the possibility to wait at the depot instead.

Finally, the charging events occupy around 31% of the operational time of the vehicles. This along with the low percentage of time consumed by deadheading trips suggest that the *bottleneck* of the process is not the lower autonomy of the vehicle, characterized by the additional charging trip itself, but the large amount of time it takes to charge the battery, decreasing the vehicle possible operational window.

For *Scenario 2*, the same analysis is carried out, comparing its time per task with the corresponding vehicles at the STCP' solution. Table 5.20 refers to the STCP' solution for *Scenario 2*, while Table 5.21 describes for CPLEX solution for the same scenario.

Table 5.20: Time per task on the STCP solution for *Scenario 2* (in minutes)

Task	*Vehicle 4*	*Vehicle 5*	Total	%
Service Trips	692	666	1358	47%
Depot	535	584	1119	39%
DH Trips	23	23	46	2%
Idle Time	190	167	357	12%
Total	1440	1440	2880	100 %

Table 5.21: Time per task on the CPLEX solution for *Scenario 2* (in minutes)

Task	*Vehicle 1*	*Vehicle 2*	*Vehicle 3*	Total	%
Service Trips	336	547	475	1358	31%
Depot	749	389	422	1560	36%
DH Trips	11	25	18	54	1%
Idle Time	164	119	165	448	10%
Charging Station	180	360	360	900	21%
Total	1440	1440	1440	5760	100%

Once again, the total time spent on service trips is the same on both solutions, but its percentage decreases with the replacement for EV. As for the DH trips, its value does not increase substantially, highlighting afresh the higher influence of the charging time, along with the percentage of time consumed by the charging events.

A further analysis to the battery level evolution is carried out. As observed in the Figure C, for the *Scenario 1*, and Figure C, for the *Scenario 2*, the value of the energy level of each vehicle never surpasses its limits, avoiding extreme operational points. Although it helps increasing the battery lifespan, it reduces the vehicle capacity by 40%. In the *Vehicle 6* of the *Scenario 1*, for example, the limitation of charging only 80% of the battery maximum capacity (E_0) forces a second trip to the charging station that could probably be avoided if it charged the entire amount (i.e. 100%).

Observing the final value of the objective function, the cost of the vehicles seems more preponderant on both scenarios. However, it is necessary to take into account that this value is an one-time purchase, while the remainder is a daily expense. Since generally the clients prefer to have less vehicles, a high value corresponding to its acquisition is considered. However, in same cases, it may be preferable to have a higher initial investment leading to less energy or operational costs. Thus, in a more meticulous approach, c_k could take the value of the cost of a vehicle per day, dividing its total cost by the expected lifespan by days.

Furthermore, it is important to remind that the energy and operational costs of each service trip are not included in the objective function, since they are fixed and independent of the solution. Therefore, for a daily expenses calculation, that expense would have to be subsequently added.

In order to confirm the results interpretation described above, a test to the influence of the charging time d_f and the vehicle autonomy, represented by the maximum capacity of the battery of the vehicle E_0, in the *Scenario 1* is carried out. In the first run, the charging time is changed from 180 min to a third of the time (60 min). Contrarily, in the second test, the charging time returns to its initial value of 180 min, while the battery maximum capacity becomes 600 kWh, three times higher than the actual one. In both tests, the parameter k takes the value of 4 to accelerate the acquisition of solution. The results are summarized in the Table 5.22.

Table 5.22: Results of the tests to the influence of the two limitations of EV

Parameter in study	Value	Objective Function	Number of vehicles required
Charging Time [min]	60	1,500,052.260 €	3
Maximum Capacity [kWh]	600	2,000,052.260 €	4

When the charging time improves to a third of its initial time, the number of vehicles needed is the same as the STCP' solution. However, this could not be directly assumed to be the main limitation as a higher autonomy could lead to lower visits to the charging station, turning this charging time a night event without any impact on the daily operational time. Therefore, a test to the battery maximum capacity is carried out. A higher value of this parameter is interpreted as a higher autonomy, leading to less charging events. When executed, the result still demands a higher number of vehicles compared to the STCP' solution. Taken together the previous results, the confirmation of the past conclusions becomes obvious: the charging time has a higher impact on the electric vehicle scheduling problem than the vehicle autonomy itself.

It is important to highlight that both solutions are non-optimal.

When repeated for the *Scenario 2*, with $k = 3$, the results of the test are not that satisfactory. Due to the lower number of service trips in this scenario and the higher percentage of time consumed by these tasks, it gets harder to obtain improvements. Therefore, lowering the charging time to a third or increasing the autonomy three times do not reduce the number of vehicles needed, not adding any value to the study. However, it also does not contradict the previous conclusions.

5.3.6 Performance

The experiment is carried out using a *Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz* processor with a RAM of 16,0 GB.

Being the most challenging part the setting of the initial parameter k value, a test to its influence in the script's efficiency is also performed, using *Scenario 1*.

Knowing the number of vehicles required to complete the 46 service trips, a second experience is carried out, modifying only the value k to the precise number of vehicles needed. Its results are reported in the Table 5.23.

Table 5.23: CPLEX results of the *Scenario 1* with $k = 4$

Parameter	Value
Objective Function	2,000,052.26 €
Computational Time	26min01s

Comparing with the computational time of the first experience, an improvement of nearly 50% is confirmed, as expected. Having a larger *k* value leads to more possible combinations that have to be experimented which contribute to a higher computational time. Being this one of the major limitations of the operational research, this initial input becomes an important step in the processing of the model.

Once again, the value of the objective function obtained is not optimal, due to the integrality tolerance. The results are, once again, organized in the Table 5.24.

Table 5.24: CPLEX solution of the *Scenario 1* with $k = 4$

Vehicle	Sequence
1	**H** - 3 - 48 - 9 - 54 - 13 - 58 - **F** - 29 - 74 - 34 - 79 **H**
2	**H** - 88 - 2 - 46 - 7 - 52 - 95 - **F** - 22 - 67 - 27 - 72 - 32 - 77 - 37 - 82 - **F** - **H**
3	**H** - 87 - 1 - 44 - 4 - 49 - 94 - *F* - 90 - *F* - 41 - 86 - 99 - **F** - **H**
4	**H** - 91 - 43 - *F* - 15 - 60 - 19 - 64 - 24 - 69 - *F* - 39 - 84 - 42 - 100 - **F** - **H**

Since none of the schedules consists of *false* deadheading trips between the same depot, it is observed that all 4 vehicles are required in order to complete all the tasks, confirming the initial results.

Chapter 6

Conclusions and Future Work

This project describes a new formulation for the Electric Vehicle Scheduling Problem. Nowadays, EV are a very debated topic, turning this study into a pertinent contribution to this research area. At an early stage of this paper, a mathematical formulation is described, aiming to minimize the total cost regarding the vehicle scheduling. Adapted to the electric transportation, it gives special attention to the new characteristics that this entails, such as lower autonomy leading to consequent charging events in the middle of the schedule and a battery level monitoring. Also, the possibility of multiple depots and charging stations is considered. In a later stage, the development of this model into the CPLEX software was carried out. Although this is an adapted computer program for optimization problems, several modifications to the initial formulation are mandatory in order to answer to the software limitations and to improve its efficiency. Finally, real data is experimented, leading to new improvements and adaptions to the case in study, with the main goal of decreasing the computational time. The programming language selected was *Optimization Programming Language* due to its simple and evident interpretation.

Analyzing the results, the model's feasibility is confirmed, achieving coherent and valid solutions. Comparisons between the current STCP solution and the one defined by CPLEX allow to understand that the impact of the charging time in the eVSP is higher than the vehicle autonomy. The adoption of electric vehicles requires a larger investment not only because of its higher cost, but also due to the its lower actual operational time, leading to the demand of a larger number of vehicles.

Fast charging would improve the efficiency of EV since it could be incorporated in every terminus of a line, saving deadheading trips and charging time. The formulation developed can solve cases with this charging mode, but the large number of charging stations would increase substantially the computational time.

As for the performance of the script, a test to the influence of k was carried out, highlighting the importance of the setting of its initial value. A larger k leads to a higher computational time, but a too low value may not be sufficient to accomplish all the service trips, transforming the formulation into a problem with no feasible solution.

Being the charging time the main drawback of the EV, the possibility of charging only the amount of time needed for the energy level to reach its maximum capacity was considered. However, since the charging time is not linear, charging faster at the beginning while decreasing exponentially its rate with time, the benefits of this additional feature were called into question when compared with the incremental computational time. However, it would be interesting to study its impact in further work.

The adaption of each schedule to different types of vehicles would also be of interest, since not all itinerary demand the same autonomy, resulting in a new parameter on the decision variable. This would, however, have the disadvantage of increasing its computational time.

Finally, the development of an adapted algorithm for the formulation presented would permit the analyze of larger instances with lower computational time, achieving more realistic results.

In conclusion, despite the improvements opportunities, the present dissertation meets its initial objectives, finding good solutions for the electric vehicle scheduling problem.

Bibliography

A. Pepin, G. Desaulniers, A. H. e. a. (2008). A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, 12:17–30.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

C. Xu, H. Xuan, B. L. F. S. (2010). Vehicle scheduling problem for the single-depot based on genetic algorithms. In *2010 International Conference of Information Science and Management Engineering*, volume 1, pages 254–256. IEEE.

D. Henderson, S. Jacobson, A. J. (2003). The theory and practice of simulated annealing. In F. Glover, G. A. K., editor, *Handbook of Metaheuristics*, pages 287–319. Springer.

D. Pisinger, S. R. (2010). Large neighborhood search. In M. Gendreau, J. P., editor, *Handbook of Metaheuristics*, pages 399–419. Springer US.

Deb, K. (1999). An introduction to genetic algorithms. *Sadhana*, 24:293–315.

E. L. Lawler, D. E. W. (1966). Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719.

G. Desaulniers, J. Desrosiers, S. S. (2011). Cutting planes for branch-and-price algorithms. *Networks*, 58(4):301–310.

H. Wang, J. S. (2007). Heuristics approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation*, 190:1237–1249.

J. Adler, P. M. (2017). The vehicle scheduling problem for fleets with alternative-fuel vehicles. *Transportation Science*, 51(2):441–456.

J. Desrosiers, M. L. (2005). A primer in column generation. In G. Desaulniers, J. Desrosiers, M. S., editor, *Column Generation*, pages 1–32. Springer US.

Lemaréchal, C. (2001). Lagrangian relaxation. In M.Jünger, D. N., editor, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2241, pages 112–156. Springer Berlin Heidelberg.

M. Dorigo, L. M. G. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

M. Rogge, E. van der Hurk, A. L. D. U. S. (2018). Electric bus fleet size and mix problem with optimization of charging infrastructure. *Applied Energy*, 211:282–295.

Mitchell, J. (1988). Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of Applied Optimization*.

P. Hansen, N. M. (2001). Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467.

P. van Laarhoven, E. A. (1987). Simulated annealing. In P.J. van Laarhoven, E. A., editor, *Simulated Annealing: Theory and Applications*, pages 7–15. Springer Netherlands.

S. Bunte, N. K. (2009). An overview on vehicle scheduling models. *Public Transport*, 1:299–317.

S. Kulkarni, M. Krishnamoorthy, A. R. A. T. E. R. P. (2018). A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem. *Transportation Research Part B: Methodological*, 118:457–487.

STCP, S. (2017). Portuguese procedure announcement n.º n.º 3403/2017.

STCP, S. (2018). Indicadores-chave. `https://www.stcp.pt/pt/institucional/governo-societario/indicadores-chave/`. (Accessed: 07.06.2019).

T. Paul, H. Y. (2014). Operation and charging scheduling of electric buses in a city bus route network. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1:2780–2786.

Transport&Environment (2018). Electric buses arrive on time - marketplace, economic, technology, environmental and policy perspectives for fully electric buses in the eu. Technical report, Transport&Environment.

V. Franco, M. Kousoulidou, M. M. e. a. (2013). Road vehicle emission factors development: A review. *Atmospherics Environment*, 70:84–97.

Z. Chao, C. X. (2013). Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. *Procedia - Social and Behavioral Sciences*, 96:2725–2736.

Z. Wan, D. Sperling, Y. W. (2015). China's electric car frustrations. *Transportation Research Part D: Transport and Environment*, 34:116–121.

Zheng, Y., Dong, Z. Y., Xu, Y., Meng, K., Zhao, J. H., and Qiu, J. (2014). Electric vehicle battery charging/swap stations in distribution systems: Comparison study and optimal planning. *IEEE Transactions on Power Systems*, 29(1):221–229.

Appendix A

Detailed Information Regarding the Service Trips

Table A.1: Detailed Information Regarding the Service Trips

Number	Route ID	Starting Node	Ending Node	Starting Time	Duration
1	1	25	38	365	27
2	1	25	38	395	28
3	1	25	38	420	32
4	1	25	38	438	32
5	1	25	38	453	32
6	1	25	38	468	37
7	1	25	38	483	40
8	1	25	38	500	40
9	1	25	38	518	40
10	1	25	38	542	39
11	1	25	38	566	39
12	1	25	38	590	37
13	1	25	38	614	37
14	1	25	38	638	37
15	1	25	38	662	37
16	1	25	38	686	37
17	1	25	38	710	36
18	1	25	38	734	40
19	1	25	38	758	36
20	1	25	38	780	37
21	1	25	38	800	37
22	1	25	38	820	37
23	1	25	38	840	39
24	1	25	38	860	39
25	1	25	38	880	39
26	1	25	38	900	39
27	1	25	38	920	39
28	1	25	38	940	39

Continued on next page

Table A.1 – continued from previous page

Number	Route ID	Starting Node	Ending Node	Starting Time	Duration
29	1	25	38	960	40
30	1	25	38	980	40
31	1	25	38	998	40
32	1	25	38	1015	43
33	1	25	38	1035	43
34	1	25	38	1055	43
35	1	25	38	1075	45
36	1	25	38	1095	45
37	1	25	38	1117	43
38	1	25	38	1140	40
39	1	25	38	1165	40
40	1	25	38	1190	35
41	1	25	38	1220	33
42	1	25	38	1250	28
43	2	38	25	375	30
44	2	38	25	400	33
45	2	38	25	420	36
46	2	38	25	435	36
47	2	38	25	450	37
48	2	38	25	465	40
49	2	38	25	480	42
50	2	38	25	495	42
51	2	38	25	510	42
52	2	38	25	528	41
53	2	38	25	545	40
54	2	38	25	565	40
55	2	38	25	588	40
56	2	38	25	612	40
57	2	38	25	636	40
58	2	38	25	660	40
59	2	38	25	684	40
60	2	38	25	708	40
61	2	38	25	732	40
62	2	38	25	756	41
63	2	38	25	780	41
64	2	38	25	804	41
65	2	38	25	825	41
66	2	38	25	845	41
67	2	38	25	865	41
68	2	38	25	885	41
69	2	38	25	905	41
70	2	38	25	925	41
71	2	38	25	945	41
72	2	38	25	965	43
73	2	38	25	985	43

Continued on next page

Table A.1 – continued from previous page

Number	Route ID	Starting Node	Ending Node	Starting Time	Duration
74	2	38	25	1005	43
75	2	38	25	1025	46
76	2	38	25	1045	45
77	2	38	25	1065	47
78	2	38	25	1085	47
79	2	38	25	1105	46
80	2	38	25	1125	44
81	2	38	25	1145	41
82	2	38	25	1165	40
83	2	38	25	1185	36
84	2	38	25	1210	34
85	2	38	25	1230	33
86	2	38	25	1255	30
87	3	800307	25	346	14
88	3	800307	25	376	14
89	3	800307	25	408	16
90	3	800307	25	766	19
91	4	800307	38	355	14
92	4	800307	38	393	14
93	4	800307	38	422	17
94	5	25	3	530	14
95	5	25	3	575	13
96	5	25	3	1175	15
97	5	25	3	1230	11
98	5	25	3	1265	10
99	5	25	3	1290	10
100	6	38	800307	1280	16

End of Table

Appendix B

CPLEX results

Table B.1: Values of x_{ij}^{kp} for *Scenario 1*

Number	Number	nodes (size 48)	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
2	46	48	42	1	3	1
79	F	47	2	6	14	1
74	34	46	33	6	12	1
69	29	45	32	6	10	1
64	24	44	31	6	8	1
58	F	43	2	2	7	1
46	7	42	35	1	4	1
95	F	41	2	1	7	1
42	100	40	38	3	13	1
88	2	39	48	1	2	1
100	F	38	2	3	14	1
84	42	37	40	3	12	1
52	95	36	41	1	6	1
7	52	35	36	1	5	1
39	84	34	37	3	11	1
34	79	33	47	6	13	1
29	74	32	46	6	11	1
24	69	31	45	6	9	1
19	64	30	44	6	7	1
13	58	29	43	2	6	1
9	54	28	16	2	4	1
82	F	27	2	2	14	1
77	37	26	13	2	12	1
67	F	25	2	1	10	1
60	19	24	30	6	6	1
48	9	23	28	2	3	1
99	F	22	2	1	14	1
41	86	21	18	1	12	1
90	F	20	2	3	9	1

Continued on next page

Table B.1 – continued from previous page

Number	Number	nodes (size 48)	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
91	43	19	15	6	2	1
86	99	18	22	1	13	1
72	32	17	12	2	10	1
54	13	16	29	2	5	1
43	F	15	2	6	3	1
3	48	14	23	2	2	1
37	82	13	27	2	13	1
32	77	12	26	2	11	1
27	72	11	17	2	9	1
22	67	10	25	1	9	1
15	60	9	24	6	5	1
4	49	8	5	3	5	1
94	F	7	2	3	7	1
87	1	6	3	3	2	1
49	94	5	7	3	6	1
44	4	4	8	3	4	1
1	44	3	4	3	3	1
F	39	2	34	3	10	1
F	41	2	21	1	11	1
F	90	2	20	3	8	1
F	27	2	11	2	8	1
F	22	2	10	1	8	1
F	15	2	9	6	4	1
F	H	2	1	6	15	1
F	H	2	1	3	15	1
F	H	2	1	2	15	1
F	H	2	1	1	15	1
H	88	1	39	1	1	1
H	91	1	19	6	1	1
H	3	1	14	2	1	1
H	87	1	6	3	1	1
H	H	1	1	5	1	1
H	H	1	1	4	1	1

End of Table

Table B.2: Values of E_i^{kp} for *Scenario 1*

Number	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
2	48	1	3	144,3
79	47	6	14	59,5
74	46	6	12	78,7
69	45	6	10	97,9
64	44	6	8	117,1
58	43	2	7	97,9
46	42	1	4	134,5
95	41	1	7	112,4
42	40	3	13	126,9
88	39	1	2	153,7
100	38	3	14	123,1
84	37	3	12	136,3
52	36	1	6	115,3
7	35	1	5	125,1
39	34	3	11	146,1
34	33	6	13	69,3
29	32	6	11	88,5
24	31	6	9	107,7
19	30	6	7	126,9
13	29	2	6	107,7
9	28	2	4	126,9
82	27	2	14	97,9
77	26	2	12	117,1
67	25	1	10	136,3
60	24	6	6	136,3
48	23	2	3	136,3
99	22	1	14	133,4
41	21	1	12	146,1
90	20	3	9	153,7
91	19	6	2	153,9
86	18	1	13	136,3
72	17	2	10	136,3
54	16	2	5	117,1
43	15	6	3	144,1
3	14	2	2	146,1
37	13	2	13	107,7
32	12	2	11	126,9
27	11	2	9	146,1
22	10	1	9	146,1
15	9	6	5	146,1
4	8	3	5	125,1
94	7	3	7	112,4
87	6	3	2	153,7

Continued on next page

Table B.2 – continued from previous page

Number	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
49	5	3	6	115,3
44	4	3	4	134,5
1	3	3	3	144,3
F	2	3	10	160
F	2	1	11	160
F	2	3	8	160
F	2	2	8	160
F	2	1	8	160
F	2	6	4	160
F	2	6	15	160
F	2	3	15	160
F	2	2	15	160
F	2	1	15	160
H	1	1	1	160
H	1	6	1	160
H	1	2	1	160
H	1	3	1	160
H	1	5	1	40
H	1	4	1	40

End of Table

Table B.3: Values of x_{ij}^{kp} for *Scenario 2*

Number	Number	nodes (size 39)	nodes (size 39)	vehicles (size 4)	orders (size 20)	Value
5	50	39	31	3	2	1
83	85	38	30	2	16	1
78	38	37	25	2	14	1
73	33	36	24	2	12	1
68	28	35	23	2	10	1
63	F	34	2	3	9	1
59	18	33	28	3	7	1
55	14	32	22	3	5	1
50	10	31	29	3	3	1
85	F	30	2	2	17	1
10	55	29	32	3	4	1
18	63	28	34	3	8	1
23	68	27	35	2	9	1
89	F	26	2	1	2	1
38	83	25	38	2	15	1
33	78	24	37	2	13	1
28	73	23	36	2	11	1
14	59	22	33	3	6	1
11	56	21	14	2	6	1

Continued on next page

Table B.3 – continued from previous page

Number	Number	nodes (size 48)	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
92	45	20	9	2	2	1
85	98	19	12	3	14	1
76	H	18	1	1	11	1
71	31	17	6	1	9	1
66	26	16	5	1	7	1
62	21	15	4	1	5	1
56	F	14	2	2	7	1
51	11	13	21	2	5	1
98	F	12	2	3	15	1
40	85	11	19	3	13	1
81	40	10	11	3	12	1
45	6	9	8	2	3	1
6	51	8	13	2	4	1
36	81	7	10	3	11	1
31	76	6	18	1	10	1
26	71	5	17	1	8	1
21	66	4	16	1	6	1
17	62	3	15	1	4	1
F	23	2	27	2	8	1
F	36	2	7	3	10	1
F	17	2	3	1	3	1
F	H	2	1	3	16	1
F	H	2	1	2	18	1
H	5	1	39	3	1	1
H	89	1	26	1	1	1
H	92	1	20	2	1	1
H	H	1	1	4	1	1

End of Table

Table B.4: Values of E_i^{kp} for *Scenario 2*

Number	nodes (size 39)	vehicles (size 4)	orders (size 20)	Value
5	3	2	146,1	
83	2	16	78,7	
78	2	14	97,9	
73	2	12	117,1	
68	2	10	136,3	
63	3	9	78,7	
59	3	7	97,9	
55	3	5	117,1	
50	3	3	136,3	
85	2	17	75,8	
10	3	4	126,9	
18	3	8	88,5	
23	2	9	146,1	
89	1	2	67,9	
38	2	15	88,5	
33	2	13	107,7	
28	2	11	126,9	
14	3	6	107,7	
11	2	6	115,5	
92	2	2	153,9	
85	3	14	117,1	
76	1	11	78,7	
71	1	9	97,9	
66	1	7	117,1	
62	1	5	136,3	
56	2	7	105,7	
51	2	5	124,9	
98	3	15	114,2	
40	3	13	126,9	
81	3	12	136,3	
45	2	3	144,1	
6	2	4	134,7	
36	3	11	146,1	
31	1	10	88,5	
26	1	8	107,7	
21	1	6	126,9	
17	1	4	146,1	
F	2	8	160	
F	3	10	160	
F	1	3	160	
F	3	16	160	
F	2	18	160	
H	3	1	160	

Continued on next page

Table B.4 – continued from previous page

Number	nodes (size 48)	vehicles (size 6)	orders (size 20)	Value
H	1	1	74,2	
H	2	1	160	
H	4	1	40	
End of Table				

Appendix C

Evolution of the energy level per vehicle

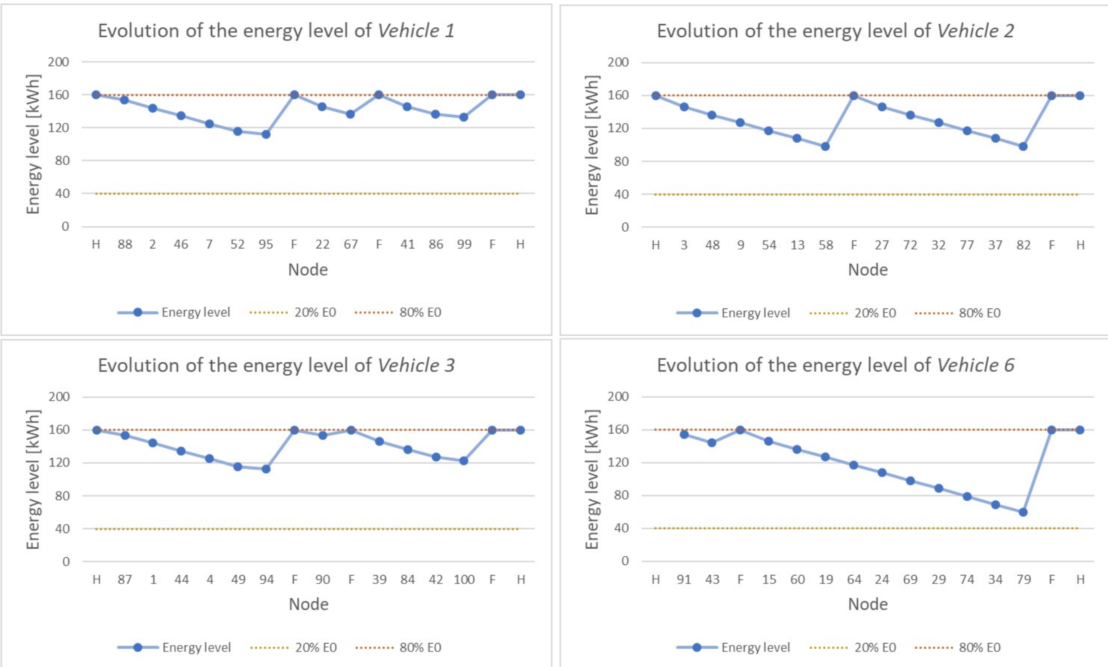

Figure C.1: Evolution of the energy level on the *Scenario 1*

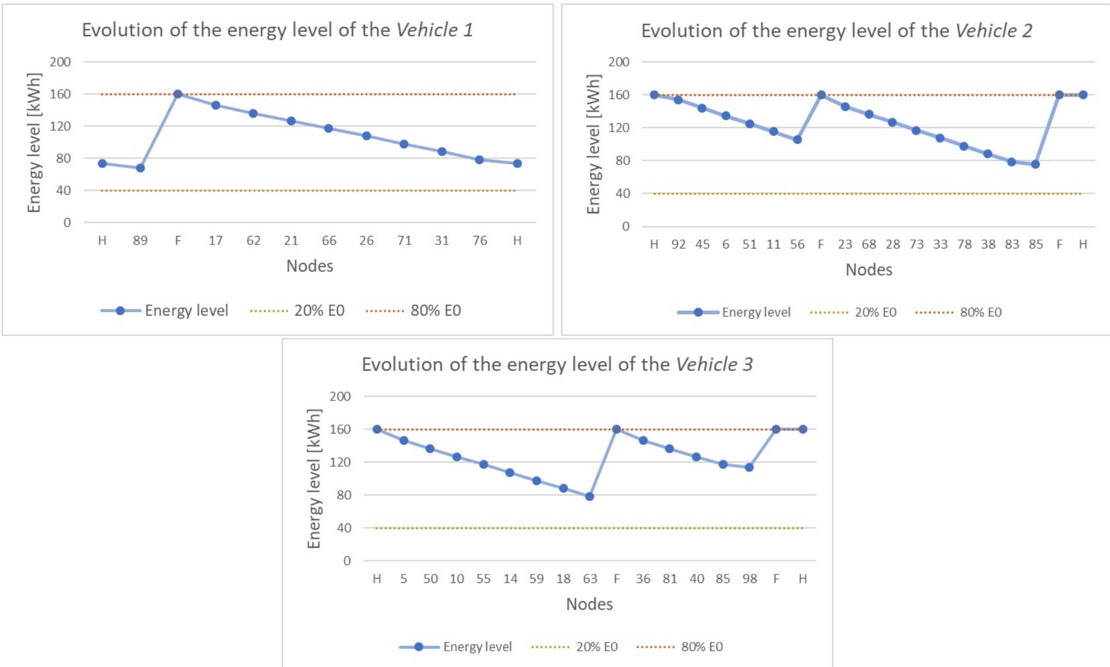

Figure C.2: Evolution of the energy level on the *Scenario 2*