An heuristic approach for the Green Vehicle Routing Problem

Paulo Roberto de Oliveira da Costa, B.Sc.
Stefano Mauceri, M.Sc.

September 3, 2016

A thesis submitted to University College Dublin in part fulfilment of the requirements of the degree of Master of Science in Business Analytics

Michael Smurfit Graduate School of Business,

University College Dublin

September, 2016

Supervisor: Dr.Paula Carroll

Head of School: Professor Ciarán Ó hÓgartaigh

Dedication

We dedicate this work to our beloved families who provided us the necessary support to accomplish this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

Foreword

This thesis was written as completion to the UCD MSc in Business Analytics degree. The master's programme focuses on developing quantitative techniques to explore management issues and support decision-making within a business context.

As this dissertation will illustrate, the subject of this thesis falls under the scope of optimisation models. Our aim in this work is to provide a vehicle routing optimisation model that has greenhouse emissions (CO_2) as its main objective function. The reduction of greenhouse gas emission is a modern problem. Therefore, governments in Europe and in the rest of the world have been joining forces to reduce its impacts in different fields and scales. This work goes in the same direction. We believe that optimised vehicle fleets and routes can be used to reduce greenhouse gas emissions and its impacts for the planet.

The subject was selected in co-operation with Xpreso. Xpreso is a start-up company based in Dublin, Ireland. Xpreso provides mobile applications for delivery companies and their customers. Xpreso's core business is providing optimal routes for delivery companies, and allowing final customers to select the most suitable time to receive their parcels.

This project is an academic initiative focused on practical applications of green vehicle routing problems. Our work will demonstrate the impact of selecting emissions as our main objective function and its impacts in real-world operations. We demonstrate the main results of our work in monetary, environmental and operational terms.

Preface

It always seems impossible until it is done.

— Nelson Mandela

Our research question was formulated alongside with our academic supervisor Dr. Paula Carroll and our business supervisor Fabiano Palonetto. The research of this project was a valuable asset to both of us, since it represented an opportunity to delve deeply into a relevant optimisation problem, discover its particularities and apply the results obtained in a real-world operation. We present now the parts of this dissertation thesis.

We begin in Chapter 1 with a short overview of the context of green logistics and its relationship with routing problems. We also provide a brief introduction of vehicle routing problems, its main assumptions and a formulation.

In Chapter 2 we provide the most important business contributions associated with this work. We present different aspects that can be associated with the solutions of our problem and its implications to delivery companies.

Chapter 3 introduces the operations research literature regarding routing problems. We present an extensive review of the different routing problems found in the literature as well as the most promising solutions currently applied to the different problems.

In Chapter 4, we first describe how we selected and cleaned the real-world data used in our method. We present and select the emission models used in our formulations. We also provide insights in how this data can be used in further improvements of a vehicle routing model using weight modifications.

Moreover, we describe the methods used to solve the travel salesman problem and the vehicle routing problems associated with fuel emissions. We provide a set of tests of these models to assure accuracy.

We explore in Chapter 5 the experimental results of the applications of our emission and solver model into a real-world dataset. We compare our results with the *ex-post* data and with optimised solutions minimising distance.

Chapter 6 examines the main results found by our methods. We present the main results and their implications for Xpreso and for delivery companies.

Finally, in Chapter 7, we present the conclusion of this work and indicate some possible avenues for further research following on from the ideas introduced in this thesis.

University College Dublin — Paulo Roberto de Oliveira da Costa
September 3, 2016 — Stefano Mauceri

Acknowledgements

We would like to thank our supervisor Dr. Paula Carroll first for suggesting this interesting topic of research and second for being extremely helpful and supportive throughout the course of this dissertation. Dr. Carroll was consistently available to steer our work in the right the direction whenever it was needed it.

We appreciate greatly the agreement of the company Xpreso to provide data and business expertise to our project. We are very grateful to Xpreso's team members for their help and hospitality during our stay at their facilities. We would like to thank specifically Mr. Fabiano Pallonetto for his support and background guidance.

Lastly, we would like to thank our mothers, fathers and sisters who have supported us during the course of this work.

Abstract

The Green Vehicle Routing Problem (G-VRP) is relatively new in the network optimisation literature. We propose a method to solve the G-VRP and the Green Travel Salesman Problem (G-TSP) using CO_2 emissions in a real-world database and compare with distance-based solutions. This works uses several Emission Models (EM) to estimate the amount of emission based on an upper bound according to the European Union regulations and models using vehicle's speed, weight and road slope. We solve both the G-TSP and G-VRP using a Genetic Algorithm(GA) based on the implementation of Prins (2001). In terms of solutions, we were able to reduce the amount of emissions for a single driver (G-TSP) in up to 50% when compared to the original tour taken by the driver and in up to 4% when compared to the optimal tour found for road distance. Further improvements can be achieved in emissions if a G-VRP is solved for a group of drivers. In our example, 15% more in emissions could be saved by using a G-VRP for a group of three drivers. Further improvements (post optimisation) in CO_2 emissions can be achieved given that the weight of the vehicle diminishes with time.

Chapter 1

Introduction

1.1 Background

Green logistics have been receiving attention from both general public and government over the past years. The main objective of green logistics is to meet economic and strategic goals at the "least cost" to the environment. In that sense, the word cost does not mean economic cost, but it relates to climate change, air pollution, soil degradation, waste disposal and their effects on the environment. Governments across the world have started programmes that incentive *green* actions in different fields, such as reducing greenhouse gas emissions, the usage of alternative fuels or electronic vehicles, reduced water consumption, etc.

Moreover, in the context of transportation and logistics these effects play an important role in both general public and industry points of view. According to the European Environment Agency the levels of greenhouse emissions coming from road transportations in 2014 reached the equivalent of 887 million tones of CO_2, which corresponds 21 percent of the total emissions in that year and an increase of 12 percent compared to 1990 levels. Thus, the entire society can and would benefit more from *green* actions taken over a long period time in the logistics field. In the social point of view, people can live better and longer in an environment free of pollution and where natural resources are used responsibly. Economically speaking, the adoption of sustainable strategies can benefit companies due to a more efficient usage of resources, the presence of government funding for sustainable actions and a possible increase in brand awareness.

Clearly a more environmentally cost effective vehicle routing solution will also benefit more sustainable transportation schemes. In this context, creating a more sustainable fleet of vehicles considering fuels, emissions and the environmental costs is important. Sustainability can be achieved if, for example, more efficient decisions are made in the planning phases of delivery operations. In the work of J.M. Bloemhof-Ruwaard and Wassenhove (1995) the close relationships between Operations Research (OR) and environmental impacts are provided. The author also notes the high correlation between OR environmental focused research and the change from corrective towards prevention policies over time.

The studies of routing problems in OR deal with the fundamental problem of delivering goods from plants to warehouses to final customers. The first attempt to optimally solve the routing problem was first developed by Dantzig and Ramser (1959). They developed an exact mathematical formulation to find the best route for a petrol delivery fleet. Their problem formulation can also be considered as a generalisation of the Travel Salesman Problem (TSP) where the aim is to minimise the route of a single vehicle. This type of routing problems were later defined as Vehicle Routing Problems (VRP), and in its traditional form the focus is to minimise the economical cost (distance) of serving all the customers in a determined network.

Usually, the consideration of different objectives and constraints in the VRP formulation increases the complexity and applicability of the model. Sustainable constraints and objectives are no different, and more often than not new constraints and objectives contribute to new application scenarios and models. In the context of Green Logistics the application of VRP using sustainable constraints and objectives is relatively new in the literature. The first works in this area date of 2007, making it an interesting and relatively new subject of study. The set of problems so called Green Vehicle Routing Problems (G-VRP) can be defined as problems that are concerned with reducing energy consumption, recycling refuse and managing waste disposal (Sbihi and Eglese, 2007). These problems try to blend the economic and environmental costs by creating optimal routes that meet both environmental and financial purposes. It is fair to assume that the reduction in total distance will also provide environmental benefits due to the reduction of fuel consumption and by consequence pollutants. On the other hand, other authors may argue that the lack of consideration of other effects such as traffic congestion, weather and unusual events might increase fuel consumption. In the VRP literature, the environmental effect is generally not measured or considered in the classical VRP approach (Sbihi and Eglese, 2007).

To solve these type of problems we must consider several factors. Both the TSP and VRP are NP-Hard combinatorial problems. This means that exact approaches with a large number of nodes might be impractical to solve to optimality in a reasonable amount of time. There are two main families of approaches for such problems, those are *exact* and *approximate algorithms*. The first aims to solve the problem to optimality and generally produces good results for relatively small problems with a reasonable set of constraints. According to Laporte (1992), exact algorithms for VRP can be classified into three categories: (1) *direct tree search algorithms*; (2) *dynamic programming*; (3) *integer linear programming.*

Approximate algorithms aim aim to find a solution in reasonable time which is guaranteed to be within a certain approximation factor of the optimal solution. Those methods usually find a satisfactory near-optimal solution for large-scale problems within a small computational time (Lin *et al.*, 2014). Different approximate algorithms, including classical heuristics and metaheuristics have been proposed to solve different variants of the VRP. Metaheuristics generally carry out more thorough search of the solution space, allowing infeasible and sometimes inferior moves. As a result, metaheuristcs are normally able to produce more accurate results for the VRP. The metaheuristics can be split into two different types. (1) Local Search and (2) Population Search (Lin *et al.*, 2014). Local Search methods explore the solution space by moving the current best known solution to another promising solution in its local neighbourhood, whereas population search based methods keep a pool of good candidates to produce promising offspring to update the set of current solutions and so are capable of searching a much wider neighbourhood and avoid being trapped at local optima.

Differently from the classical approach, our work aims to apply route optimisation using fuel emissions as our primary objective goal, therefore classified as a G-VRP instance. More specifically, this work aims to implement an heuristic that is capable of solving the VRP using CO_2 emissions as our primary goal. This dissertation thesis is completed in partnership with *Xpreso*. *Xpreso* is a start-up company based in Dublin, Ireland. Their business is based on providing an application software to parcel delivery companies. This service allows the customer to track, reroute or reschedule the parcel for a different time, giving the customer the power to decide the more suitable way to receive a parcel. *Xpreso* provides the real data instance used in this work and the access to third party VRP solvers. Our literature review has provided interesting insights on the most promising methods found in the literature to solve the G-VRP concerning greenhouse gas emissions.

However, two main tasks are crucial to the success application of the methodology presented in this dissertation, specially into real-world data. The most important factor is the data and model used to estimate the amount of fuel emissions. The right application of emissions estimations are paramount to the success of this project. The second most important factor relies on the algorithm used to efficiently solve the G-VRP. It is known that real-world applications exhibit lack of stability and specificness usually found in the academia, thus a method that can be used in real-world applications is to be preferred in this case. To guarantee the mitigation of risks and some type of quality measurement in this project, our work will both be applied to benchmark VRP datasets and real-world datasets. This provides a fair basis of comparison with other methods and provide a test framework for the methodology.

1.2 The Vehicle Rounting Problem (VRP)

The classical VRP can be defined as the problem to find a set of optimal routes beginning and ending at the depot, visiting all customers in the network only once so that the known demand of all nodes are fulfilled by a vehicle with fixed capacity.

1.2.1 VRP Formulation

The VRP is a combinatorial problem that can be defined as a complete graph $G(V, E)$.
$V = \{v_0, v_1, \ldots, v_n\}$ is a vertex set, where:

- Consider a depot to be located at v_0.
- Let $V = V \setminus \{v_0\}$ be used as the set of n customers.

$A = \{(v_i, v_j) | v_i, v_j \in V; i \neq j\}$ is an arc set.
C is a matrix of non-negative costs or distances c_{ij} between customers v_i and v_j.
d is a vector of the customer demands.
R_i is the route for vehicle i.
m is the number of vehicles (all identical). One route is assigned to each vehicle.
When $c_{ij} = c_{ji}$ for all $(v_i, v_j) \in A$ the problem is said to be symmetric.
With each vertex v_i in V is associated a quantity q_i of some goods to be

delivered by a vehicle. The VRP thus consists of determining a set of m vehicle routes of minimal total cost, starting and ending at a depot, such that every vertex in V is visited exactly once by one vehicle. (*http://neo.lcc.uma.es/vrp/vehicle-routing-problem/, Accessed on: June/2016*).

1.3 Academic Contributions

This project also brings several academic contributions to the VRP field. As our main academic contribution we aim to apply an heuristic method to the G-VRP with CO_2 emissions minimisation. The G-VRP is a relatively new problem in the literature and works using emissions minimisation are still scarce.

This work also aims to explore and apply a model for calculating fuel emissions levels. Although previously explored in the literature, this work aims to implement this relationship using real-world data. This allow us to produce a robust model that takes into account important variables for the delivery operation. By achieving this outcome, we are able to assess the route optimisation procedure using a different metric, compare and quantify the results.

Lastly, using real-world data is another novelty of this work, most literature in the G-VRP realm lack a more applicable approach. We aim to explore the close relationship with the industry and use this advantage to evaluate flaws of pure academic approaches in dealing with real data.

As a result, our main academic goal is to assess and quantify the effects of solving a Vehicle Routing Problem using fuel emissions as objective function. We propose to compare such results with the results of the classic VRP concerned with road distances in terms of distance travelled, fuel emissions and conclude on the results obtained.

Chapter 2

Business Contributions

2.1 Introduction

We begin this chapter by presenting the business and main contributions of the development of this project. This research is built in partnership with the start-up company *Xpreso*. *Xpreso* operates in the Irish market as a provider of routing services for parcel delivery companies. Briefly explaining, *Xpreso* receives from its customers a set of deliveries that have to be handled within a given day. An optimal route, in the meaning of minimising the travelled distance, is calculated for all the vehicles in a network. The optimal path is integrated in a software that will interactively present the optimal roads and stops to the vehicle driver. As additional optimisation strategy, customers are contacted with a text showing the Expected Time of Arrival (ETA) and asked for confirmation of the parcel delivery. The delivery can then be cancelled, rerouted to a different location or assigned a new delivery time.

As per its strategic plane *Xpreso* could strengthen its competitive advantage by expanding its competences in VRP algorithms through the inclusion of greenhouse gas saving properties. According to the generally high level of interest of both private and governmental organization in the reduction of atmospheric concentrations of greenhouse gases, it can be argued that if successful this might easily become one of *Xpreso's* core strategic competences.

As assessed previously, there is a large effort in applying and creating more *green* actions from both governments and companies. Benefits of such actions are economically, socially and environmentally proved. In the case or this work and its contributions to *Xpreso* we can identify three main groups of contributions: (1) *Environmental and Social*; (2) *Economical* and (3) *Operational*.

2.2 Environmental and Social

This work aims to provide a method for solving the G-VRP minimising fuel emissions. The direct impact of this method is to create a set of optimal routes that have a positive impact in reducing the amount of emissions for a parcel delivery fleet. We measure and compare the solutions of this work with the classic VRP concerned with minimising travelled distances. For the company purposes that will provide a benchmark of how effective a G-VRP solution is in reducing emissions, both in the long and short terms and its economical impacts.

Moreover, this work is aligned with public actions towards more sustainable actions. Governments have posed more attention to the environmental effects of their actions. For example, European governments have created several policies to incentive the application of environmentally friendly practices. Currently, the EU region has set new (signed in 2009) legislations to ensure that the region meets its climate and energy targets by the year 2020. The package has three main goals: (1) 20 percent cut in greenhouse gas emissions; (2) 20 percent of EU energy from renewable energies and (3) 20 percent improvement in energy efficiency.

In Ireland the Sustainable Energy Authority of Ireland (SEAI) has created several goals for sustainability and fuel usage in the island. The SEAI also awards grants for companies that comply with more sustainable actions. Those goals involve for example, monetary grants, electric vehicle schemes and investing in more sustainable homes.

2.3 Economical

Governments invest in companies that adopt more sustainable actions. Those types of investment come in form of tax allowance, international certifications or additional funding. Also, companies that adopt such actions can benefit from increased brand awareness. As a result, a successful implementation of the G-VRP can lead to more economical factors than those originally considered in the methodology.

However, the most direct economical factors are those considered in the context of the cost for courier companies. Those companies main concern is related to minimising the costs of delivering goods to their customers. In the bottom level this means minimising the travel time, working hours and fuel consumption. There is a direct relationship between fuel usage, travel time and speed. For example in the work of Hickman *et al.* (1999) a mathematical model for fuel emissions is derived using its direct relationship with driving velocity. Therefore, reducing fuel emissions can have an impact in reducing travel time according to the proposed formulation. In *Xpreso's* point of view this means providing a more cost effective service in terms of not only fuel emissions, but in terms of travel time and fleet efficiency if a multi-objective function is considered.

2.4 Operational

Xpreso's current practices aim to solve the classical VRP and TSP (depending on the customer's needs) minimising the total distance of serving all customers. In our work we will evaluate different approaches currently used in terms of methodology and application. We aim to provide an evaluation of such approaches and their different solutions regarding solution quality and speed. At the end, we aim to construct a possible different solution to the problem using an heuristic approach. Considering fuel emissions, and therefore, travel times as our primary object function we can improve the solutions currently used regarding efficiency. By approaching the problem using a broader objective, better results can be found in terms of operational purposes (travel time, working hours, emissions, etc.) for the company's clients. Therefore, this potential business advantage could be used to provide more efficient solutions in terms of economical costs to delivery companies while complying with the the fuel emissions reductions.

Chapter 3

Literature Review

3.1 Introduction

In this literature review we present a broad overview of the VRP and G-VRP problems. The main methods for solving the VRP are presented with emphasis on VRP instances related to the G-VRP and on approximate algorithms and efficient methods to solve the problem. Lastly, an overview of the benchmark datasets is also provided.

3.2 The Vehicle Routing Problem (VRP)

Since its introduction, the VRP and its variations have been studied extensively making this field one of the most widely explored within the OR community; for instance, a search on *Google Scholar* of the key "Vehicle Routing Problem" will produce approximately 384,000 results in June, 2016. While much has been done about the VRP, and some notable innovations have been introduced throughout the time, new challenges are still arising. At the same time, the set of techniques employed for modelling and solving the VRP is constantly improving. As a matter of fact, the VRP is the core of every distribution system design and management. Toth and Vigo (2014) have mentioned in their excellent work the non-negligible economic impact of a computerised solution of the VRP, both at the planning and operational levels.

Thanks to the effort of several researches we are now able to solve large-scale problems in a reasonable computational time obtaining more accurate solutions. Accuracy in this context means solutions that are closer to optimal values when applied to benchmark datasets. It is worth mentioning that real-world datasets are not as well behaved as benchmark datasets and solutions previously considered good for academic datasets may not be applicable for real-world data.

The VRP was first introduced by Dantzig and Ramser (1959) as the "Truck Dispatching Problem", the present name appeared for the first time in 1977, Golden *et al.* (1977). The problem can be defined as the problem to find a set of optimal routes beginning and ending at the depot, visiting all customers in the network only once so that the known demand of all nodes are fulfilled by a vehicle with fixed capacity. The problem was considered as a generalisation of the TSP presented by Flood (1956). Indeed the TSP is a special case of the VRP where only one vehicle of infinite capacity is available. In its most typical formulation the VRP is known as the Capacitated VRP or the Classical VRP, (CVRP). Dantzig and Ramser (1959) provided the first mathematical formulation of the problem, they also suggested an heuristic approach to find a near optimal solutions when the size of the problem would make an exact algorithm inapplicable.

In 1964 Clarke and Wright (1964) published a so called saving algorithm for the solution of CVRP which established a brilliant innovation. The basic idea of the algorithm is to join two existing routes in a new feasible one and retain this new route if and only if its cost is lower than the sum of the costs associated with the two initial routes. According to the properties of the cost matrix, so far defined as a distance matrix, if the distance between locations does not depend on the direction, we are dealing with a *Symmetric* VRP, otherwise we are in the *Asymmetric* case. The main consequence is that while in the first scenario we are dealing with a complete graph, in the second instance we are dealing with a complete digraph which increases significantly the number of arcs, Toth and Vigo (2001).

Again considering the cost matrix, we might argue that the *Euclidean* distance is not a realistic estimation of travelling costs due to the several reasons that may affect the viability within the considered network, for instance: rush hours, weather conditions, random events, etc. These considerations are included in the Time-dependent VRP (TDVRP) which was first introduced by Cooke and Halsey (1966). The authors created a function to estimate the travel time between two locations as a quantity depending on the starting time at the location being left. However, this formulation represents a modification of the Shortest Path Problem rather than a TDVRP.

Malandraki and Daskin (1992) for the first time, took into account variable traffic conditions according to the time of the day. They used a Mixed Integer Programming (MIP) formulation of the TDVRP and a nearest neighbour heuristic for solving a time-dependent formulation of the TSP and the VRP. Assuming the time of the day as the factor with most influence on the travel time, they modelled the travel time between two nodes as a known step function of the time of the day at the origin node. The authors underline that the TDVRP represents a more accurate approximation of an urban environment compared to its distance based counterpart. Furthermore, it is important to notice that this formulation considered an asymmetric problem, for that reason some heuristics like the k-OPT cannot be easily extended to solve this variant.

Subsequently, Ichoua *et al.* (2003), considered soft time windows, that is, if the arrival time at a node is later than the end of the time window, the cost function (for instance, the total travel time) will be penalised by some amount. The authors used a tabu search heuristic, in the context of a dynamic environment, where not all service requests are known in advance; the capacity constraints for the trucks are not considered. They concluded that time-dependent models provide substantial improvements over a model based on fixed travel times.

In another research article about the TDVRP, Donati *et al.* (2008), showed that when dealing with time constraints, like hard delivery time windows for customers, the known solutions for the classic case become infeasible, whereas if no hard time constraints are present, the classic solutions become suboptimal. When comparing TDVRP and VRP solutions one should bear in mind that comparing the quality of routes is not straightforward. The solutions of the TDVRP can have larger distances than those proposed by the classical VRP. But they can have a lower travel time when compared to the classical approach. Defining an effective way of comparing those solutions involves considering the same type of metric to different problems.

As mentioned before, in the service industry, customer's deliveries are usually scheduled within time intervals which are called time windows. Time windows can be hard or soft. In case of hard time windows, a vehicle that arrives too early at a customer must wait until the time service starts. In general, waiting before the start of a time window incurs no cost; late arrival is not allowed. In the case of soft time windows, every time window can be violated at the price of a penalty cost. Solomon (1987) considered a version of the VRP with Time Windows constraints (VRPTW). The authors introduced a variety of heuristic methods, applications and solutions which are still used as benchmark instances. Recently, Baldacci *et al.* (2012) produced a survey about the exact algorithms for solving the VRPTW stating that the improvement achieved allows to solve problem instances with more than 100 nodes. Vidal *et al.* (2013) presented a new class of hybrid genetic algorithms for solving a large range of VRPTW variants stating that their innovations have significantly improved the state of the art in terms of solution accuracy and computational efficiency.

A further relevant variation of the VRP is known as the Pick and Delivery Problem (PDP) in which goods or passengers have to be transferred from different origins to different destinations. This problem is extensively studied in the literature and two main versions can be found in Parragh *et al.* (2008). In the first one we deal with the transportation of goods from the depot to target node and from backhaul nodes to the depot; this problem is often referred to as Vehicle Routing Problem with Backhauls (VRPB). In the second one, goods are transported between pickup and delivery locations; this is cosidered the Pickup and Delivery Vehicle Routing Problem (PDVRP).

Extending our considerations about the VRP we cannot overlook the concept of integrated logistics systems. This concept aims to increase the overall efficiency of a distribution system recognising the interdependence among the location of depots, the allocation of suppliers and customers to the depots, and the vehicle routing strategies consequently applied Min *et al.* (1998). A recent excellent review of this problem is provided by Nagy and Salhi (2007).

So far we have considered a deterministic operational environment where all the information is known in advance, however in real world applications we often have to deal with uncertainty. It is not unusual that elements of the problem like the travel time, the customer demand, and the set of customers are random (Gendreau *et al.*, 1995) and (Laporte *et al.*, 1992). This class of problems is known as the Stochastic VRP (SVRP) and is modelled using tools from the probability theory to deal with uncertainty. Because of uncertainty, it may not be possible to follow a planned route, so when a constraint is violated the vehicle has to come back to the depot and a penalty is added to the cost function. These formulations are particularly useful to plan and control strategic operations.

Sometimes it is required that while information evolves decisions must be changed accordingly in a mutable environment. The goal is not only to anticipate future random events as in the SVRP but also to react to new happenings.

This approach is known as Dynamic VRP (DVRP), and its formulation dates back to Psaraftis (1980), it requires not only a specific formulation approach, but advanced technological support and real-time communication between the vehicle and the dispatcher. The variability can affect one or more problem parameters like the customer locations, the demands, and the travel times. Also the execution of the routing plan can change over the time to take into account service cancellations and vehicle availabilities. Haghani and Jung (2005) proposed a genetic algorithm for solving the DVRP with soft time windows, multiple vehicles with different capacities, real-time service requests, real-time variations in travel times between demand nodes, and pickup and deliveries.

A further version of the VRP is known as the Open VRP (OVRP) and is of particular importance when the deliveries are entrusted to a third part company, a common practice in the distribution and service industry. The main idea is that each route is a *Hamiltonian Path* rather than a *Hamiltonian Circuit*, that means the vehicles are not required to return to the depot. The problem was first introduced by Sariklis and Powell (2000). Li *et al.* (2007) provide a review of the algorithms employed for solving this problem and also propose their version of the record-to-record travel algorithm.

3.3 The Green Vehicle Routing Problem (G-VRP)

The environmental considerations are an essential strategic factor of success for modern companies. As described by Lyon and Maxwell (2008), in the last decade environmental concerns have gained a leading role in effective corporate social responsibility policies. The authors emphasise the importance for firms to promote sustainable development ensuring the necessary balance between environmental, economical and social aspects. These initiatives are welcomed by employees, consumers, investors, regulation agencies, and governments. Transport services are a vital component of our industrial system but at the same time have negative environmental impacts including greenhouse gas emissions, pollution, noise and accidents (Maibach *et al.*, 2007).

The discussed considerations explain the interest of researchers for a recent formulation of the VRP named as Green VRP (G-VRP), which aims to minimise the polluting effects of transportation on the environment including fuel consumption and emissions. A valuable review of the VRP, since its first formulation to the recent attention to environmental sensitivy, that has led to the formulation of the G-VRP was produced by Lin *et al.* (2014). Energy consumption and pollution considerations are very rare in the literature before the 2000s. Cairns (1999) considered the environmental impact of grocery home delivery simply converting distance into emissions without attention for other parameters. Woensel *et al.* (2001) demonstrated the impact of traffic flow in emissions explaining how the assumption of constant speed leads to unrealistic emissions estimations and consequently to suboptimal routes. Kara *et al.* (2007) introduced a new cost function based on distance and load to minimise the energy consumption in the CVRP.

Substantially, what is important in the G-VRP is the formulation of the problem rather than the solution approach that usually requires only little modifications of an existing method. Palmer (2007) discussed extensively the VRP with time, distance and CO_2 underlying how the estimated total emissions vary for different minimisation and traffic congestion criteria. Results confirmed that minimising CO_2 emissions instead of time can allow about 5 percent more savings in CO_2 emission. The amount of CO_2 emitted by a vehicle is directly proportional to fuel consumption. The fuel consumption is influenced by numerous parameters including the travelled distance, average speed, average acceleration, vehicle load, engine type and size, road gradient, temperature and several other aspects (Boulter *et al.*, 2007).

In general, in the literature there are two ways to estimate fuel consumption for vehicles: on-road measurements models, which are based on real-time collection of emissions data from a running vehicle, and emission models, which estimate fuel consumption based on a variety of vehicle, environment, and traffic related considerations (Baškovič and Knez, 2011). While the first class of models can be point out that it is extremely difficult to accurately measure emissions on a continuous basis; the second one is increasing its popularity due the easy of use and data availability Wang and McGlinchy (2009). Esteves-Booth *et al.* (2002) provide us with a taxonomy of the emission models:

(1) Emission factor models: The estimation of the emissions is expressed by the use of an emissions factor usually expressed per unit of distance, related to one category of vehicles and a specific driving model (i.e. urban, rural or motorway). This class of model is particularly useful for large scale studies (e.g. national emissions estimation). Practically, emission factors are derived from the mean value of repeated measurements over a specific driving cycle. This approach is not very accurate on micro scale, but becomes effective on average.

(2) Average Speed Models: The estimation of the emissions is based on speed-related emission functions, realised sampling the emission rates over a variety of tours at different speeds. These models are often used for road networks while they do not include enough parameters for a micro scale use. A simple and extensively documented average speed model for real road networks is named COPERT and is described by Ntziachristos *et al.* (2000). A possible alternative found its roots in the MEET report published by the European Commission and implemented by Hickman *et al.* (1999).

(3) Modal Emission Models: These models are more complex and detailed than the previous one and allow an effective use at a micro scale. The emission rate is expressed of different levels of speed and driving modes like: acceleration, deceleration, steady-speed cruise and idle. The emissions are presented as a function of the engine demand and other physical parameters requiring a large amount of data. Demir (2012) and Demir *et al.* (2011) have conducted comparative studies of several emissions models.

3.4 Exact Algorithms for the VRP

VRP is known to be an NP-hard combinatorial problem. For that reason optimal or near-optimal solutions are obtained using either exact or approximate algorithms. Exact algorithms are suitable for relatively small problems (normally no more than 50-200 customers with a small set of constraints). Usually, the application of exact approaches were developed from successful algorithms applied to the TSP problem (Toth and Vigo, 2014). In the literature it is possible to find several papers that analysed exact algorithms approaches of the VRP, those include the work of (Laporte, 1992) and (Toth and Vigo, 2002). There are mainly three exact method categories used to solve the VRP problem, those can be divided as (1) *Direct Tree Search Methods*; (2) *Dynamic Programming* and (3) *Integer Linear Programming* (Laporte, 1992).

3.4.1 Direct Tree Search Methods

To solve the VRP using the Branch-and-Bound (most famous Direct Tree Search Method) authors incorporated basic combinatorial relaxations based either on the Assignment Problem (AP) or the Shortest Spanning Tree (SST). After these first developments the methods were based on Lagrangian relaxations and additive approaches, which brought the Branch-and-Bound algorithm to its peak capability prior to the introduction of cutting planes approach. The combinatorial relaxation first used in the literature was an extension of Little *et al.* (1963) for the TSP. It is constructed ignoring the sub-tour elimination constraints, the resulting problem is a transportation problem, solved for a minimum cost collections of circuits. For example, this type of relaxation was used by Christofides and Eilon (1969).

Other relaxations for the VRP used solutions of Shortest Path and Spanning Tree Problems. These relaxations were obtained by weakening the General Sub-tour Elimination Constraints (GSECs) to impose the connectivity of the solution but ignoring the degree requirements. Methods developed by Christofides *et al.* (1981a) were based on a method for symmetrical VRPs based on the k-degree center tree relaxation where constraints are relaxed in a Lagrangian fashion, based in the works of Fisher (1994) and Christofides and Eilon (1969) with k-trees formulations. However, the first combinatorial relaxations attempts for solving the VRP are of poor quality and allow for only a reduced number of instances of the problem. As a result, Fisher (1994) and Miller (1995) proposed to use the dual of the problem in order to strengthen the relaxations, by including both degree constraints and GSECs in the objective function.

3.4.2 Dynamic Programming

The structure of dynamic programming was first used for the VRP by Eilon *et al.* (1971). The method relies on the recursion structure of the problem, minimising the costs at each step of the algorithm. Efficient use of dynamic programming requires a substantial reduction of the number of states by means of a relaxation procedure, feasibility or dominance criteria. State-space relaxation provides another efficient way of reducing the number of states. The method was introduced by Christofides *et al.* (1981b). More recent work by Ou and Sun (2010) shows that this approach can be used for more complex problems, given the development of computer power throughout the years.

3.4.3 Integer Linear Programming

Set partitioning (SP) formulations were introduced by Balinski and Quandt (1964). Their basic formulation accounted for binary variables a_{ij} which assume value 1 if the route (i, j) is feasible. "The SP Formulation cannot be used directly to solve non-trivial CVRP instances because of the large number of potential routes"(Baldacci *et al.*, 2012). However, "if the size of the problem is relatively small the linear relaxation of this formulation often provides an integer solution" (Toregas and ReVelle, 1972). If the solution x^* does not provide an integer solution then a cutting plane can be introduced in the formulation. More recent methods based on set partitioning algorithms have been developed. The work of Baldacci *et al.* (2008) proposed a different SP modification accounting for strengthened capacity inequalities and clique inequalities. Also, they proposed a new algorithm to solve the problem to optimality.

Fisher and Jaikumar (1981) have developed a three-index vehicle flow formulation for VRPs with capacity restrictions, time windows and no stopping times. In three-index formulations, variables x_{ijk} indicate whether (i, j) is traversed by vehicle k or not. Whereas in two-index formulations, variables x_{ij} do not specify which vehicle is used on (i, j) (Laporte, 1992). The authors developed an algorithm for this formulation that guarantees a solution is found in a finite number of steps.

Exact approaches have been improving due to new techniques using branch-and-cut algorithms and have taken advantage of the evolution of computer capabilities. Nonetheless, real-world problems in general are not tractable using exact algorithms. More often than not, those problems represent larger instances with more constraints to be considered. Even though the exact algorithms have improved, they are still not suitable for most real-world problems due to the large amount of time needed to run exact models in large problems and the necessity of different formulations to deal with different sets of constraints.

3.5 Approximate Algorithms for the VRP

The necessity of near-optimal heuristics that are efficient and flexible in practice were the subject of research of many authors in the literature. A variety of *constructive* and *improvement* heuristics have been developed to tackle the VRP problem and its variants. More recently, the advent of *metaheuristics* made possible to find sub-optimal solutions that vary in less than one percent of their best known values (Toth and Vigo, 2014).

This section aims to represent some basic ideas about the possible heuristic solutions found in the literature. Our aim here is not to explain and analyse every single heuristic paper present in the VRP literature. More thorough researches in this sense have been made by Toth and Vigo (2014) and Laporte *et al.* (2014).

Our goal is then to present the general ideas of constructive and improvement heuristics and of metaheuristics. Also, we aim to present the idea of the current most promising algorithms applied to some instances of the VRP problem with the objective of selecting the best approaches to be related with this work.

3.5.1 Constructive Heuristics

Constructive heuristics are normally employed to provide a starting point to an improvement heuristic. They usually start with an empty solution (or infeasible) and improve until a complete initial solution is found. Most metaheuristics can also be initialised using constructive heuristics, however, as those methodologies carry a much thorough search in the solution space, metaheuristics are able to find feasible solutions even when starting from infeasible solutions (Toth and Vigo, 2014).

Three constructive heuristics are still of particular interest, the nearest-neighbour heuristic, the Clark and Wright Savings heuristic and Petal Algorithms. The latter relates to set partitioning problems and it was first proposed by Balinski and Quandt (1964) and posteriorly improved by other authors. Petal algorithms are particularly well suited for problems containing constraints such as time windows, other than capacity and route duration constraints. Column generation then becomes a solution methodology of choice, especially for tightly constrained instances (Toth and Vigo, 2014).

3.5.2 Improvement Heuristics

Classical improvement heuristics perform route moves. They improve the best current known solution by making changes on its structure. One of the most effective improvement heuristics for the TSP that performs intra-route changes is the Lin-Kernighan (LK) heuristic. This heuristic is a generalisation of the $2-OPT$ and $3-OPT$ heuristics. The LK heuristic performs λ-OPT changes at each iteration, λ edges are removed and replaced by other edges that minimise the overall cost. The value of λ is dynamically chosen within the search procedure that may result in a shorter tour (Lin and Kernighan, 1973).

The work of Helsgaun (2000) proposed an improvement of the original heuristic, where only exchanges with a potential to improve the algorithm are performed and provisional gains must always be positive. Simplicity and speed are the two main concerns of this implementation. Results have shown promising results regarding both speed and accuracy. The size of the tested problems using the LK heuristic ranges from a few hundreds of nodes up to 85,000. The Lin-Kernighan-Helsgaun (LKH) open source package has reached the current best known solution to the World TSP with nearly 2 million nodes. David Soler has made available online 126 asymmetric instances of the LKH with known optima (Soler, 2016).

Inter-route changes are also important to guarantee good to improvement heuristics results. These include relocating customers in a route and placing in a different route, swapping different customers between different routes or removing edges between different routes and connecting them elsewhere Toth and Vigo (2014). For large problems evaluating all the possible changes is impractical, thus some pruning techniques have been developed (Johnson and McGeoch, 1997) to restrict the moves between distant customers. Inter-route improvements can also be seen as operators within destroy and repair schemes. In the Adaptive Large Neighbourhood Search (ALNS) moves are randomly selected using a roulette wheel mechanism (Toth and Vigo, 2014). Prins (2001) developed an interesting route split procedure that first builds a big TSP tour and select the best combination of trips regarding some criteria.

In general, *improving heuristics* represent interesting methodologies when previous sub-optimal local optima are available. They provide a fast and accurate set of algorithms for the search of improvements in the final solution.

3.5.3 Metaheuristics

Metaheuristics applied for the VRP can be classified into local-search methods and population search methods (Lin *et al.*, 2014). Local search methods explore the solution space by iteratively moving the current solution to another promising solution in its neighbourhood. These methods start from an initial solution x_1 and move at each iteration from a solution t to a solution $t+1$ in the neighbourhood of t, $N(x_t)$ (Toth and Vigo, 2014). Local-search methods do not necessarily improve the cost of the objective function at each iteration, so extra care must be taken to avoid cycling. The main local search algorithms for the VRP include (1) Simulated Annealing (SA); (2) Deterministic Annealing (DA); (3) Tabu Search (TS); (4) Variable Neighbourhood Search; (5) Large Neighbourhood Search;

Population search methods were derived to avoid falling into local optima and possibly cycling in a set of solutions. These methods build their theory using natural concepts e.g. evolutionary algorithms. However, usually all known good implementation of such methods for the VRP problem are also based on local search algorithms (Toth and Vigo, 2014). For that reason, those algorithms are considered hybrid. The main methodologies used in the VRP literature are: (1) Ant Colony Optimisation (ACO) and (2) Genetic Algorithm (GA).

It is extremely hard to determine which type of heuristic or metaheuristic is the “best” for a designated VRP instance. The amount of literature and the combination of different type of heuristics (hybridization) in the subject makes it almost impossible to define for sure an implementation that can outperform all the others. What is usually more common is that each method uses a combination of heuristics to identify more efficient ways of finding the optimum for a set of VRP problems. Nevertheless, one can possibly try to evaluate the quality of a given heuristic by imposing them to the same conditions and testing for their results. Usually, this type of evaluation is performed using some type of benchmark datasets. Those benchmark data allow researchers to define the "best" approaches and keep improving on previously assessed "good" methods.

To assess the quality of some metaheuristics we focus our attention into the work of Toth and Vigo (2014). Their heuristics comparison used two benchmark set of datasets. Their comparison of metaheuristics is mainly based on the quality of the solution (compared to known optima) and speed. They created rough estimates to impose the same conditions to all the heuristics and used the best results for each of the methods to evaluate their quality.

However, even analysing the results can be very tricky. For instance, for the set of datasets with smaller instances the papers of (Nagata and Bräysy, 2009); (Subramanian *et al.*, 2013); (Groër *et al.*, 2011) (large running time) and (Vidal *et al.*, 2012) were able match the exact solutions on the tests performed, but other metaheuristics tested also presented very close results (up to 0.64 percent of gap). These works have different combinations of heuristics, for example, the work of Nagata and Bräysy (2009) is based on the edge assembly (EAX) crossover with local search combination; (Subramanian *et al.*, 2013) developed another hybrid algorithm that combines local search, set partitioning and variable neighbourhood search; (Groër *et al.*, 2011) based their method on a local search improvement procedure combined with integer programming.

For large dataset instances, the works of (Groër *et al.*, 2011);(Vidal *et al.*, 2012) (population evolutionary search and neighbourhood-based method), (Nagata and Bräysy, 2009) and (Zachariadis and Kiranoudis, 2010) (local search based method) were able to find the best solutions with a variety of running times. There is an interesting result in the G-VRP field provided by Erdougan and Miller-Hooks (2012), but in general G-VRP solutions are still scarce in the literature.

As presented, determining the best solution approach for a the VRP and G-VRP is not an easy task. Usually, the considered best methods are based in some type of neighbourhood search combined with other heuristics/exact methods. Our research has a clear understanding that determining one best approach for all VRP instances is not possible and for the sake of this work, our main line of research will be based on improvement heuristics and neighbourhood search (combined with population search) that can achieve the best results, since their vast applicability in the literature and simplicity.

3.6 Benchmark Datasets for the VRP

In the literature is possible to find a good amount of benchmark solutions which differ in the size of the instance as well as in the type of problem provided. Some of the commonly employed benchmark instances Solomon (1987) who applies his heuristic approaches to Solomon's test set and to four real world problems of 50, 100, 249 and 417 nodes.

An appropriate TSP benchmark set is provided by Helsgaun (2000) who solved several instances of the TSP using an optimised implementation of Lin and Kernighan (1973). The size of the tested problems ranges from a few hundreds of nodes up to 85,000. Christofides *et al.* (1981a) and Christofides and Eilon (1969) also provide widely used instances for the VRP.

The usage of such databases is important to guarantee a fair comparison between methods and to keep track of best current solutions for such problems. In this work we intend to use such databases to provide this type of fairness when evaluating the most suitable approaches in terms of robustness, speed and accuracy of the final solution.

Chapter 4

Methodology

4.1 Introduction

The objective of our research is solving an instance of the G-VRP. As previously mentioned, the goal is to define a route for delivering a set of goods to a set of known destinations with the primary objective of minimising greenhouse gas emissions.

Modelling the emissions is one of the most challenging part of this study due to the lack of empirical data at our disposal. Starting from the dataset provided by *Xpreso* and the geospatial information retrieved through *Google Maps* we have modelled variables like the distance, the average speed of the journey and the road gradient between each pair of nodes in a considered road network. The mentioned variables have been used as input for our best emissions model allowing us to look at the VRP from the *Green* point of view. After the definition of the emissions metric to assess solutions, we have developed a G-VRP heuristic genetic algorithm which wisely combines solid optimisation strategies in a framework designed to guarantee flexibility, accuracy and efficiency in terms of computational time.

4.2 Input Data

In this paragraph we will illustrate the input data exploited to model the emissions matrix needed for our G-VRP.

Xpreso is an innovation and technology driven start-up company developing software solutions for small and large parcels delivery companies. The business objective is to provide the partners with an all-in-one Android based application that allows:

- Drivers' route optimisation
- Real time tracking
- Time window notifications to customers
- Real time routes adjustments
- Data gathering and analytics

This study focuses on the "route optimisation" solutions. Our work starts with a dataset containing the typical raw data used at *Xpreso*: a set of GPS locations stored as latitude - longitude pairs associated with a driver code. As it will be clarified ahead, the company optimises a single route for each driver given a set of destinations already associated with him/her. This is the equivalent of solving a number of TSP and not a single VRP. This decision can lead to suboptimal solutions for all stops.

Typically, the carrier company allocates the packages to be delivered in a day to its drivers. The companies load the delivery vans in the morning before all deliveries start. As per business practice, each driver is associated with a portion of the city area, so a driver is assigned if a delivery falls within his area. There are two main constraints in the operation: the volume constraint, which requires the delivery companies to have enough space on the van to store all the packages in a delivery route and the time constraint. In fact, a typical work shift lasts 8 hours and no more, from 8 a.m. to 5 p.m. including one hour for lunch break.

According to these constraints and some other technical considerations, the delivery company produces a dataset where for each row we find the full address for a given delivery, the post code associated with it and a driver code to identify the person in charge for the transport. It is easy to imagine that in the real world, things are slightly more complicated, and sometimes the dataset does not reflect the current distribution of the packages within the vehicles or some post codes are missing. These limitations are overcome through data validation and data cleaning processes which are not presented in this paper.

Given the delivery company data, *Xpreso* retrieves the latitude and longitude associated with each address, finds the optimal route for a single driver and associates with each delivery an expected delivery time. The optimal route is shown on map and displayed on a tablet supplied to each driver. According to the expected delivery time a text is sent to the customer which could reject the delivery and ask to move it to a different time window or a different day. From the customer point of view, this can sensibly increase the quality of the service experience but at the same time this increments the organisational complexity for either *Xpreso* or the delivery company.

In this study we are dealing with *ex-post* data, the data hold by *Xpreso* after each workday is consolidated for statistical purposes. A typical row of the dataset at our disposal is represented in Table 4.1.

Latitude	Longitude	Parcel Weight	Delivery Time	Driver Code
51.503	-2.685	1000	9.345	Driver_1

Table 4.1: Raw data provided by Xpreso

In the data presented latitude and longitude points out the geospatial position of a customer; the parcel weight is expressed in grams and the delivery time is the clock time of the delivery. It must be mentioned that this data allow us not only to model the emissions and trace the optimal route but also to compare our results with the actual route taken by the driver in the real business practice. Finally, the driver code identifies the person carrying that parcel.

Xpreso has provided us with a dataset of almost two months of daily data where we deal with a variable number of drivers (between 30 and 40 per day) and around 3000 deliveries. Each driver has been assigned to a variable number of deliveries, usually between 20 and 160. As far we are concerned we have counted multiple deliveries happening at the same address as a single one. The reason of this choice is straightforward: in solving a TSP or a VRP, a road address is a node of the road network, considerations like the number of packages we plan to delivery at that point or their weight are not immediately relevant in this study. However, we have to mention that the number of deliveries influences the service time, usually estimated in 3 minutes per stop.

All the locations are found in the urban Bristol area. The city and county area are 110 Km^2 large with a radius of 6 Km. The territory is considered to be flat with some smooth hills in the surroundings. The population is around 428,000 people. Also, the approximate distance between the depot and the city centre where most of the deliveries are intended to be dispatched is 12.8 Km.

Figure 4.1: Bristol City Area Map

4.3 Building Up The Emissions Matrix

4.3.1 Calling the API

Vehicles' emissions are dependent on a variety of parameters which span from the travelled distance, the speed, the acceleration, the road gradient, the average temperature of the period, the temperature of the engine, the type of engine, the vehicle's volume, weight and load just to mention the most common ones. As a thorough model accounting for all possible variables is beyond the scope of this study, we have based our assumptions on models using the available data: distance, speed and road gradient.

Using a script written in *Python Programming Language* we have exploited the *Google Maps API* named *Distance Matrix API* to retrieve the distance and the travel time between each pair of nodes of our road network. The method consists in writing an *ad hoc* internet *URL* where some parameters like the latitude and longitude of two locations can be set. Once arranged, a proper *URL* can be used as input of a query through an *HTTP* interface; the request will be answered by *Google Maps* with some lines of text where the information is embedded in *JSON Programming Language*.

In order to use this service *Google Maps* asks you to create your *API key*, a sort of personal ID to trace your requests to the *Google Server*. Private users can have a key for free and this allows up to 2500 queries per day. However, an average sized distance matrix with 60 nodes has 3600 entries; *Xpreso* has allowed us to use one of their professional key which consents up to 100,000 requests per day.

4.3.2 Distance and Speed

The distance between a start and an end point is based on *Google Maps* "recommended route" and expressed in meters and kilometres. The travel time parameter is available only when the “travel_mode” is set to “driving” and the “departure_time” is set to “now”. Furthermore, the “traffic_model” has been set to “best_guess”.

In this way the estimated travel time should be the best estimate given what is known about both historical traffic conditions and live information. We decided to launch the queries always at 10 a.m., assuming this time as representative of the average traffic conditions of a workday: not busy as rush hours but not clear as night hours.

The time is expressed in minutes and hours. For every variable appropriate unit conversions are required. For instance, defining the speed in kilometres per hours requires us to express all the distances in kilometres and all the travel times in hours.

To call the API we need to set up the origin’s latitude and longitude and the destination’s latitude and longitude within an *ad hoc* built *URL*. For instance, using the coordinates from *Bath* to *Bristol* in the *United Kingdom* the output generated by *Google Maps* can be checked in Figure 4.2.


```
{
   "destination_addresses" : [ "12 Whitby Rd, Bristol BS4 3QH, Regno Unito" ],
   "origin_addresses" : [ "143 Calton Rd, Bath BA2 4PP, Regno Unito" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "18,2 km",
                  "value" : 18166
               },
               "duration" : {
                  "text" : "29 min",
                  "value" : 1749
               },
               "duration_in_traffic" : {
                  "text" : "29 min",
                  "value" : 1748
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```


Figure 4.2: Google Output Embedded In JSON Code

As mentioned above, the output of the *Google* query is expressed as *JSON* code that can be easily stored in a dictionary data structure using an appropriate *Python* library, e.g. "json". When this is done we can retrieve the information we need from the data structure and store them in appropriate matrices.

For example in Figure 4.2, we have learned that the distance between *Bath* and *Bristol*, driving a car, in normal traffic condition is equal to 18.2 kilometres and can be travelled in 29 minutes. Furthermore, after appropriate unit conversions, we divide the travelled distance by the travel time finding the average speed of the journey, in this example around 37.9 Km/h.

4.3.3 Road Gradient

Additionally, the average road gradient (that measures the inclination of a start and an end point) expressed as a percentage can be calculated exploiting the origin and destination postcodes. *Xpreso* has provided us with a dataset where all the postcodes of the *Bristol* area are listed along with the corresponding altitude at that point. During our tests we have found that some postcodes are missing from the given dataset. This can happen because some postcodes are no longer active and they have been substituted by new ones.

However, looking at a postcode e.g. "BS4 3QH" we can notice that it is made up by two components: the first "BS4" refers to a specific portion of the city area, while the second "3QH" points to a specific location within it. We have decided that in the unlikely event of a missing postcode, the altitude for that point is set as the average altitude of the corresponding city area. For example, given two points, A with altitude 18 meters and B at a distance of 2000 meters with an altitude of 6 meters, the road gradient is equal to 0.3%. Formally this is expressed in equation 4.3.1.

$$Gradient\,(\%) = \frac{Rise}{Run} * 100 \tag{4.3.1}$$

Where *Run* is the horizontal distance and *Rise* is the vertical distance between two points (linearity is assumed). The equation 4.3.1 is performed iteratively through all the pairs of points of our road network which includes the depot coordinates as well.

The proposed approximation of the road gradient is sensible to the distance. In fact, when the distance increases to values greater than 1 km we lose accuracy in the estimation of the slope. However, as will be graphically shown ahead, the most part of the nodes of the road network are close to each other; a small number of densely populated clusters are linked by longer arcs.

Another issue identified about the road gradient is the appearance of anomalous values of the slope, too large or too small. This happens when for a normal value of the rise is divided by a run that is too small. The average approximation to deal with missing postcodes is also a cause of this problem. Considering *Bristol* as a quite flat area and analysing the slope data for all other arcs we have decided to limit the slope within the range from -6% to +6%. Every value outside of this range is rounded to its closest bound.

In order to significantly reduce the number of *Google* queries and obtain a set of symmetric matrices we compile only the upper triangle of a given matrix and simply copy the values to the lower one. First, imagine that a driver has to visit 59 locations, if we add the depot to this set we have now a road network with 60 nodes. The corresponding distance matrix has 60^2 entries, equal to 3600 entries. We have to take into account that the distance between a node and itself is equal to 0, so all the entries in the diagonal will be equal to 0. Following that we have $3600 - 60 = 3540$ entries left to compile. Dealing with a symmetric matrix allow us to reduce this number to $3540/2 = 1770$. The symmetry property does not only allow us to reduce the number of queries but also makes easier to check that the triangle inequality property holds within the network.

Finally, the distance, speed and slope matrices modelled as described above are used as input of the emissions model.

4.4 Emissions Models

4.4.1 The Emission Factor

As a fulfilment of international treaties on climate change, European Union (EU) member states are required to report their annual emissions and honour their underlying obligations. Policy makers are interested in reliable estimates of emissions for current and future technologies. In the case of this work, we have decided to rely on methodologies established by EU research organisations accounting either on the credibility of these institutions and on the quality of the work done.

The impact on emissions of the transport sector is particularly remarkable as shown in section 1.1. To comply with real-world conditions (considering *Xpreso's* partners), in this study we focus on diesel light duty vehicles with a net weight lower or equal of 3.5 tons and maximum operating weight lower than 7.5 tons.

The basic relation between CO_2 emissions and travelled distance in average speed dependent models can be expressed in equation 4.4.1.

$$Emissions\,(g) \;=\; EF\,(g/Km) \,*\, Distance\,(Km) \tag{4.4.1}$$

Where EF stands for "Emissions Factor". Modelling the EF is the key point that will impact the final results. Usually, the emphasis is on the average speed while a range of constants and coefficients are adopted according to the vehicle and engine category.

As previously mentioned, in 2013, the European Parliament and the Council of the European Union released two regulatory proposals that will be mandatory by 2020. CO_2 emission targets for passenger cars and light duty vehicles in the EU, will be 95 g/km for the former, and 147 g/km for the latter. As our first EF we will apply the formal limit posed by the EU. While this is not significant for route finding purposes, since the emissions would be perfectly linear in the distance, it would be useful to understand whether or not our optimal solution, is within the limits posed by the EU2020 legislation.Therefore, the EU20 emissions model rely on equation 4.4.2.

$$Emissions\,(g) \;=\; 147(g/Km) \,*\, Distance(Km) \tag{4.4.2}$$

As can be seen in Figure 4.3, the emissions are linear in the travelled distance.

Thus, the longer the travel the higher the emissions level will be.

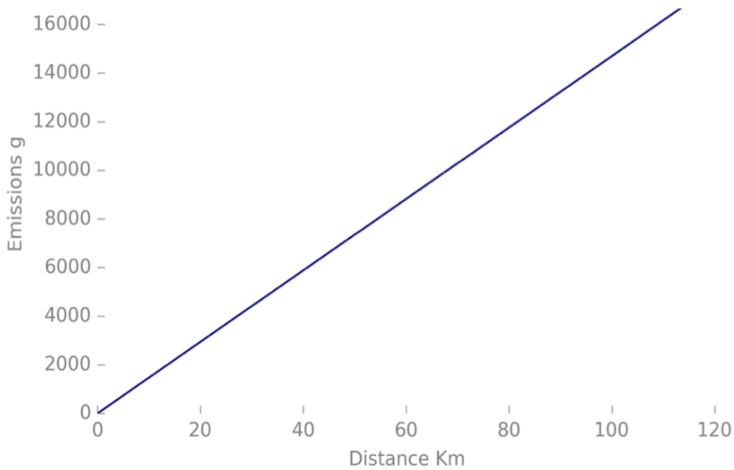

Figure 4.3: EU20 Emissions Model

4.4.2 Average Speed Emissions Model

Increasing the complexity of our approach, we model the EF using the average speed dependent method developed for diesel light duty vehicles complying with *EURO I*, (Council of European Union, 1993) emission standards. However, the *EURO I* standards established in 1993 have been gradually replaced by more restrictive requirements. Consequently it can be assumed that recently released vehicles have to be more efficient in terms of emissions.

Between 1996 and 1998 the *European Commission* funded the project *MEET*, leaded by Hickman *et al.* (1999) “Methodology for calculating transport emissions and energy consumption”. *MEET* has produced a comprehensive inventory of methods for estimating pollutant emissions and energy consumption from transportation. It covered all the vehicle technologies available at that time, differentiating for classes of road vehicles, as well as rail, shipping and air transport. The results from *MEET* are being incorporated in all subsequent studies currently used by member states of the EU.

It can be pointed out, that often the correlation coefficients between the rates of emission and average speed for the provided functions are low. However, the authors claim that the two main reasons for the poor correlation are: primarily the very large variability observed in the data and secondly it has been shown that there are remarkable differences between the emissions from different vehicles within the same category and even from the same vehicle when measurements are recorded in different laboratories.

In our opinion, what makes the model adequately reliable, is that according to the *MEET* report, the effect of most variables which are not directly modelled in the EF has been somehow statically captured thanks to the large variability of conditions (temperature, vehicle model, driving mode) in which data have been recorded. Furthermore, the large number of vehicles sampled is assumed to be fairly representative of the European vehicles fleet, and consequently the provided emissions factors are considered reasonably accurate despite the low correlations.

The average speed EF derived from the *MEET* report, for the considered vehicle category is represented in equation 4.4.3.

$$EF\,(g) \,=\, 0.0617 * V^2 - 7.8227 * V \,+\, 429.51 \tag{4.4.3}$$

Where V stands for speed expressed in Km/h. It can be noticed that when the speed is equal to zero the EF is equal to 429.51 g/Km. However, in an average speed model a speed equal to 0 is not consistent with the underlying assumptions because variables like red traffic lights, traffic jam or other conditions which could force the vehicle to stop while the engine is still on, are not considered. We have tried to imagine a real world application: a driver stopping at a customer's place, delivering a package; concluded this first job, aware that the next delivery is at a distance lower or equal to 100 meters, it is very likely that the driver will not move its van, but simply walk to the next customer and complete its task. In numbers this means that for a distance of 90 meters, travelled by car in 1 minute according to *Google Maps*, we have an average speed of 5.4 Km/h and this would produce according to the model 389 g/Km of CO_2. Considering a medium sized route generates 25,000 grams of CO_2 emissions we find out that 389 would be 1.6 % of the total. In order to increase the reliability of the model we round distances lower than 100 meters to 0. In these cases, the emissions associated to arcs shorter than 100 meters will be equal to 0. Now if we take into account that the smallest travel time that can be suggested by *Google Maps* is one minute, and the smallest distance allowed by us is 0.1 Km, follow that the smallest possible speed is 6 Km/h (0.1 Km/0.02 h) and the smallest possible amount of emissions is 38.47 g.

In Figure 4.4 the average speed dependent emissions model is graphically illustrated. Given a fixed distance of 10 Km, the emissions are plotted as functions of a range of speed values spanning from 0 to 120 Km/h. Looking at the graph we can notice that when the speed is equal to 0, the emissions are equal to 0. As said, while this is not coherent with the equation described above, this choice has been imposed within the body of our speed dependent emissions function to be consistent with the nature of the proposed average speed model. Furthermore, we can see that for small speeds, the emissions are particularly high.

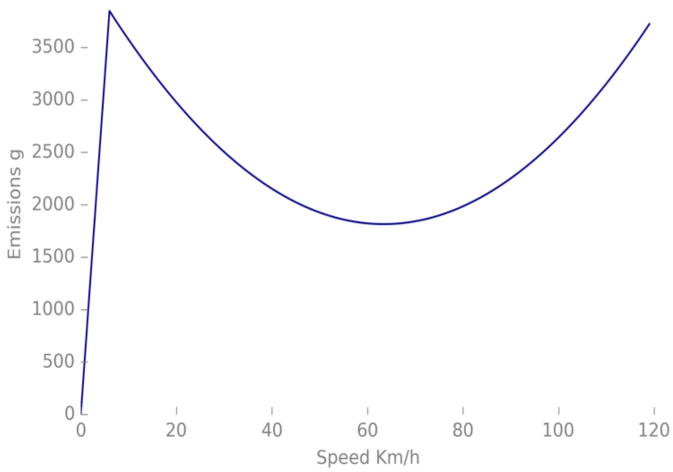

Figure 4.4: Speed Dependent Emissions Model

Summarising all the considerations discussed, our speed dependent emissions model can be formally expressed in equation 4.4.4.

$$Emissions\,(g) \;=\; \begin{cases} EF\,(g/Km) \,*\, Distance\,(Km) & \text{if } V \,>\, 0\,;\, Distance \,>\, 0.1 \\ 0 & \text{Otherwise} \end{cases} \tag{4.4.4}$$

4.4.3 Average Speed Gradient Weight Emissions Model

As mentioned, there is a large number of variables that can impact the amount of emissions per travelled distance unit. Now we focus our attention on the road gradient and vehicle load. The gradient of a road has the effect of increasing or decreasing the resistance of a vehicle to traction causing a corresponding effect on rates of emission and fuel consumption.

On the other hand, the driving resistance of a vehicle is influenced by the vehicle's weight because a higher vehicle mass requires higher power from the engine during driving, especially in acceleration (extremely frequent in city driving mode). Load considerations are particularly appropriate in the context of this study where we are trying to model the emission behaviour of duty vehicles used for parcels delivery.

The proposed approach is to use the previously defined speed dependent EF and update it according to an adjusting EF that will take into account the road gradient and the vehicle load.To adjust the EF we again base our assumptions in the *MEET* report. The adjusting EF, for the considered vehicle category, is defined by the equation 4.4.5.

$$adjusting\, EF \,=\, k \,+\, n * G \,+\, q * G^2 \,+\, r * V \,+\, \frac{u}{V} \tag{4.4.5}$$

Where V is the average speed in Km/h, G is the road gradient expressed in %, k is a constant and n, q, r and u are coefficients. Specifically, k = 1.27; n = 0.0614; q =-0.0011; r =-0.00235; u =-1.33. Those parameters are found in the empirical data used by Hickman *et al.* (1999) and are kept the same in our implementations.

The adjusting EF is then multiplied to the average speed EF to obtain the gradient and load dependent emissions factor in equation 4.4.6.

$$Gradient\, EF\, (g/Km) \,=\, EF * adjustingEF \tag{4.4.6}$$

The total amount of emissions is then calculated in equation 4.4.7.

$$Gradient\, Emissions\, (g) \,=\, Gradient\, EF\, (g/Km) * Distance(Km) \tag{4.4.7}$$

While speed and road gradient are directly input in the adjusting EF equation,

the weight has been included during the experimental phase of the modelling process. All the vehicles were loaded with an average load of 1000 Kg.

To illustrate, the adjusted emissions factor is graphically illustrated in Figure 4.5. Given a fixed distance of 10 Km we have calculated the emissions according to a road gradient ranging from -6% to $+6\%$ and a speed range spanning from 0 to 120 Km/h.

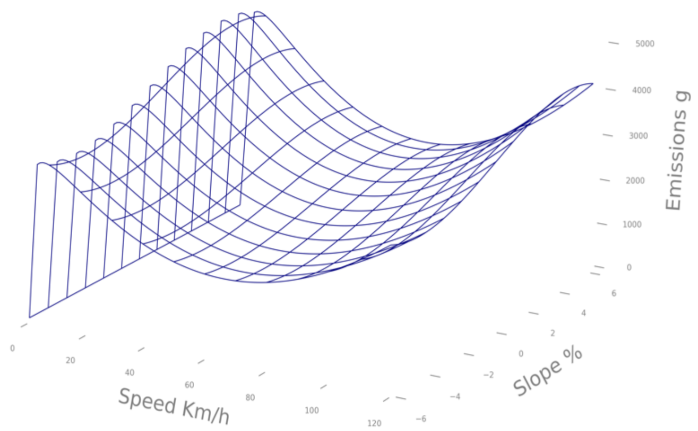

Figure 4.5: Speed Gradient Weight Dependent Emissions Model

The model behaviour is easier to understand when one variable is taken as constant. In Figure 4.6 is shown the behaviour of the function for three different fixed values of the slope. The shape is very close to the one in the average speed dependent model. As expected the emissions are lower while going downhill and higher on flat and hill roads.

Now, assuming the speed is constant at the levels shown in the legend just underneath the graph, and the travel distance is set equal to 10 Km, we can look at the emissions behaviour for different values of the road gradient. While going uphill increases the emissions level, it is the speed level that has the most significant impact on total amount of emissions (Figure 4.7).

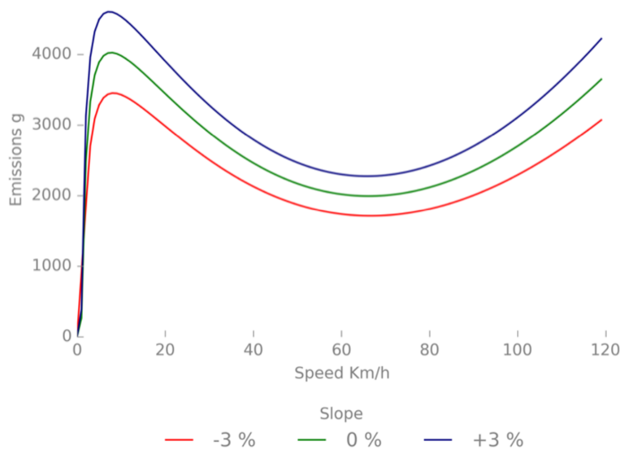

Figure 4.6: Speed Gradient Weight Dependent Emissions Model – Constant Slope

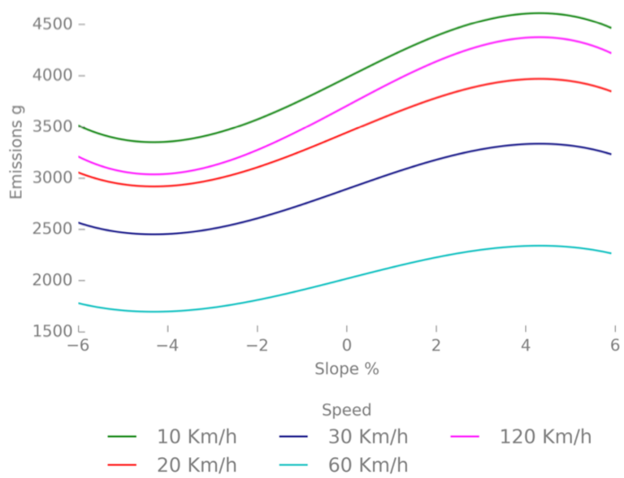

Figure 4.7: Speed Gradient Weight Dependent Emissions Model – Constant Speed

Additionally, to further improve our methods, we have attempted a post optimisation process trying to increase or decrease the emission factor associated with each arc of the optimal route according to the weight loaded at the expected time of the delivery.

4.5 Load Weight Model

As per the *Euro 2020* targets, light duty vehicles emissions per Km corresponding to 147 g/Km can be further increased by 0.96 g/Km each 100 Kg of load. If we imagine to load a van with 300 Kg we can have an average speed emissions factor corresponding to $147 + 3 * 0.96 = 149.88$ g/Km. Starting from this hint and looking for opportunities to post optimise our best solution, we have analysed the total weight of the load and the single packages weight distribution for the vans.

Looking the total weight loaded on a van (Figure 4.8) we have found that the average weight is around 350 Kg, with peaks that rarely goes close to 500 Kg. The minimum weights recorded are found to be around 100 Kg.

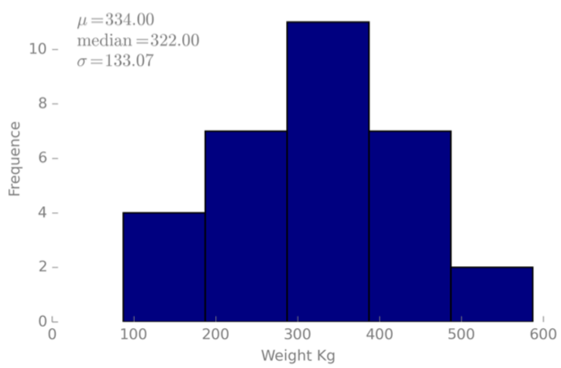

Figure 4.8: Total Loaded Weight Distribution In A Given Day

By analysing Figure 4.9, it can be noticed that 50% of the packages loaded on this randomly selected van have a weight equal to 2 Kg while the packages with a weight between 10 and 11 Kg represent less than 10 units and only 1 package has a weight between 14 and 15 Kg. The vast majority of the customers served by the analysed delivery company are offices which receive small parcels. Using this data we can infer that, at each stop the weight loaded on the van will more likely decrease by a small amount.

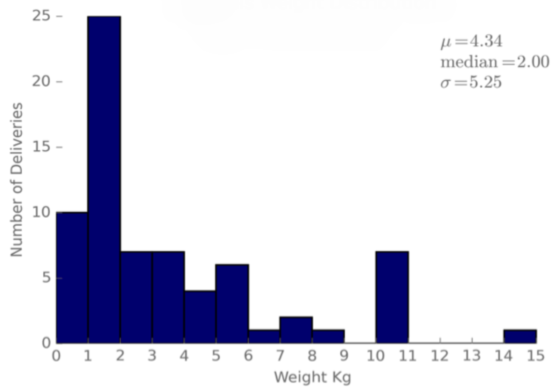

Figure 4.9: Average Weight Per Package In A Given Day

By plotting the average weight of a delivery van per time window as shown in Figure 4.10 it can be drawn that the weight decreases as the time passes. To supply a suitable model that can connect both weight and time we selected a linear regression approach.

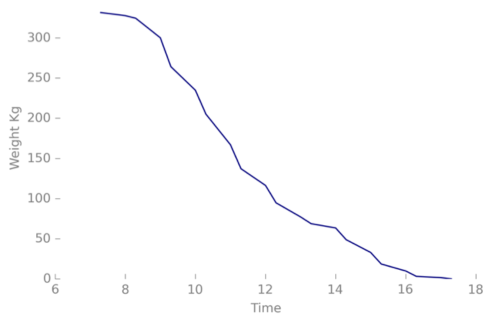

Figure 4.10: Total Loaded Weight Distribution Per Time Window

The first attempt has allowed us to obtain a coefficient of determination equal to 0.94. It can be noticed that while some of the observations are above the regression line others are below. This may be the main reason to explain the high coefficient of determination found. Thus, given the apparently non-linear behaviour of the data a non linear model could appear to be more appropriate in this case. However, empirical tests in our data have shown very accurate results using the linear mode in weight forecasting (Figure 4.11). The behaviour of the weight through the time is modelled using equation 4.5.1.

$$Weight\,(Kg) \;=\; -37.52 * Time \;+\; 599.65 \tag{4.5.1}$$

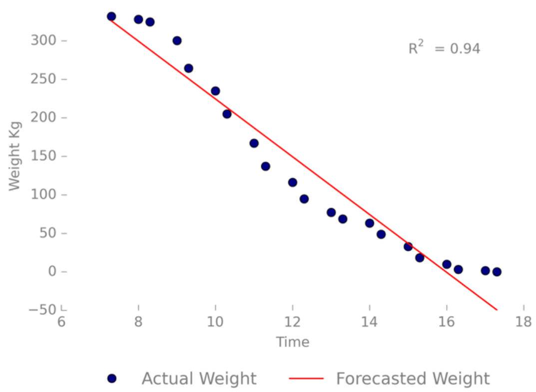

Figure 4.11: Forecasted Weight Vs Actual Weight

As it can be seen from Figure 4.11, in the last two hours the van is almost empty, with a load lighter than 50 Kg. The reason why the van is not completely empty is that some customers along the route were not available for the delivery and so the driver has to carry the parcels back to the depot. However, the decreasing trend would generate negative weight which have no practical sense. In order to prevent this flaw of the model, we have set the regression formula equal to 0 when a negative weight is found. This has allowed us to reach a considerable coefficient of determination equal to 0.96. The updated model the weight is described in equation 4.5.2. The adjusted model compared with the actual weights can be seen in Figure 4.12.

$$Weight\,(Kg) \;=\; \begin{cases} -37.52 * Time \;+\; 599.65 & \text{if } Weight > 0 \\ 0 & \text{Otherwise} \end{cases} \tag{4.5.2}$$

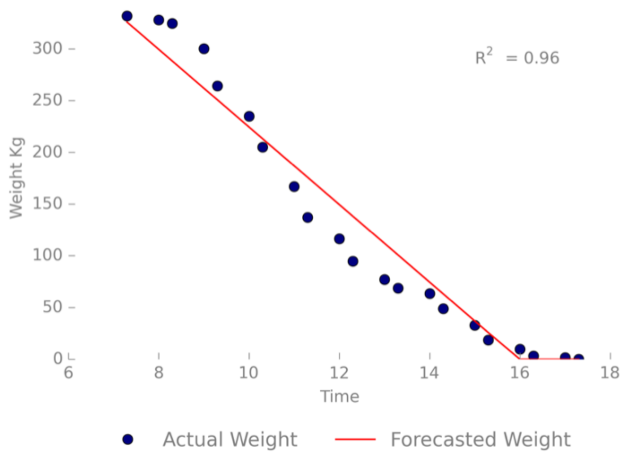

Figure 4.12: Adjusted Forecasted Weight Vs Actual Weight

We have attempted the inclusion of such model in the post optimisation phase of our routes. However, given the small impact on the optimal solution in our real-world data and the lack of any empirical measurement of the impact in emissions on the final results obtained, we set aside this hypothesis as further research opportunities.

4.6 Heuristic Approach

After a thorough review of the literature, we have considered several promising heuristic approaches. The algorithm methodology will be based on well established approaches of proven efficacy. At this stage we believe that a local search heuristic algorithm represents the most convenient alternative in the context of a real world application with the intent of corroborating the solution reliability and computational efficiency.

A combination between local search and population methods also seems reasonable. As seen in our literature review, the most promising heuristic methods applied VRP instances are based on hybrid versions of other methods, usually a combination of local and population search methods.

The main challenge is not only the elaboration of the algorithm; while this is not a trivial task, another substantial difficulty can be found in the use of the Emission Matrix as cost matrix in the objective function. The problem formulation can preclude or make very hard the application of some heuristic methodologies. In order to overcome this obstacle, we plan to use a top down approach, after having built up a functional classic implementation of a suitable heuristic based on a distance matrix we will explore all the possible modifications required to include the emissions in the formulation. The simplest scenario here is the substitution of the distances between two nodes with the expected amount of emission associated to that pair.

4.7 Problem and Model Definition

The final outcome after creating the emission models for the routes is to find optimal emissions routes for the real-world data. To achieve this outcome this work proposes the usage of a heuristic method to find optimal tours for both TSP and VRP variants. As presented before, heuristics represent the most common algorithms applied for both the TSP and the VRP in the literature, they provide a fast and easy-to-use framework to achieve either optimal or suboptimal solutions. As per the purpose of this work, the problem of finding optimal tours regarding emissions is separated in two different problems.

4.8 n-TSP Model

As presented, the decision of solving the TSP is made based on practical reasons, but also regarding the running time of algorithms. By construction, solving TSPs is usually faster than solving a full VRP every single day before deliveries start (number of nodes increase and more constraints play a role). Also, by solving a VRP instead of n-TSPs one can assign drivers to different regions that are already mapped to a specific employee by the delivery companies (altering the form how companies function in the real-world) or create some redundancy in the number of drivers and shift hours already determined. On the other hand, solving n-TSPs may provide suboptimal tours and consequently suboptimal costs/emissions for delivery companies. It is also possible to use the VRP solution to define the most accurate number of drivers necessary for a given day. However, this solution is usually not close to reality since drivers cannot be hired and dismissed on a daily basis in most companies. Yet improvements in this sense can be made if a longer period of time is considered and demands are estimated in advance.

Due to practical applications, parcel delivery companies usually assign drivers to predetermined regions. This practice allows the companies to improve delivery times by using the drivers' experience over a region to improve efficiency. Also, this practice do not cause much disruption in assigning drivers to non-familiar regions and to different shifts. For such reasons, this work will be first interested in solving n-TSPs (where n stands for the number of drivers available) for each driver instead of a full VRP for all parcel deliveries in a given day. After solving n-TSPs, the differences in solving n-TSPs and a VRP in terms of emissions, number of drivers and economic costs will be assessed in the final solution to provide a different view of the same problem.

To solve both the n-TSPs and the VRP this work makes use of a Genetic Algorithm (GA), introduced by J. Holland in 1975, combined with local search heuristics. In the literature, Genetic Algorithms are not the first choice for solving TSPs and VRPs. This is usually due to the fact that these heuristics demand larger running times when compared to other heuristics like Simulated Annealing (SA) and Tabu Search (TS), that are considered the best heuristic approaches (alongside with some local search heuristics) for solving TSPs and VRPs. Nevertheless, good implementations of GAs for the VRP can be found in the work of Prins (2001) and Baker and Ayechew (2003). GAs also offer a wide range of setting parameters and are one of the most used heuristics in optimisations problems. Our choice for the GA also takes into account the easiness of implementing the algorithm and the opportunity to apply a challenging method to the *status-quo* in solving TSPs and VRPs. Our implementation follows the main idea of a Genetic Algorithm (crossover, mutation, population management) and incorporates some of the ideas developed by Prins (2001) and Potvin (1996). Their implementations and tests provide insightful frameworks that will be used as a starting point for our algorithm.

4.8.1 Problem Assumptions

For the purpose of solving the TSP using an emission model some assumptions are required. Our model assumes that each pair of customers (i, j) has an associated non-negative travel cost $c_{i,j}$ that varies according to the emission model/distance matrix used. To successfully apply our model in solving TSP instances, symmetry is required $c_{i,j} = c_{j,i}$. Indeed, symmetry is not an issue when we solve the TSP problem using symmetric road distances, the EU2020 model or the model based on the average speed. However, using the model that takes into account the road gradient, automatically generates an asymmetric cost matrix. For the purpose of this work, we assume symmetry even when using the gradient model, using the lower diagonal matrix as our travel costs. This assumption is made to guarantee that our model will be able to improve a set of initial solutions properly. However, to fully apply the gradient method further research may be done in implementing an Asymmetric TSP solver in order to comply with such characteristics of the G-TSP and G-VRP.

Additionally, our TSP model does not assume time constraints. Our final data contains a set of locations already visited by each driver in a given day. Therefore, the data structure allows our formulation to drop time or capacity constraints, given that completing the entire route is by construction, feasible. Lastly, the input cost matrix must satisfy the triangle inequality (4.8.1).

$$\forall i; j; k \in V : c_{i,k} + c_{k,j} \geq c_{i,j} \tag{4.8.1}$$

The final goal of the algorithm is to minimise total cost of the trips. As we shall see, the GA is flexible enough to handle more complicated objectives and constraints, if needed.

4.8.2 GA Design

The main points of the GA design were based in the work of Prins (2001) and Potvin (1996). The main design ideas are described as follows:

1. TSP Permutation Chromosomes - Each chromosome is defined as a tour starting and ending at the depot ($i = 0$) covering all nodes in the network.

2. Fitness and Tour Cost - The fitness of each chromosome is defined as the total tour cost.

3. Small Populations / No Clones Allowed / High Elitism - We aim to keep the populations in our GA implementation small, the main idea is to keep the implementation fast and allow enough diversification for improvements in the local search procedure. Also for that reason we do not allow solutions with the same fitness level in the population and we apply elitism to generate better offspring more quickly. Even though some running time can be saved if we do not check for the presence of clones in the population.

4. Inclusion of good heuristics solutions in the initial population - The inclusion of construction heuristics in the initial population also aims to reproduce better solutions quickly.

5. Mutation: Local Search Method - Mutation is performed as a Local Search operator instead of simple moves of swapping nodes. This allows for a faster implementation of the GA, reducing the number of members in the population.

6. Incremental Population - Population is managed in an incremental manner, at each step an offspring can substitute a mediocre chromosome.

7. Exploration phase, followed by restarts - Restarts strategies are added to the GA to improve the search for new neighbourhoods in the population in hard problems.

The main ideas presented here are similar to those present in the work of Prins (2001) and Lacomme *et al.* (2001). As a matter of fact, these authors provide good implementations of GAs for the VRP and CARP based on those key ideas with results that surpass Tabu Search (TS) algorithms in some VRP instances at the time they were constructed (Prins, 2001). We use those ideas to give foundation to our implementation as we make some amendments in the GA structure. Our implementation employs the usage of 3-OPT moves and higher elitism in the population construction, leading to a different implementation.

4.8.3 Chromosome Structure

Like in most GAs for the TSP, a chromosome is simply is a sequence (permutation) S of n client nodes. The fitness $F(S)$ is the total cost of this solution starting and ending at the depot ($i = 0$).

0 1 2 3 5 6 4 7 8 10 11 9 0

Figure 4.13: Example of a chromosome

4.8.4 Crossover

Given their structure, our chromosomes can undergo any type of classical crossover techniques. In the case of this work we based our decision of the most suitable crossover method based on the work of Potvin (1996). In his work, the author evaluates different methods of crossover for TSP-like chromosomes. According to the author, Ordered Crossover (**OX**) and Edge Recombination (**ER**) provide the best results in terms of solution quality for TSP instances. In our work, we chose **OX** to be the crossover method. **OX** was implemented in $O(n)$ and provides an easy to use and implement technique for us, whereas

ER requires additional data structures to compute the edges incidents to a given node. However, in further research paths, the application of **ER** and other crossover techniques is recommended since it can possibly improve the quality of the solutions found.

The example in Fig.4.14, show how **OX** performs the crossover and constructs the first child (C_1). First two cutting points i and j are randomly selected in the first parent P_1. Then the substring is copied from P_1 to C_1. Finally, the rest of C_1 missing nodes are completed by crossing circularly P_2 from $j + 1$ onward. C_2 is obtained by swapping P_1 and P_2.

$$P_1 : 1\ 2\ 3 \mid 5\ 6\ 4 \mid 7\ 8\ 10\ 11\ 9$$

$$P_2 : 8\ 9\ 2 \mid 3\ 1\ 7 \mid 5\ 11\ 10\ 4\ 6$$

$$C_1 : 3\ 1\ 7 \mid 5\ 6\ 4 \mid 11\ 10\ 8\ 9\ 2$$

$$C_2 : 5\ 6\ 4 \mid 3\ 1\ 7 \mid 8\ 10\ 11\ 9\ 2$$

Figure 4.14: OX example

4.8.5 Mutation

Or algorithm uses Local Neighbourhood improvement heuristics as mutation operators. The main reason for this decision is to provide a much faster convergence of the GA, while taking advantage of the inherent diversification provided by the GA . According to Potvin (1996) a pure GA with no LS heuristics would take much longer to converge and tests showed that the solution for a 100-city TSP can stay 25% above the best solution, whereas a GA powered with a LS operator can improve the solution and stay at 1.7% above optimum on average. In our tests the presence of the LS as a mutation operation has improved the gap from 67.81% to 1.08% using the 76-city TSP instance eil76. We tested our implementation with and without LS procedures to select the best parameters for our algorithm. The results support the decision of maintaining LS heuristics as mutation operators as they provide a much faster convergence rate (even if it is a just a local optimum).

A child produced by the crossover operator is mutated with probability p_m. Our first Local Search method scans in $O(n^2)$ the neighbourhoods listed below, LS then performs a 3-OPT operation with probability p_{3opt}. This LS is based in the LS developed by Prins (2001) in his very efficient GA for the VRP. Our implementation in this work is quite naive and changes slightly some design decision made by the original author, for example by the inclusion of an additional 3-OPT step and the fact that this implementation does not consider different trips, since we are dealing with a TSP in our first case.

Local Search: Each iteration of LS scans all possible pair of client nodes (i, j). For each pair of nodes the following moves are attempted in the order described, $i+1$ and $j+1$ are the successors of i and j in their tours respectively.

First Phase:

1. Remove i then insert it after j.
2. Remove $(i, i+1)$ and insert them after j.
3. Remove $(i, i+1)$ and insert $(i+1, i)$ after j.
4. Swap i and j.
5. Swap $(i, i+1)$ and j.
6. Swap $(i, i+1)$ and $(j, j+1)$.
7. 2-OPT move between $(i, i+1)$ and $(j, j+1)$, resulting a tour with edges $(i, j+1)$ and $(i+1, j)$.

Second Phase:

1. 3-OPT *algorithm* is called to attempt improvements with probability p_{3opt}. It swaps edges $(i, i+1)$, $(j, j+1)$, $(k, k+1)$ into a feasible permutation of those edges.

Moves 1-7 are performed in the following manner. First, all moves are tested for all pair of nodes (i, j). The move that leads to the best improvement in the tour is then selected at each iteration. The LS stops when no other improvement is found in the tour as can be seen in Algorithm.1. Here the $Change_k$ functions correspond to the cost change when the move k is performed. The function SwitchTour(i_{best}, j_{best},$move_{best}$) performs the change in the tour correspondent to $move_{best}$. The decision to select only the best movement at each iteration intends to reduce the number of times we scan a tour. Since every iteration of moves 1-7 takes $O(n^2)$ the number times we actually stay looking for better neighbourhoods can grow if we make the decision to stop at the first improving move like in the work of Prins (2001). However, this decision can be reviewed if more efficient implementations of this heuristic is attempted, our method is far from being the most effective in terms of running time. An improvement for this method could be the usage of K-D trees to efficiently select nearest neighbours nodes at each iteration for spatial problems. This type of implementation can drastically reduce the running time of the LS heuristic.

minchange ← 0
for $i \leftarrow 1$ **to** $toursize$ **do**
    for $j \leftarrow i+1$ **to** $toursize-1$ **do**
        $ch_1 \leftarrow Change_1$(i,j);
        $ch_2 \leftarrow Change_2$(i,j);
        $ch_3 \leftarrow Change_3$(i,j);
        $ch_4 \leftarrow Change_4$(i,j);
        $ch_5 \leftarrow Change_5$(i,j);
        $ch_6 \leftarrow Change_6$(i,j);
        $ch_7 \leftarrow Change_7$(i,j);
        change ← min(ch_1,ch_2,ch_3,ch_4,ch_5,ch_6,ch_7)
        move ← min.Index(change)
        if $change < minchange$ **then**
            $move_{best}$ ← move;
            $i_{best} \leftarrow i$;
            $j_{best} \leftarrow j$;
            minchange ← change;
        end
    end
end
tour← SwitchTour(i_{best}, j_{best},$move_{best}$)

Algorithm 1: Local Search Main Algorithm

4.8.6 2-OPT and 3-OPT Heuristics

In the literature improvement heuristics are one of the preferred methods for solving TSP models. In that regard, heuristics that perform edges swap are one of the most effective found in recent papers(eg. Lin-Kernighan (LK) and Lin-Kernighan-Helsgaun (LKH) heuristics). In this work we use k-opt heuristics, which are related to a robust LK since they both use k-opt moves. Our idea is to use 2 and 3-opt moves to find better locally better tours and use those as new chromosomes in our population.

The 2-OPT algorithm corresponds to removing two edges form a tour, and reconnecting the two sub-tours created by the deletion of those edges. In the 2-OPT case there is only one way to reconnect such edges and keep the tour feasible. The deletion and insertion of the tours is repeated for all pair of nodes (i, j) in the tour as long as a new shorter tour is found. In our implementation we first run through all nodes in $O(n^2)$ to select a 2-OPT move that leads to the most reduction in cost (happens inside the first phase of LS). Every pair of edges (for example $(i, j+1)$ and $(j, i+1)$) is checked and if an improvement is possible $(cost(i, i+1) + cost(j, j+1) > cost(i, j+1) + cost(j, i+1))$ the pair of nodes is saved as the best move seen. The procedure is repeated until no further improvement can be done. The algorithm is then called again with the updated tour to search for more improvements. The moving and reconnecting of edges leads to a local optimum route. It is important to notice that in the case of our TSP model the 2-OPT moves occur at the same with the other moves in LS, the best move selected at each iteration is the one that leads to the best reduction in cost, which can mean that for some tours 2-OPT moves may not be performed.

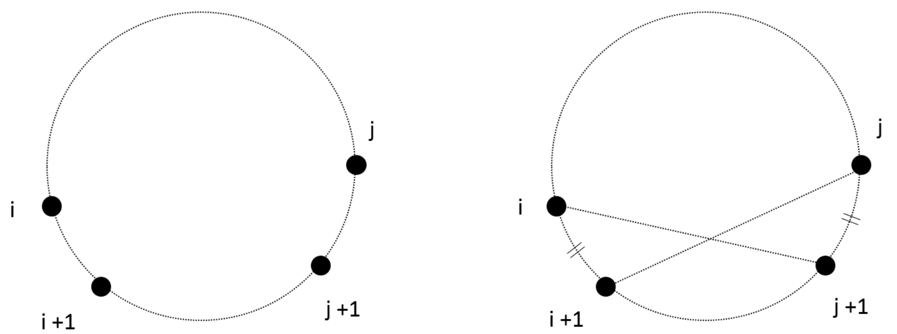

Figure 4.15: 2-OPT Move

Differently from the 2-OPT moves which can happen every time a mutation happens. The 3-opt algorithm is called with probability p_{3opt}.The 3-OPT algorithm works in a similar fashion to the 2-OPT. However, instead of removing two edges, the algorithm attempts to remove 3 edges and reconnects them forming a feasible tour. There are 8 ways to perform a 3OPT exchange, but only four correspond to real 3-opt moves. In our implementation, only those four pure 3-OPT moves are tested at each iteration and the one that leads to the best reduction in cost is selected. Moreover we randomly select a node to start looking for improvements ate each call of the algorithm, this aims to teak the algorithm for looking at different neighbourhoods and also aims to save running time. Generally a 3-OPT exchange produces better solutions, but it is much slower $O(n^3)$ than the 2-opt algorithm (if all 8 possibilities are tried a 3-opt tour is also 2-OPT optimal). For that reason, we use a stochastic application of the 3-OPT algorithm.

4.8.7 Population Structure

The population P is implemented as a list of n chromosomes sorted in increasing order of Fitness $F(P_i)$. That means that our best solution is always located at position P_1. We keep a tuple of 3 values for each chromosome $(fitness, tour, cost)$ this allows us to reduce the number of times we need to recalculate the tour cost, and add a possibility of rescaling fitness values. Identical solutions are not allowed in our implementation. This aims to avoid the reproduction of locally optimal solutions misleading the algorithm structure into that direction. However, one should notice that we also prevent different tours with the same cost in our population. Since we are interested in minimising the final emissions cost, this is not considered a problem in the population evolution (except that a different solution with the same cost could give rise to a better final solution, this problem however can be covered by running the algorithm more than once),for simplicity the algorithm always keep the first solution with a determined Fitness. The exact detection of clones are preferred in our case (using a Bisection Search), since no improvement has been seen in defining a minimum fitness threshold between solutions.

At the start, each initial population has three good solutions to compose the initial population. We make use of three construction heuristics to provide good initial solutions to our initial population. The Nearest Neighbour Heuristic (NN), the Clarke and Wright Heuristic (CW) (sequential version) and the Random Insertion Heuristic (RI) are the construction heuristics used to provide good initial guesses in our population. These three heuristics provide good and fast initial solutions for our algorithm. Further information on the construction heuristics can be found in the work of G. Clarke (1964) and Nilsson (2003). The rest of the population is completed with random tours starting and ending at the depot.

At each iteration of the GA each chromosome can be replaced with a new chromosome that has gone through crossover and mutation with probability p_m. The new child only enters the population if it has a better fitness value than the current chromosome to be replaced (very high elitism), if not the offspring is discarded and a new iteration of the algorithm is called. To check if the Fitness value is already present in the population we make use of the fact that the population is always sorted. Thus, we use a binary search algorithm to search for a Fitness in the population. This implementation allows us to check for clones in $O(log(n))$ using $O(1)$ space.

4.8.8 Restarts

The GA performs its main phase and stops when it reaches a maximum number of productive of offspring or when it reaches a maximum number of iterations with the same optimal value. The GA is restarted using the partial replacement procedure inspired by Cheung, 2001. His technique was amended to provide a faster implementation and can be described as follows:

1. Generate n random new tours $a_1, a_2, a_3, ..., a_n$ which are different from the tours in the population, and arrange them in descending order of fitness.

2. For each a_i, if the fitness of a_i is better than the worst fitness in the population then replace it with a_i. Otherwise, crossover a_i with all other

members in the population. Update the population by updating the worst tour with the best tour generated by the crossovers if its fitness is better than the current worst fitness.

As it can be seen our implementation is slightly different than the one described by Cheung. First, we can generate as many candidates as necessary and we make no restriction in the minimum number of input members in the population. In other words, if no improvement is found in the restart procedure, the GA is restarted with the same population for a new series of crossovers and mutations. Our implementation aims to let the GA move ahead if no inputs have been performed, by doing this we assure that the GA will not be at the restarting phase for a long period of time. This restarting technique can also be seen a type of migration, each time n new chromosomes try to enter the population by copying the populations best characteristics.

4.8.9 GA Main Phase

In our implementation parents are selected using the *binary tournament method*. For each parent, two chromosomes are randomly selected from the population, the chromosome with better fitness is then selected for reproduction. OX is then applied to generate two children (C_1 and C_2) after two parents have been selected. One child is then randomly selected to replace a chromosome P_k in the population (we have noticed no difference in the solution quality if we select a child with the best fitness). Where k is a random integer between the median and the length of the population (n). The child is inserted if the child fitness is not already present in the population and if its fitness is better than the current candidate to leave the population. The child enters the population in the position k and the entire population is re-sorted to maintain its structure.The main phase of the algorithm can be described as follows:

1. Population Initialisation:

    (a) Generate the first solution (P_1) using the CW heuristic. Add to the population.

    (b) Generate the second (P_2) solution using the NN heuristic. If $F(P_1)$ is equal to $F(P_2)$ skip. Otherwise, insert it in the population.

    (c) Generate the third (P_3) solution using the RI heuristic. If $F(P_1)$ is equal to $F(P_3)$ or $F(P_2)$ is equal to $F(P_3)$ skip. Otherwise, insert it in the population.

    (d) For the rest of the population generate random tours P_k. If $F(P_k)$ is not in the population, insert. Do until the length of the population reaches n. Sort the population in increasing order.

2. Main Loop of the GA:

    (a) Set $\alpha = 0$ and $\beta = 0$.

    (b) While $\alpha < \alpha_{max}$ and $\beta < \beta_{max}$ do:

        i. Select two parents P_1 and P_2 by binary tournament.

        ii. Apply OX and generate C_1 and C_2.

        iii. Select one child randomly C, among (C_1, C_2).

        iv. Select $r = random$. If $r < p_m$ then mutation $C = Mutation(C)$.

        v. Select $k = random.integer(\lfloor n/2 \rfloor, n)$.

        vi. If $F(C)$ not in the population and $F(C) < F(P_k)$. Insert C in the population at position k. $\alpha = \alpha + 1$. Sort the population.

        vii. If P_1 has not changed. $\beta = \beta + 1$

Figure 4.16: GA Main Phase

Given the structure of the GA we need to maintain parameters to control the number of successful offspring in our implementation and select the best mix between population size (n), mutation rate (p_m), 3-OPT rate (p_{3opt}), number of restarts(r), number of chromosomes that attempted to enter the population at each restart (ρ), maximum number of successful offspring (α_{max}) and maximum failures to improve the best solution (β_{max}) to achieve interesting solutions to our problems. As noticed by Potvin (1996) usually when we increase the size of the population we achieve better solution for the TSP. However, since our mutation is a very strong Local Search heuristic a large population combined with a high mutation rate can lead to an impractical algorithm in terms of running time and the parameters should be used with parsimony.

4.8.10 Model Evaluation

In our real world data we have available problems with nodes between 21 and 164 stops for each of the drivers. Thus, to assess the solution quality of the algorithm we tested our implementation in problems with similar amount of nodes to mimic the same structure in our real data. We are interested in evaluating both the final value found by the heuristic and its running time. It is important to mention that our implementation is not the most efficient implementation and it is not aimed to be. Further improvements in running times can be achieved by exchanging our implementations with much more efficient implementations. For example, our LS heuristic could be improved by the usage of K-D trees to store the nearest neighbours of each node, or by the application of parallelism computing as can be seen in the work of Kamil Rocki (2012); and Blazinskas and Misevicius (2011).

Furthermore, we selected data in the TSPLIB whose optima are already known to test the efficacy of our implementation and assess the results. Our instances contain from 29 to 198 nodes and the results found are presented in the Table 4.2 below.

4.8.11 GA Parameters

To determine our GA parameters we tested our model on TSPLIB instances varying n, p_m, ρ and presence of initial solutions. As expected, different configurations are more suitable for different problem instances. However, our goal was to find a set of standard parameters that could be used as default to solve all the instances and compare the results. As a result of such tests, two main configurations were chosen to be tested.

1. One configuration reduces the amount of chromosomes in the population to no more than $n = 30$ while maintaining a high mutation rate ($p_m = 0.1$), a low number of restarts ($r = 5$) and the attempt to insert ($\rho = 8$) new chromosomes at each restart (GA1).

2. The other configuration increases the amount of individuals in the population to $n = 160$ and restarts the algorithm $r = 10$ times, while maintaining the mutation rate at $p_m = 0.005$ and restarts with $\rho = 8$ new chromosomes attempts (GA2).

In both cases the number of successful offspring and number of non-improvements is set to 1000 ($\alpha_{max} = \beta_{max} = 1000$ and $p_{3opt} = 0.5$).

As we shall see below, GA1 is more accurate in general and produces the optimal more often than GA2. However, GA2 has faster running times and does not degrade the final solution in more than 0.67% for the tests done. It is also important to notice that the GA can support a variety of parameters combinations and can reduce/increase running times according to its configurations. Also, by improving some implementations of this work, one could ameliorate the running times for both GA1 and GA2 and use them as a practical solver for the TSP problem.

4.8.12 Results on TSPLIB Instances

As we can see in the table below the two GAs, (GA1 and GA2) were evaluated using TSPLIB benchmark instances with known optima. Both algorithms were applied using a Core i7, 8 GB ram laptop running Windows 7. All code and structures were developed in Python and all the running times represented in the table correspond to real running times on the described machine.

GA x TSPLIB								
Instance	n	Optimum	GA1	Gap(%)	RT1(s)	GA2	Gap(%)	RT2(s)
bays29	29	2020	2020	0	5.76	2020	0	2.2
eil51	51	426	426	0	34.06	426	0	9.09
berlin52	52	7542	7542	0	23.29	7542	0	7.66
eil76	76	538	538	0	114.95	538	0	28.09
pr76	76	108159	108159	0	76.24	108159	0	13.61
rat99	99	1211	1211	0	158.71	1211	0	37.18
kroA100	100	21282	21282	0	145.03	21282	0	45.63
eil101	101	629	629	0	266.36	631	0.32	60.24
ch130	130	6110	6110	0	379.75	6151	0.67	114.09
ch150	150	6528	6570	0.64	551.65	6528	0	154.65
d198	198	15780	15803	0.15	821.43	15900	0.76	319.21
mean	NA	NA	NA	0.07	234.29	NA	0.16	71.97

Table 4.2: GA-TSP Results: TSPLIB Instances

It is worth mentioning that both algorithms were applied 20 times for each instance and their best results achieved is represented in the table. However, for all the instances tested, GA1 was able to find the optimal in the first run, while GA2 took several trials to reach its best performance. Another important factor about GA1 is that we set 5 restarts as its default configuration, but in reality the optimum was found in the first or second restart for almost all instances tested.

Thus, to really evaluate running times one should take into account that it is necessary to run GA2 more than one time to find its best solution and that GA1 has overestimated running times. The results provided in this table were quite exciting, since both configurations can either achieve the known optimum or stay within a low GAP range.

4.9 VRP Model

To solve the VRP this work also makes use of a Genetic Algorithm (GA) combined with local search heuristics. We change some design decisions made by the previous model to make it able to deal with a VRP problem instead of a TSP. In the next sections we will describe the changes made in the algorithm to make it able to deal with a G-VRP.

4.9.1 Problem Assumptions

The same assumptions of symmetry and triangle inequality 4.8.1 must hold for the VRP case, road distances, speed, travel time and emissions are adjusted in the same fashion as for the TSP model. However, a new set of constraints are introduced in the problem. To break the VRP into smaller trips, some type of trip delimiter constraint(s) must be defined. In the basic VRP implementation the capacity of the vehicles and customer's demands are the variables used to break a big tour into smaller trips. Usually, a trip is broken when the demands exceed the vehicle capacity (CVRP) but other constraints are also common as seen in 3.2.

In our case, demands are not used to break the VRP solutions into smaller trips, as noted by the business subject matter expert our vehicles possess enough space to deal with all deliveries for a group customers. As seen before, a standard delivery van can usually carry up to 3.5 tons while in this study we have dealt with vehicles loaded on average with 350 Kg. Since parcels have on average 4 Kg or less, capacity are not our main concern. Nevertheless, there is another very important variable for our case that matters the most for delivery companies in the real case: time.

Time is very important for such companies given that through time reduction the deliveries can become financially more efficient. By reducing time spent on traffic, companies can reduce fuel costs, increase customer satisfaction and comply with driver's shift hours. In order to use time as our trip delimiters it is made necessary to determine the travel times between locations and service times for each of the customers. Using the data extracted from the Google API we already have the information of travel time between locations. Thus, to complete the calculations we only need to calculate service times for each one of the customers. Unfortunately, this information is not available in the real data provided. For this reason we set the average service time as three minutes for all customers. This information comes from previous analysis performed by Xpreso in their data and represent the average time a driver takes to deliver a parcel.

We use the time constraints in the following manner.Each driver is assigned with a maximum number of shift hours h_{max}. Thus, each driver should be able to go and come back from any location i before h_{max} (4.9.1). Moreover, in a group of deliveries S the sum of the travel times $t_{i,j}$ between all nodes (i, j) in a tour plus service times σ_j cannot exceed h_{max} (4.9.2).

$$\forall i \in V : t_{0,i} + \sigma_i + t_{i,0} \leq h_{max} \tag{4.9.1}$$

$$\sum_{\forall i,j \in S} t_{i,j} + \sigma_j \leq h_{max} \tag{4.9.2}$$

4.9.2 GA Design

The GA for the VRP follows the same main design decisions made for the TSP case. Points 1 to 7 defined previously in section 4.8.2 also form the general idea of the GA for the VRP. However, the decision to perform TSP-like permutations in the chromosomes has some implications for the method. By keeping the same permutation structure the chromosomes can undergo classic crossover method without any further treatment. This feature comes at a cost, since we are undergoing permutation on TSP-like chromosomes it is necessary to carry a transformation of such chromosomes into VRP solutions in some sort of way.

To perform the transformation of TSP-like chromosomes into VRP solutions, this work bases itself in the work of Prins (2001). In his implementation, the author first builds good TSP solutions (it may be interpreted as a single vehicle covering all the customers in a trip, by relaxing all other constraints) to posteriorly build trips using an optimal splitting procedure: Split. We use Split as our method to optimally split TSP chromosomes into m trips. The theory behind the Split is explained by Prins as follows “First, there exists at least one optimal chromosome (consider any optimal DVRP solution and concatenate the lists of nodes of its trips). Second, if a crossover generates such a chromosome, the corresponding optimal DVRP solution can always be retrieved with Split. Third, the task of finding the best chromosome is left to the intrinsic parallelism of the GA.”

4.9.3 Chromosomes and Split

Our implementation of Split uses maximum travel time constraints instead of load constraints (Prins' version) to break the chromosomes into trips. However, both implementations share the same basic ideas. First, Split constructs an auxiliary graph $H = (X, A, Z)$. X contains $n + 1$ nodes indexed from 0 to n. A contains one arc (i, j), $i < j$, if a trip visiting customers S_{i+1} to S_j is feasible in terms of time constraints, condition: 4.9.3. The weight $z_{i,j}$ of (i, j) is equal to the trip cost.

$$\forall (i,j) \in A : t_{0,S_{i+1}} + \sum_{k=i+1}^{j-1} (t_{S_k,S_{k+1}} + \sigma_{S_k}) + \sigma_{S_j} + t_{S_j,0} \leq h_{max} \tag{4.9.3}$$

$$\forall (i,j) \in A : z_{i,j} = c_{0,S_{i+1}} + \sum_{k=i+1}^{j-1} (c_{S_k,S_{k+1}}) + c_{S_j,0} \tag{4.9.4}$$

An optimal solution for S (S is a sequence of n client nodes) corresponds to a shortest path from 0 to n in H. Since H has no circuits this evaluation is not very time consuming. The top of Figure 4.17 shows a sequence $S = (a, b, c, d)$ with $h_{max} = 10$, $\sigma_j = 0$, costs in bold and travel times in brackets. H in the middle contains for instance one arc ab with weight 50 for the trip $(0, a, b, 0)$. The shortest path has three arcs and its cost is 140 (red lines). The lower part gives the resulting VRP solution with three trips.

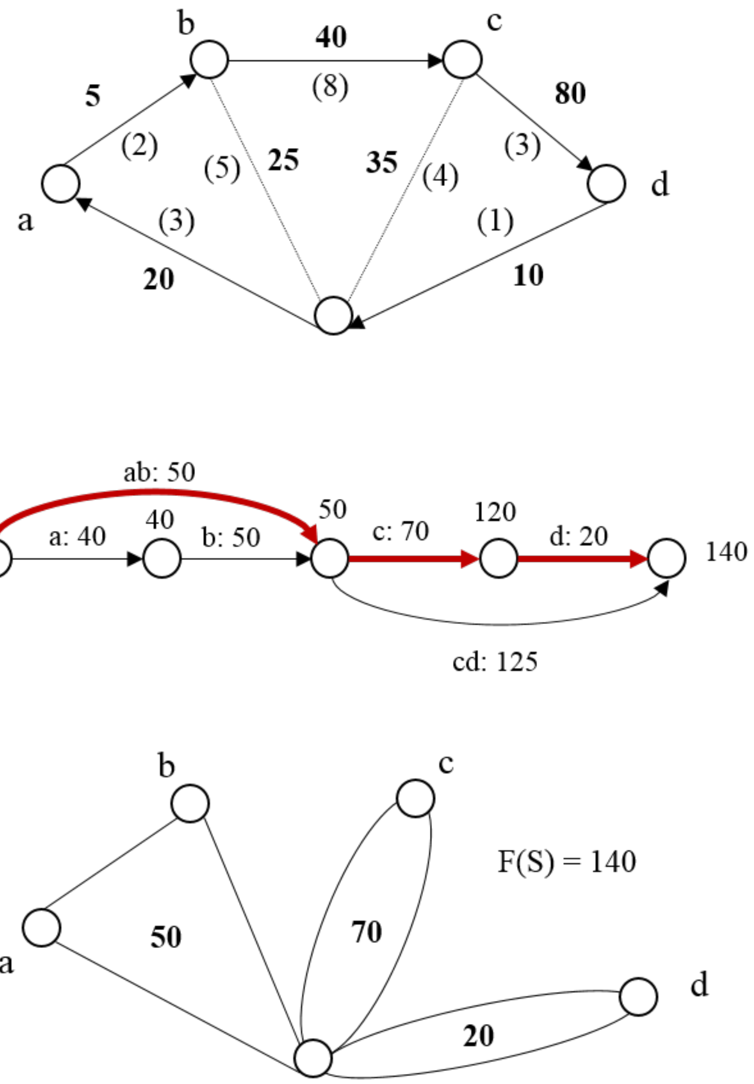

Figure 4.17: Split Procedure

To implement the Split procedure we use the same implementation applied by Prins. The algorithm is a version in $O(n)$ space that does not generate H explicitly. It computes two labels for each node $j = 1, 2, \ldots, n$ of X: V_j, the cost of the shortest path from node 0 to node j in H, and P_j, the predecessor of j on this path. The required Fitness $F(S)$ is given at the end by V_n. Refer to (Prins, 2001). The vector of labels P is used to rebuild the trips at the end of the GA. The procedure, also developed by Prins (2001) builds up to m trips (worst case with one vehicle per demand). Each trip is encoded as a list of clients starting and ending at the depot.

4.9.4 Crossover

The fitness $F(S)$ of a chromosome is calculated as the sum of the costs of all its trips. Here it is not necessary to use a structure for saving the unfitness of a solution since all out chromosomes undergo a Split procedure before entering the population again (guarantees feasibility). The same ordered crossover (OX) is applied for the GA-VRP. Split is called after a child is created to optimally create a set of trips for a given chromosome.

4.9.5 Mutation

In order to gain speed our LS changes for the GA-VRP. It maintains a $O(n^2)$ search for improvements, but reduces the number of neighbourhoods searched. Our new LS can be described as:

1. The 2-OPT algorithm is attempted and a move is made every time a swap leads to a reduction in $F(S)$.

2. After the 2-OPT algorithm is completed an improvement is searched by going through all nodes (i, j), removing i and inserting it after j. If an improvement is found the search stops.

This new LS is less sophisticated than the first one (TSP Model), but as we shall see in the future also produces reasonable good results for VRP problems. The decision to simplify the LS is based on running times of the algorithm. More complicated LS would increase the running time of the algorithm abruptly leading to small improvements in the final solution. Our aim with this LS is to keep the algorithm fast enough to deal with VRPs up to 150 nodes. It is worth mentioning again that our implementation can be improved by the usage of more efficient data structures.

4.9.6 Population Structure

The population P is implemented again as a list of n chromosomes sorted in increasing order of Fitness $F(.)$. That means that our best solution is always located at position P_1. We keep a tuple of 2 values for each chromosome $(fitness, tour)$, trips are calculated at the end of the algorithm .

Identical solutions (in terms of *Fitness*) are again not allowed in our implementation and the same construction heuristics are used to produce good initial solutions. It is important to notice that the version of the CW used here is the same previously constructed for the TSP method. Meaning that CW is applied as a single vehicle route, and posteriorly trips are calculated using the Split procedure. The other two starting solutions are also calculated in the same fashion.

The rest of the population is again filled with random tours. The replacement of chromosomes happens in the same fashion and the controlling points: maximum number of successful offspring (α_{max}) and maximum number of runs with no improvement (β_{max}) are present do stop the search.

4.9.7 GA Main Phase and Restarts

Binary selection is again used for parent selection and their offspring is tested for replacement of a mediocre chromosome above the median. The parameters that control our implementation are the population size n, mutation rate p_m, maximum number of successful offspring α_{max} and maximum number of failures to improve the best solution β_{max}. It is again needed to find the best parameters combination in order to achieve an algorithm that can run in a reasonable amount of time and produce good results. However, the results for the TSP gave good insights providing two main configurations (small populations + high mutation rate and large populations + low mutation rate) that were kept here.

1. Population Initialisation:

    (a) Generate the first solution (P_1) using the CW heuristic. Use Split to calculate its $F(P1)$. Add to the population.

    (b) Generate the second (P_2) solution using the NN heuristic. Use Split to calculate its $F(P2)$. If $F(P_1)$ is equal to $F(P_2)$ skip. Otherwise, insert it in the population.

    (c) Generate the third (P_3) solution using the RI heuristic. Use Split to calculate its $F(P2)$. If $F(P_1)$ is equal to $F(P_3)$ or $F(P_2)$ is equal to $F(P_3)$ skip. Otherwise, insert it in the population.

    (d) For the rest of the population generate random tours P_k. Use Split to calculate its $F(P_k)$. If $F(P_k)$ is not in the population, insert. Do until the length of the population reaches n. Sort the population in increasing order.

2. Main Loop of the GA:

    (a) Set $\alpha = 0$ and $\beta = 0$.

    (b) While $\alpha < \alpha_{max}$ and $\beta < \beta_{max}$ do:

        i. Select two parents P_1 and P_2 by binary tournament.

        ii. Apply OX and generate C_1 and C_2.

        iii. Select one child randomly C, among (C_1, C_2).

        iv. Select $r = random$. If $r < p_m$ then mutation $C = Mutation(C)$.

        v. Select $k = random.integer(\lfloor n/2 \rfloor, n)$.

        vi. Use Split to calculate its $F(C)$. If $F(C)$ not in the population and $F(C) < F(P_k)$. Insert C in the population at position k. $\alpha = \alpha + 1$. Sort the population.

        vii. If P_1 has not changed. $\beta = \beta + 1$

Figure 4.18: GA-VRP Main Phase

The GA for the VRP does not make use of restarts, this decision was made to reduce running times. A single run of the algorithm is preferred in this case since we are interested in obtaining a good solution in a short period of time. If necessary the algorithm can be restarted using some restarting method to improve the currently found solution. However, our tests in this work do not cover that feature of the algorithm.

4.9.8 Model Evaluation

We test our GA for the VRP in a set of CVRP instances provided by Christofides and Eilon (1969) (Set: E) and Christofides *et al.* (1979) (Set:M). Our time constraints are substituted by load constraints in order to test our implementation in those sets. Those instances represent problems with nodes from 22 to 199. The assumption is that one could find considerably better routes by solving small instances of the VRP instead of a full VRP before deliveries start. This assumption is based both in practical operational reasons mentioned before and to save running time of the algorithms.

Our proposition is that a VRP can be used to solve routing problems for a group of drivers. Given the results obtained, our recommendation is that a group of stops originally belonging to a number of drivers (in our tests, three) can be merged together in a single cost matrix. To select such drivers the delivery companies could make use of their already predetermined correspondence between driver and regions, for example. As one may notice apart from practical issues, nothing stops a delivery company of solving their entire routing problem as a full VRP. Nevertheless, for the purpose of solving a full VRP we advocate the use of other algorithms that are much more efficient in solving much bigger VRP problems.

The method here presented showed interesting results for VRP instances up to 150 nodes. Our aim is not to find optimal routes for all VRP instances, but to show that a good solution (even if suboptimal) VRP can represent more gains in terms of emissions to delivery companies.

4.9.9 GA Parameters

Two different parameters configurations were selected to test the efficiency of the algorithm in finding optimal solution for a CVRP. For the set E (up to 100 nodes), we select a GA with 30 chromosomes in the population, mutation rate: 0.1, number of successful offspring: 300, number of unsuccessful offspring : 3000 (GA-VRP1).

For the set M (between 100 and 200 nodes), the parameters were changed to 50 chromosomes in the population, mutation rate: 0.05, number of successful offspring: 300, number of unsuccessful offspring : 1500 to increase running times (GA-VRP2). The algorithms were applied once for each instance and their results are summarised in the tables 4.3 and 4.4 below.

GA-VRP1 x set: E					
Instance	n	Optimum	GA-VRP1	Gap(%)	RT(s)
E-n22-k4	22	375	375	0.0	6.39
E-n23-k3	23	569	569	0.0	12.5
E-n30-k3	30	534	534	0.0	25.12
E-n33-k4	33	835	835	0.0	28.6
E-n51-k5	51	521	521	0.0	82.78
E-n76-k7	76	682	691	1.3	150.24
E-n76-k8	76	735	750	2.0	200.51
E-n101-k8	101	815	828	1.6	321.8
E-n101-k14	101	1067	1099	3.0	200.46

Table 4.3: GA-VRP Results: set:E Instances

GA-VRP2 x set: M					
Instance	n	Optimum	GA-VRP2	Gap(%)	RT(s)
M-n101-k10	101	820	831	1.3	96.21
M-n121-k7	121	1034	1048	1.4	225.6
M-n151-k12	151	1015	1086	7.0	203.2
M-n200-k16	200	1274	1371	7.6	380.98
M-n200-k17	200	1373	1387	1.0	120.75

Table 4.4: GA-VRP Results: set:M Instances

4.9.10 Results on VRP benchmark instances.

As we can see in the tables above the two GA-VRPs, were evaluated using CVRP benchmark instances with known optima. Both algorithms were applied using a Core i7, 8 GB ram laptop running Windows 7. All code and structures were developed in Python and all the running times represented in the table correspond to real running times on the described machine.

It is worth mentioning that both algorithms were applied only once for each instance. As expected (given the TSP results) GA-VRP1 produces solutions with higher quality and with a higher running time. Whereas GA-VRP2 produces suboptimal tours to prioritize running times instead. When we analyse the gaps found by both algorithms we can see that problems with a higher number of nodes are more difficult to solve and possess higher gaps than smaller instances. For our purpose the average gap found for the set M is reasonable enough to support the application of the method in the real VRP data with emissions objectives.

Chapter 5

Results

5.1 Introduction

Our results show the applications of the EMs using the algorithms developed for both the TSP and VRP cases. We present the results for 9 drivers in the Bristol, UK area and how our solutions are related to the real route taken by the drivers. We also present the routes on a map to give the real feeling of the route changes for each driver. Furthermore, we present the comparison between the solution of n-TSPs and a single VRP to evaluate the possible gains of this approach.

5.2 Solving n-TSPs with Emissions

After assessing the algorithms we chose the configuration GA2 to run the tests with more than 100 nodes and configuration GA1 for 100 nodes or less (both selected as the best of 5 runs). Our model is applied to solve the TSP for each driver using the Road Distance (RD), the Average Speed Emissions Model (SP-EM) and the Speed and Road Gradient Emissions Model (GD-EM) as our objective function/cost matrix. Also, we provide as our base line the real route performed by the drivers (DRIVER). Each driver has all his cost matrices calculated previously and have their TSP solved for each one of the models:

RD, SP-EM and GD-EM. Their results in terms of distance and emissions (for all four models) are presented in the tables below.

Driver: H430 (73 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	167.95	24688.65	37481.27	41529.09
RD	156.94	23071.65	35834.13	39480.69
SP-EM	159.95	23512.65	34853.07	38645.67
GD-EM	159.95	23512.65	34853.07	38645.67

Table 5.1: H430 Results

Driver: M200 (70 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	146.73	21569.31	33469.67	38358.61
RD	131.46	19324.62	30329.93	35108.79
SP-EM	132.07	19414.29	30018.98	34764.0
GD-EM	133.59	19637.73	30516.56	34082.58

Table 5.2: M200 Results

Driver: M2631 (59 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	216.26	31791.69	43373.41	48391.55
RD	123.65	18178.02	28444.8	32573.01
SP-EM	123.85	18207.42	28202.39	32186.98
GD-EM	123.85	18207.42	28202.39	32127.46

Table 5.3: M2631 Results

Driver: M452356 (49 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	92.3	13568.1	25732.13	29591.29
RD	41.23	6059.34	10616.95	12114.22
SP-EM	41.03	6029.94	10539.83	12096.86
GD-EM	41.23	6059.34	10575.49	12072.14

Table 5.4: M452356 Results

Driver: K510 (38 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	101.1	15052.8	24262.3	27702.85
RD	75.63	11132.31	17587.99	19743.61
SP-EM	76.39	10979.43	17315.0	19668.68
GD-EM	76.63	11044.11	17548.34	19562.38

Table 5.5: K510 Results

Driver: N232631 (49 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	52.12	7660.17	12687.11	14308.95
RD	41.57	6109.32	10235.31	11402.58
SP-EM	41.67	6124.02	10216.78	11328.65
GD-EM	41.8	6143.13	10244.64	11251.33

Table 5.6: N232631 Results

Driver: P2652 (90 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	118.75	17457.72	27933.1	31626.27
RD	100.36	14754.39	23029.72	26042.69
SP-EM	101.32	14895.51	22703.09	25713.78
GD-EM	101.36	14901.39	22944.86	25526.95

Table 5.7: P2652 Results

Driver: T520 (94 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	87.58	12877.2	22722.4	26119.08
RD	45.51	6691.44	12871.69	14805.42
SP-EM	45.96	6757.59	12588.2	14328.37
GD-EM	46.17	6788.46	12635.71	14234.94

Table 5.8: T520 Results

Driver: W420 (89 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	71.44	10410.73	20233.29	23421.99
RD	44.27	6462.31	13297.33	15116.62
SP-EM	44.86	6518.17	13195.93	15085.5
GD-EM	45.09	6582.85	13263.69	14954.75

Table 5.9: W420 Results

To provide a method of comparison we also present the percent variance of the solutions found by the models compared to the routes performed by the drivers. Results can be seen below:

Driver: H430 (73 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-6.56	-6.55	-4.39	-4.93
SP-EM	-4.76	-4.76	-7.01	-6.94
GD-EM	-4.76	-4.76	-7.01	-6.94

Table 5.10: H430 Percent Variance

Driver: M200 (70 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-10.41	-10.41	-9.38	-8.47
SP-EM	-9.99	-9.99	-10.31	-9.37
GD-EM	-8.96	-8.96	-8.82	-11.15

Table 5.11: M200 Percent Variance

Driver: M2631 (59 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-42.82	-42.82	-34.42	-32.69
SP-EM	-42.73	-42.73	-34.98	-33.49
GD-EM	-42.73	-42.73	-34.98	-33.61

Table 5.12: M2631 Percent Variance

Driver: M452356 (49 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-55.33	-55.34	-58.74	-59.06
SP-EM	-55.55	-55.56	-59.04	-59.12
GD-EM	-55.33	-55.34	-58.9	-59.2

Table 5.13: M452356 Percent Variance

Driver: K510 (38 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-25.19	-26.04	-27.51	-28.73
SP-EM	-24.44	-27.06	-28.63	-29
GD-EM	-24.2	-26.63	-27.67	-29.38

Table 5.14: K510 Percent Variance

Driver: N232631 (49 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-20.24	-20.25	-19.33	-20.31
SP-EM	-20.05	-20.05	-19.47	-20.83
GD-EM	-19.8	-19.8	-19.25	-21.37

Table 5.15: N232631 Percent Variance

Driver: P2652 (90 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-15.49	-15.49	-17.55	-17.65
SP-EM	-14.68	-14.68	-18.72	-18.69
GD-EM	-14.64	-14.64	-17.86	-19.29

Table 5.16: P2652 Percent Variance

Driver: T520 (94 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-48.04	-48.04	-43.35	-43.32
SP-EM	-47.52	-47.52	-44.6	-45.14
GD-EM	-47.28	-47.28	-44.39	-45.5

Table 5.17: T520 Percent Variance

Driver: W420 (89 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	-	-	-	-
RD	-38.03	-37.93	-34.28	-35.46
SP-EM	-37.21	-37.39	-34.78	-35.59
GD-EM	-36.88	-36.77	-34.45	-36.15

Table 5.18: W420 Percent Variance

As we can see from the results in the tables presented, there are gains in emissions when the TSP is solved for that objective instead of road distances (this effect does not hold for EU20-EM since it is just a scalar of the RD model). Generally speaking, for the SP-EM and GD-EM models, the amount of reduced emissions is proportional to the amount increased road distances, but effects of speed and road gradients play a role in determining how this relationship occurs.

For example, driver P2652 can have his emissions reduced in 18.72% using the SP-EM when compared to the original tour taken by the driver. Using the same driver as example if we compare the solutions of the TSP (RD) and the TSP (SP-EM) in terms of road distance, we can see that the reduction in kilometres is greater when we solve the model for road distances (-15.49%) when compared to road distances solved using the SP-EM (-14.68%). However, when the same analysis is taken using the Emissions calculated by SP-EM, we see that the road distance model can decrease emissions in (-17.55%), whereas the model solved for SP-EM can change emissions in (-18.72%). That means that solving the model for emissions instead of road distance increases the amount of distance travelled but reduces the amount of emissions expelled by the vehicles.

That relationship is due to the fact that this model takes into account the non-linear relationship between emissions and speed (instead of only road distances). The same relationship can be seen for the other drivers present in our analysis and the GD-EM. However, the amount of reductions change according to the distance travelled and speed in the arcs of the final solution.

To compatourre the results obtained with the real-world data, we also present the actual route taken by the driver and the tour found using GD-EM as our objective function calculated using *Google Maps API* in R.

Figure 5.1: H430 Driver Tour

Figure 5.2: H430 Green Tour

Figure 5.3: M200 Driver Tour

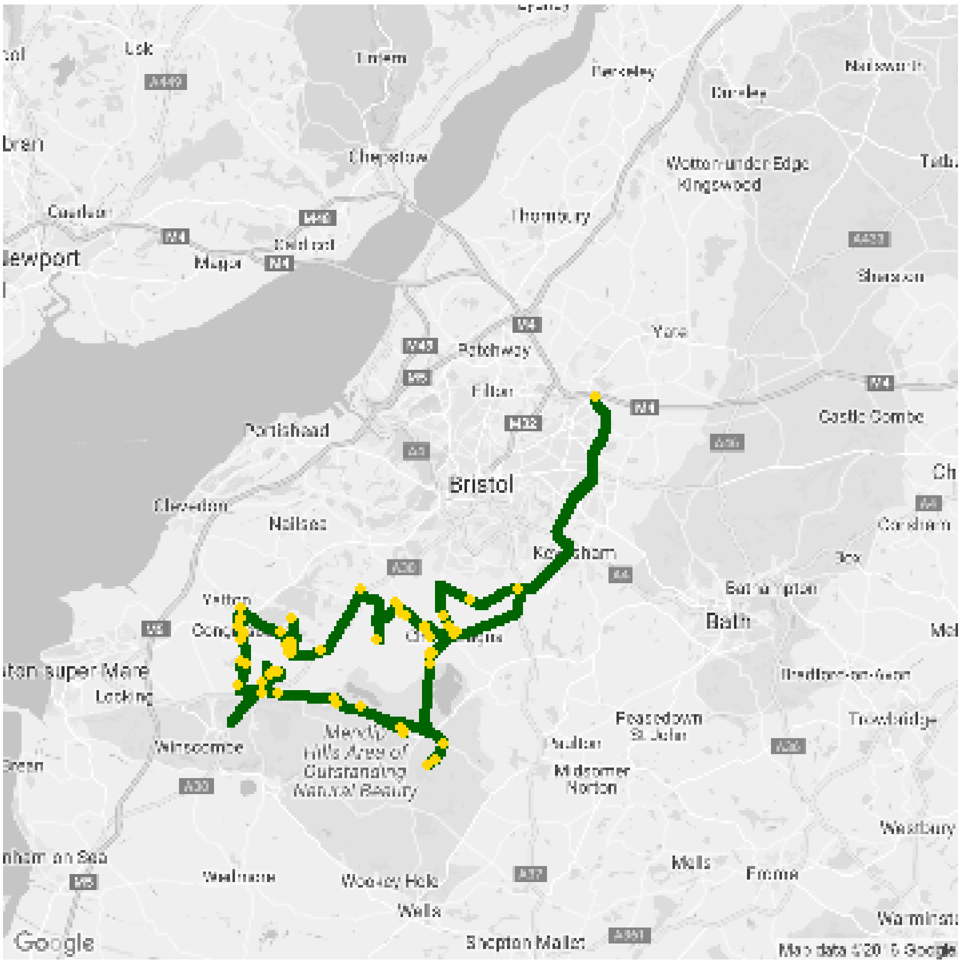

Figure 5.4: M200 Green Tour

Figure 5.5: M2631 Driver Tour

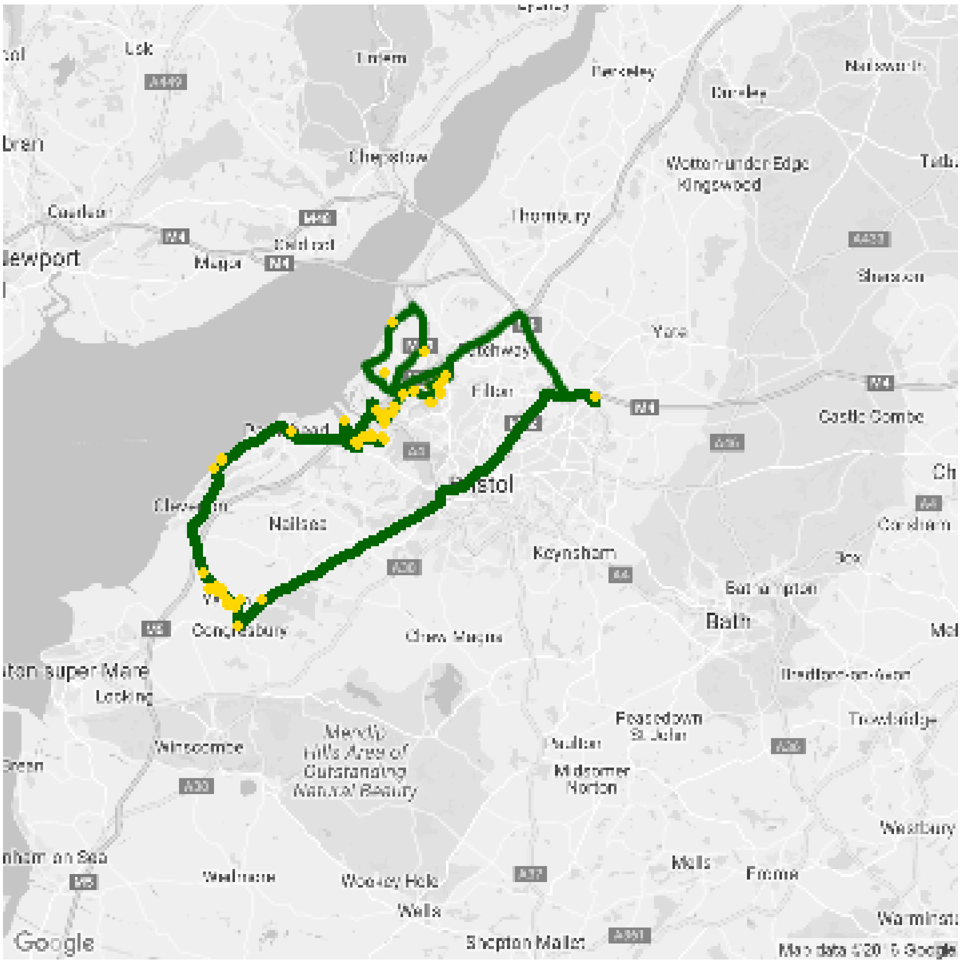

Figure 5.6: M2631 Green Tour

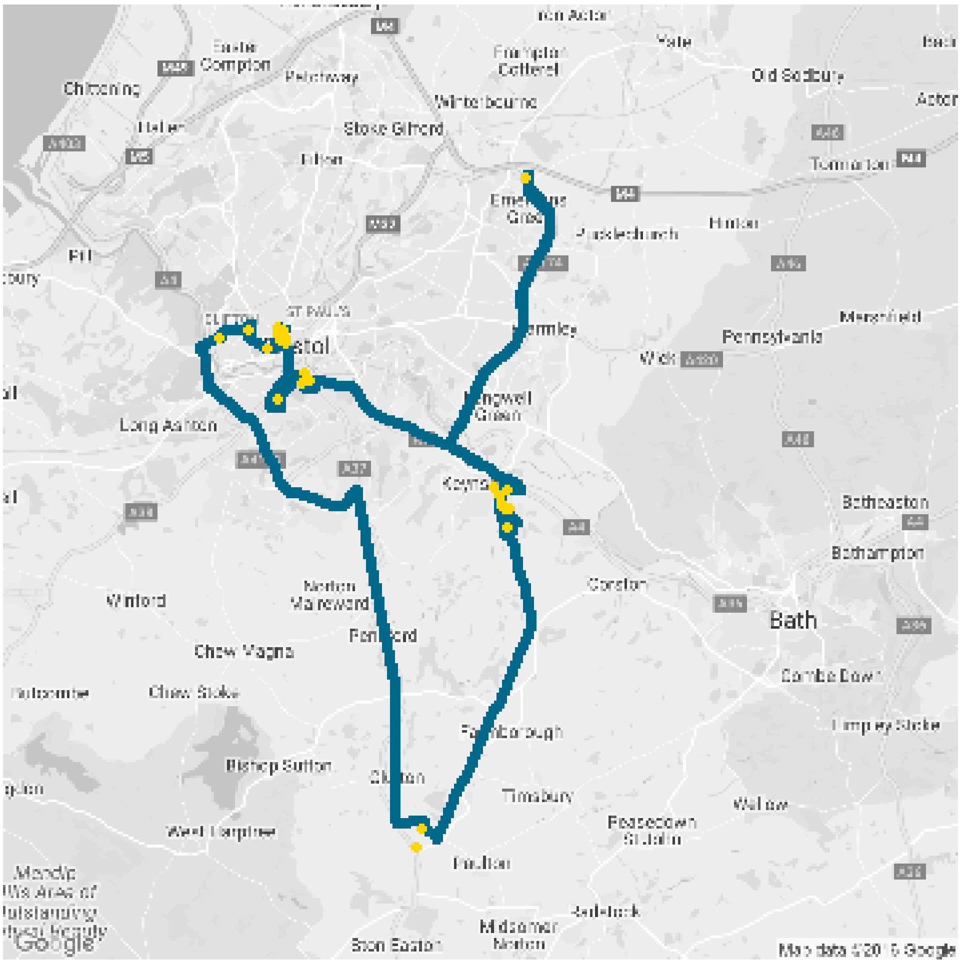

Figure 5.7: K510 Driver Tour

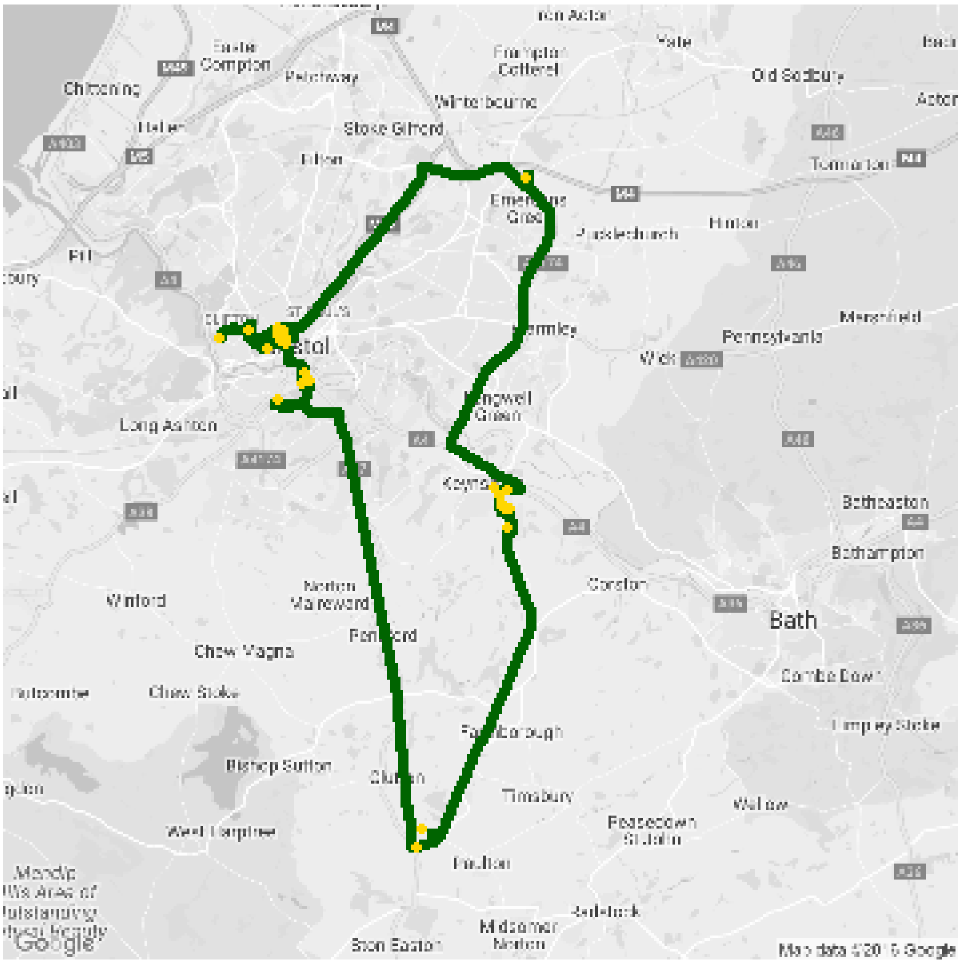

Figure 5.8: K510 Green Tour

Figure 5.9: N232631 Driver Tour

Figure 5.10: N232631 Green Tour

Figure 5.11: P2652 Driver Tour

Figure 5.12: P2652 Green Tour

Figure 5.13: T520 Driver Tour

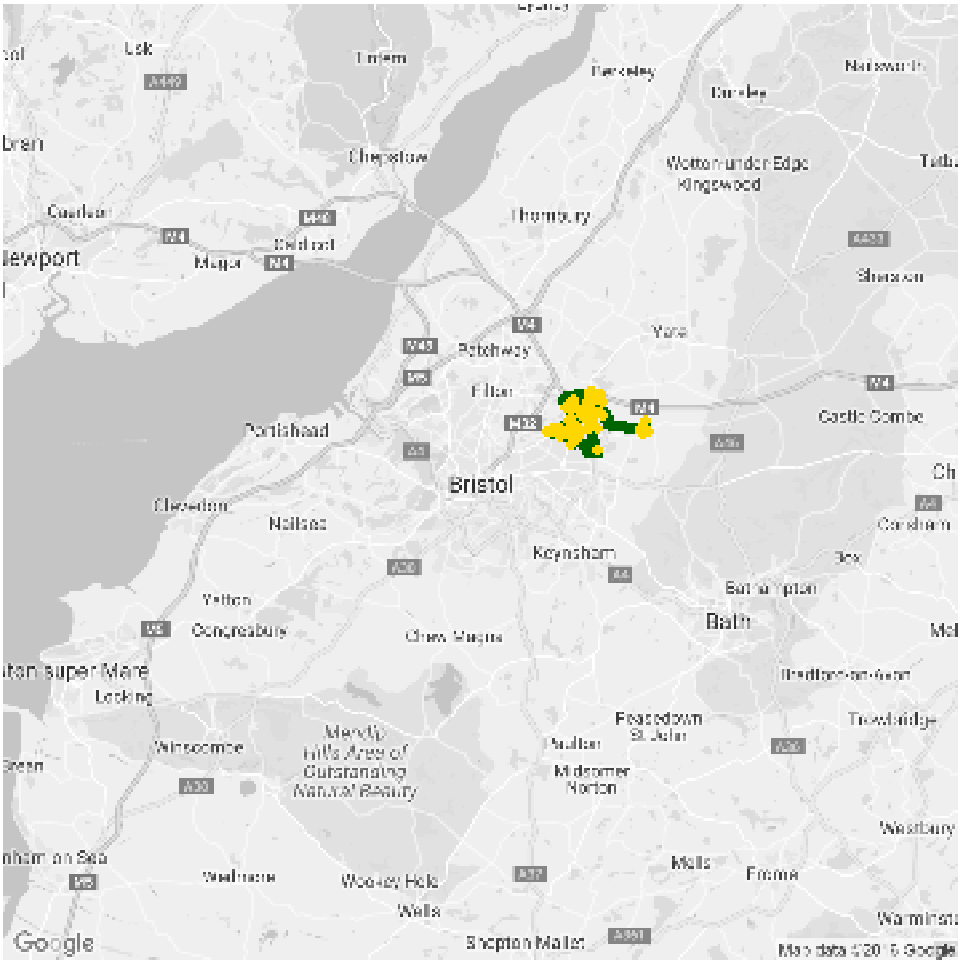

Figure 5.14: T520 Green Tour

Figure 5.15: W420 Driver Tour

Figure 5.16: W420 Green Tour

5.3 G-TSPs vs G-VRP

To solve a VRP model we advocate the usage of the EMs and the GA-VRP (2) for a combination of drivers instead of a full VRP for all stops. As mentioned before, a full VRP solution may not be practical in terms of operations and running times of the algorithm. To asses the difference in the solution of solving n-TSPs or a single VRP we selected 3 drivers (N232631, K510, M452356) and merged all their stops in a single matrix for each EMs. We solved the G-VRP using the GA-VRP2 parameters and present the results regarding road distances and emissions. The resulting VRP solution was compared to the sum of the 3 original TSPs solutions for those drivers. The results are presented in the tables 5.19, 5.20 and 5.21 and figures 5.17, 5.18 below.

3 Drivers TSP (133 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	245.52	36281.07	62681.54	71603.09
RD	158.43	23300.97	38440.25	43260.41
SP-EM	159.09	23133.39	38071.61	43094.19
GD-EM	159.66	23246.58	38368.47	42885.85

Table 5.19: 3-TSPs Results

Green VRP - Solution: 2 Drivers (133 stops)				
Model	RD(Km)	EU20-EM(g)	SP-EM(g)	GD-EM(g)
DRIVER	245.52	36281.07	62681.54	71603.09
RD	130.05	19117.35	32055.16	36609.87
SP-EM	130.75	19220.24	31963.99	36360.07
GD-EM	130.75	19220.24	31963.99	36360.07

Table 5.20: Green VRP Results

3 TSP x Green VRP (133 stops)				
Model	RD(%)	EU20-EM(%)	SP-EM(%)	GD-EM(%)
DRIVER	0	0	0	0
RD	-17.91	-17.95	-16.61	-15.37
SP-EM	-17.81	-16.92	-16.04	-15.63
GD-EM	-18.11	-17.32	-16.69	-15.22

Table 5.21: 3-TSPs x Green VRP Results

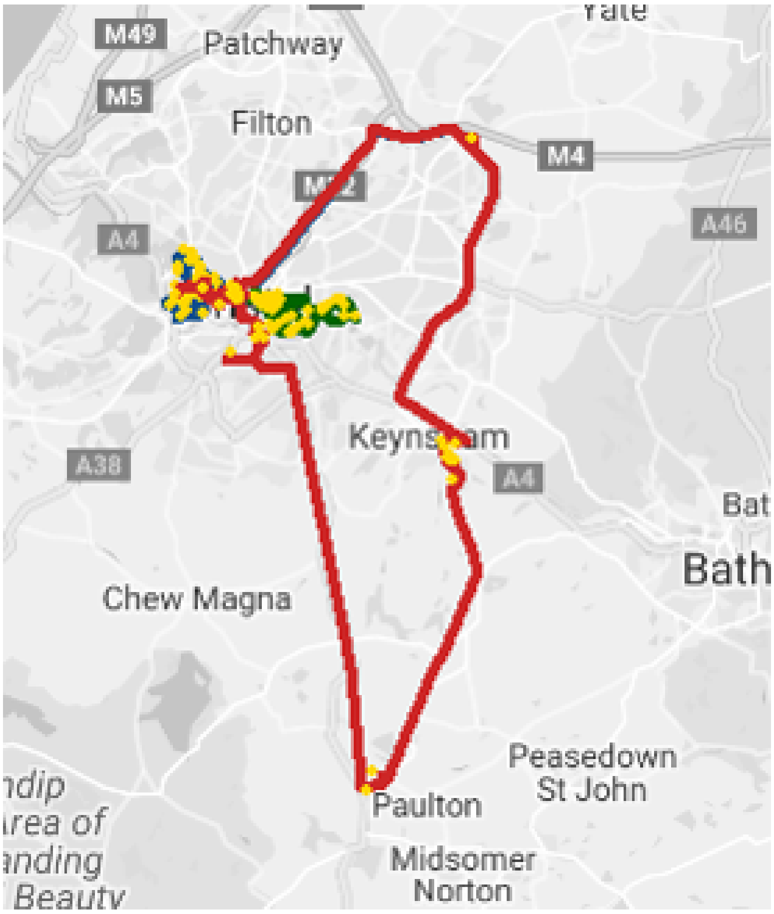

Figure 5.17: 3 TSPs Driver Tour

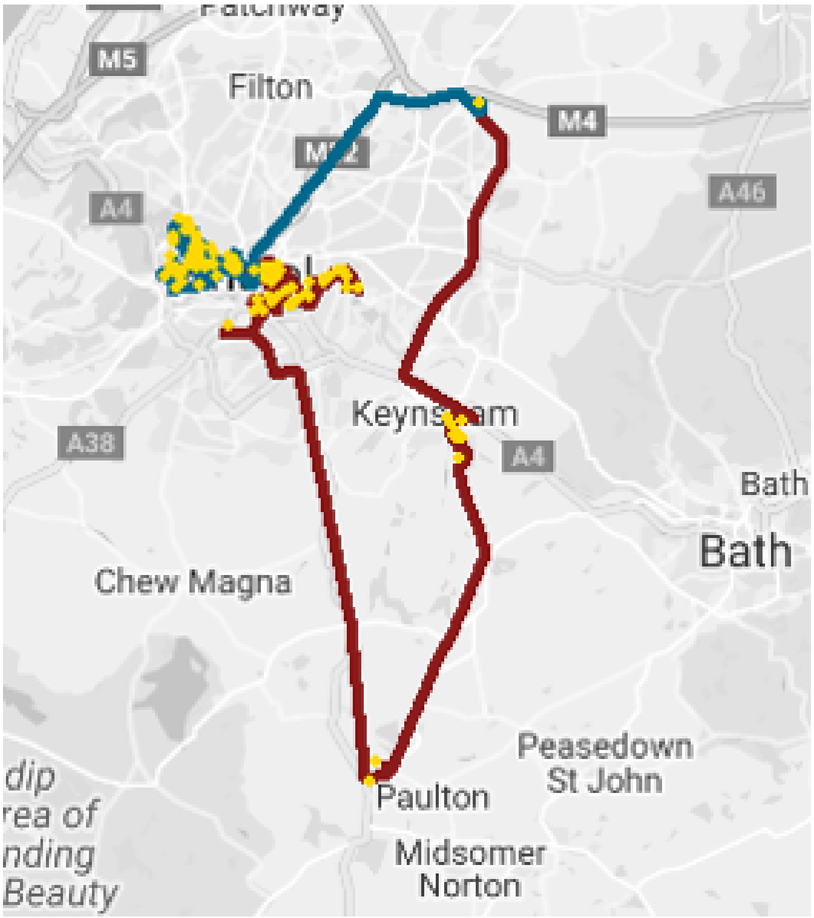

Figure 5.18: Green VRP Tours

As it can be seen from the results and tables above there are further gains in solving a VRP instead of a TSP. For the example shown the VRP solution was able to reduce in (-17.91%) the amount os distance travelled when we use a distance-based objective function. Emissions are also affected one can expect a further reduction of -15.33% even using a possible suboptimal VRP tour. Furthermore, the solution found for the VRP model only used two drivers to visit all the stops in the cost matrix with a limit of 7 hours of work. This is also an indicator that TSP solutions can only give suboptimal tours to a number of drivers and an indicator that redundancy is probably a case in this operation.

Chapter 6

Discussion

6.1 Introduction

In this chapter we examine the main results obtained in this work. We present our point-of-view of the results obtained and provide further research opportunities identified throughout the course of the development of this project.

6.2 Emissions

The lack of empirical and benchmark data does not allow us to test the accuracy of our emissions models. The fact that these models were commissioned by European Union policy makers and are currently used by the member states as tools for official reporting duties is our main guarantee of their reliability and usability.

To summarise the assumptions and the consequences behind the construction of the distance and speed matrices, we present its main outcomes. The smallest allowed distance is 0.1 Km, the smallest allowed travel time is 1 minute, follows from that the minimum allowed speed is equal to 6 Km/h. Thus, according to the SP$-$EM the minimum amount of CO_2 produced is equal to 38.47 g.

To understand what happens when we use the EF derived from the SP−EM as baseline for the GD−EM we need to consider the impact of the Adjusting EF in the Figure 6.1 below.

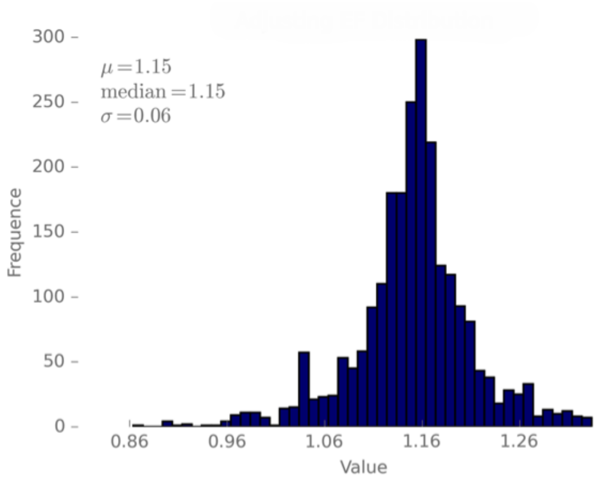

Figure 6.1: Adjusted EF Distribution

Analysing the GD-EM, we have found out that the impact of the Adjusting EF can lighten the average speed dependent EF up to −14%, and this effect makes perfect sense when travelling downhill. However, in the large majority of the cases the Adjusting EF increases the average speed dependent EF, causing on it an average increment of 15%, with peaks around 30%.

To asses the impact of the weight we now look at the distribution of the adjusting EF when the slope is maintained constant and equal to zero. It can be noticed that the average value of the adjusting EF is equal to the previous one found under variable slope conditions. However, the shape distribution is different and almost all the values are concentrated around the mean (1.15). The behaviour of the adjusting EF is mainly driven by the constant k set equal to 1.27.

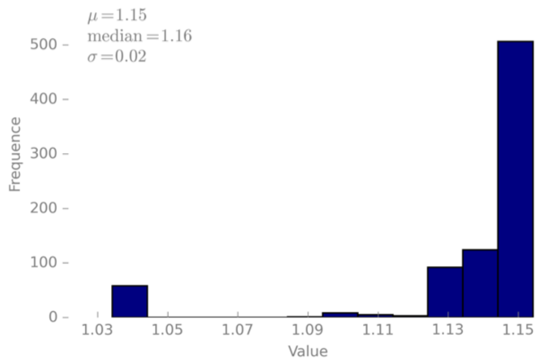

Figure 6.2: Adjusted EF Distribution − Slope = 0

Looking at the two figures (6.3 and 6.4) below it can be noticed that while the shape of the distribution is maintained unchanged the average amount of emissions per arc has raised from 625 g when we are not using the adjusting EF to 725 g when it is included in the model, that is around the 15% more.

The results are related to a single driver randomly selected in a given workday but the relations between EF and the adjusted EF holds on average for all the drivers in the examined time frame.

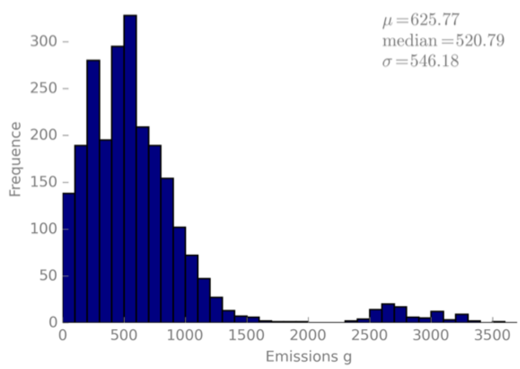

Figure 6.3: Emissions Without Adjusted EF

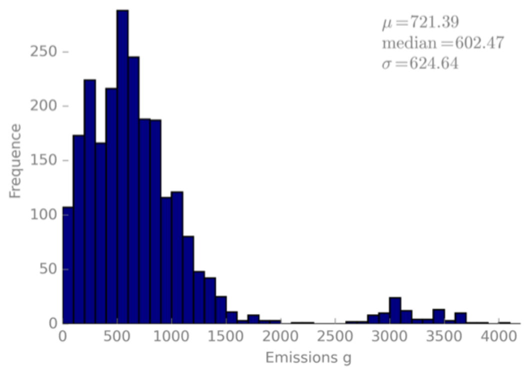

Figure 6.4: Emissions With Adjusted EF

Concluding, an important result is shown, the relation between emissions and speed (Fig. 6.5). The non-linear nature of the model is confirmed. Higher speeds do not imply higher emissions and a variety of emissions values can arise for the same speed level.

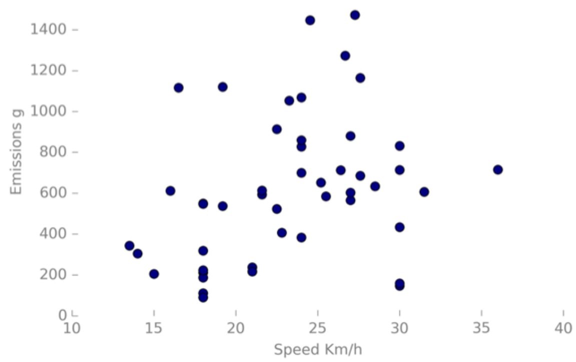

Figure 6.5: Emissions Vs Speed

6.3 Road Network Considerations

The design of an efficient road network is paramount for all modern cities. Smart and sustainable road networks can boost citizen lives, economy and environmental care. We do not have many information about the main features of Bristol road network like: rush hours, number of one way roads, number of traffic lights, number of roundabouts and crossroads.

What we have attempted is a graphical assessments of the arcs of the Bristol road network. Once selected a random driver in a random work day, we have calculated, contrarily to what we do for the algorithm, three asymmetric matrices, for the distances, the slopes and the speeds.

As mentioned, the matrices used as input in the algorithm have to be symmetric, but this does not reflect a fair evaluation of the road network. For instance, given two points, A and B, on a map, the distance between A and B could be different from the distance between B and A. This can happen because in the road network there are some one way roads which do not allow the driver go through the same route while going backwards. As a consequence, this could impact the travel speed. Furthermore, going downhill from A to B implies that we will have to go uphill while travelling from B to A. For example, the MEET report (Hickman *et al.*, 1999) points out that even in the case of large−scale applications, it cannot be assumed that the extra emissions when travelling uphill are completely compensated by the reduced emissions when travelling downhill.

All these considerations directly impact the amount of emissions. As the Figure 6.6 below indicates, given a road network of approximately 1,700 arcs. It can be noticed that the average arc has a length of 2.4 Km and the most part of the distances are lower than 2 Km, while only a minor amount of arcs has a length greater than 12 Km. Long arcs are very often describing the distance between a customer and the depot that is located outside the city area.

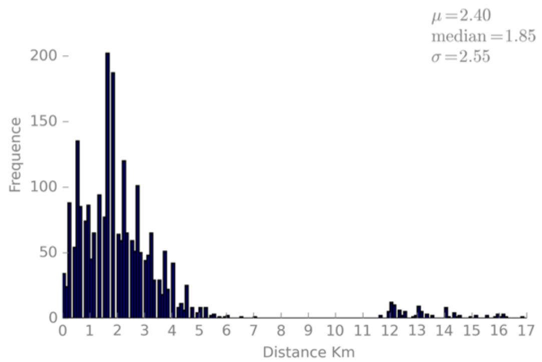

Figure 6.6: Distances Distribution

Looking at the speed distributions (Figure 6.7) we see that the speed range is typical of a city area. The lower bound is endogenously limited to 5 Km/h while the upper bound around 45 Km/h reflects the usual urban speed limit. Unfortunately, according to the model, small average speeds are the ones which cause the higher emissions level.

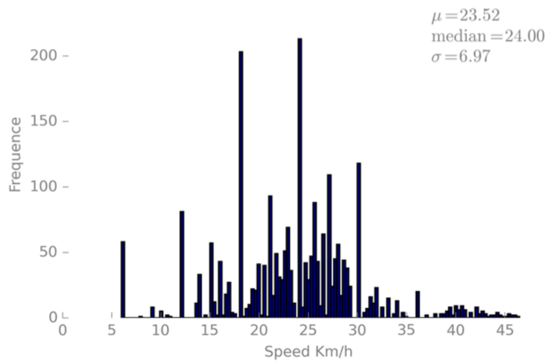

Figure 6.7: Speeds Distribution

The slopes distributions (Figure 6.8) are approximately bell shaped and while some arcs are affected by a positive or negative road gradient, the majority of the cases the slopes are close to zero.

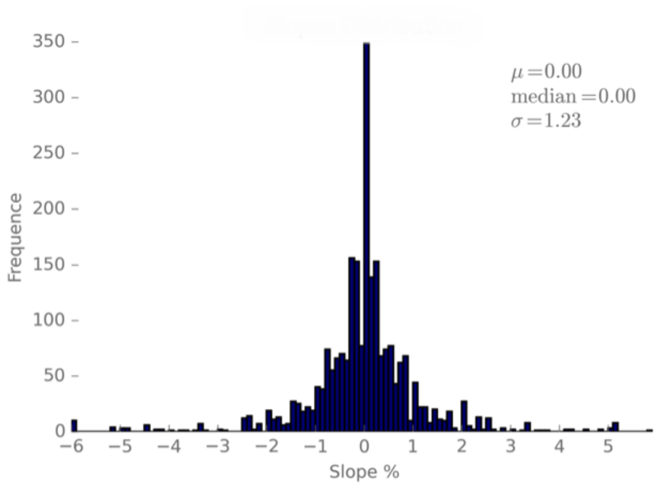

Figure 6.8: Slopes Distribution

To assess the possible difference in the final solution of the symmetry simplification. We selected calculated the travel distance and the total emissions given 100 random routes in two cases: first using asymmetric matrices and second using symmetric matrices. The variation between the two solutions has been calculated using the equation 6.3.

$$Var\,\% = Asymmetric\,Cost - \frac{Symmetric\,Cost}{Asymmetric\,Cost} \tag{6.3.1}$$

Where the cost can either be the distance travelled or the emissions produced for that route. The results after 100 trials are shown below in Figures 6.9 and 6.10. As can be seen, on average, a route cost, either in terms of distance or emissions, in an asymmetric case is less than 1% lower than in the symmetric one. This result supports the symmetry simplification of our models, since the final impact in terms of distance and emissions for our data is considered low.

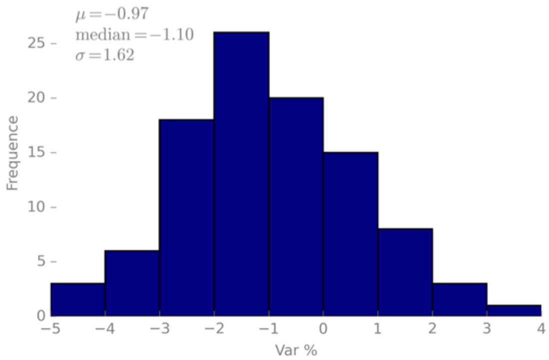

Figure 6.9: Distances Var % : Asymmetric Vs Symmetric

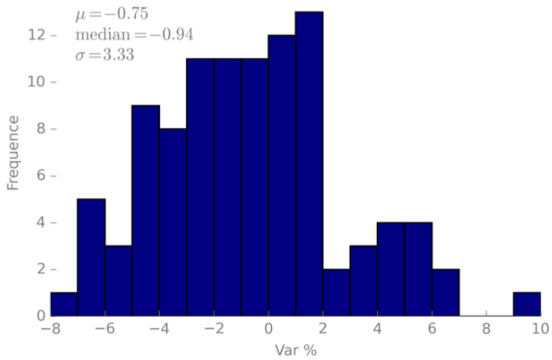

Figure 6.10: Emissions Var % : Asymmetric Vs Symmetric

6.4 Solving n-TSPs

It can be drawn from previous sections that the choice of the EM influences the final outcome of the model in terms of reduced total emission and reduced costs. Thus, the right model choice and its inputs are extremely important to determine the real impact of road emissions in the final model.

% Var. EMs vs RD		
Driver	SP-EM(%)	GD-EM(%)
H430	-2.74	-2.12
M200	-1.03	-2.92
M2631	-0.85	-1.37
M452356	-0.73	-0.35
K510	-1.55	-0.92
N232631	-0.18	-1.33
P2652	-1.42	-1.98
T520	-2.2	-3.85
W420	-0.76	-1.07

Table 6.1: Percent Variance between EMs and Road Solution

As our results show in Table 6.1, routes with slightly longer road distances can result in less emissions (up to almost 4% less emissions compared to road distances) due to the average speed, slope and weight implied in those routes. The results found match those found by Palmer (2007) where he found that minimising CO_2 emissions instead of distance reduces in 5% the amount of total emissions. As seen before, our model takes into account real-world effects in the cost reduction and seems reasonable to infer that these results are not merely occasional but caused by the combination of all the factors considered.

The relationship between reduced emissions and increased road distance does not follow a linear relationship given the non-linear relationship of our EMs. For example, the percent variance of reduced emissions between the optimal distance-based tour and the optimal emissions-based tour varies according to the arcs used, length of the tour and EMs for each of the drivers.

The consistency of the results for all drivers tested support that assuming the correctness of the EMs, it is possible to select sub-optimal routes in terms of distance that generate optimal tours in terms of CO_2 emissions. This results gives answer to the initial question of this dissertation thesis and represents the most important result of this work.

6.5 GVRP instead of TSP

To further improve this analysis we propose the usage of a VRP model as an improvement model to the Green TSP solved for each driver. Since solving a full VRP for each given day may be impractical we propose solving small VRPS, merging for example 3 drivers at each time. Although this solution is still suboptimal (since we do not consider all the nodes in the network), by using this alternative the companies can use the solutions to determine the driver's regions and check for redundancies in the operation.

For that reason we proposed a small modification of the GA proposed to deal with VRPs. The results presented showed that further improvements can be achieved both in terms of road distance and emissions if a VRP is solved. However, this solution has also practical negative implications since reducing the number of drivers (VRP Solution: 2 drivers x 3-TSPs) on a given day also implies in opportunity costs to the delivery companies. Nevertheless, if an estimation of the demands can be done for a period of time. A previous calculation of the VRP could be performed to better define the number of employees needed in the delivery operations.

6.6 Fuel Equivalence

In order to assess the economic impact of emissions reduction we need to understand how CO_2 emissions can be translated in terms of fuel consumption. While the relation between CO_2 and carbon-based fuels is naturally indissoluble and so the more fuel we burn the more is the carbon dioxide we produce, the relation between travelled distance, emissions, and fuel consumption in diesel light duty vehicles is less simple to estimate.

Along with the necessary simplifications the relation between fuel consumption and CO_2 emissions, according to the EPA (2014) can be described in equation 6.6.1.

$$CO_2\, Per\, Litre\, Of\, Diesel\, Burned\, (Kg) \;=\; 2.67 \tag{6.6.1}$$

Regarding the fuel price, at that at the time of this research the price for 1 litre of diesel fuel in Ireland is equal to 1.22 €on average.The outlined equivalences will be used ahead to assess the economic impact of emissions reductions over our optimal route versus an actual one.

To represent the possible economic impact, we can imagine to have a homogeneous fleet of 20 drivers working 5 days a week for 4 weeks a month. Assuming, as per our results, that an average route has a cost around 31 Kg of CO_2 and our algorithm on average allows saving around 30% in term of emissions, the monetary savings generated by lower fuel consumption would be approximately: 1700 €. It is important to notice that these results correspond to a given day in the Bristol, England area and emissions reductions will vary according to the amount of deliveries per day. For that reason, an estimation of the potential reduction for longer periods is only possible with a detailed evaluation of demands per day over a long period of time. It is clear however, that there is a potential for financial cost and emission reduction in our solution.

Concluding, while this amount could appear to be marginally significant from a business point of view, the abstraction of this approach to a large scale level could be of interest for policy makers and have a large impact in terms of emissions reductions.

Chapter 7

Conclusions and Future Research

7.1 Introduction

The significant and most important conclusions of this work are presented in this section. We also present further research opportunities identified in the course of this research project.

7.2 Conclusion

This work has been conducted as fulfilment of the requirements of the UCD MSc in Business Analytics. The research has been carried out with the collaborative assistance of *Xpreso*, which provided data and business insights.

The study focuses on the G-VRP, a modern point of view to a well known and widely addressed optimisation problem, common in the transport and distribution industry. The proposed approach is the modification of the metric used to scale the road network within which we are trying to find the most convenient route. In other words, the route that allow us to visit all the nodes in the network at the minimum cost. While the cost has always been defined as the distance between the nodes, a recent formulation suggested to use the amount of pollutant emissions produced by a vehicle as an innovative metric to redefine the best route.

Using a set of *freeware* tools, the distance travelled, the average speed of the journey, the road gradient and the weight of the load have been combined to model the CO_2 emissions generated by diesel light duty vehicles. A sophisticated genetic algorithm has been designed to guarantee easy of use and high performances either in terms of solution accuracy and computational time. An abundant amount of heuristic approaches have been developed and applied in two different manners. First to form an ideal population of possible solutions and second to improve single solutions with effective strategies. Concluding, despite the limitations of an emission model that is not validated by empirical data, the main findings of this research have raised the interest of the company partner.

Optimal or nearly optimal TSP tours allow significant savings in terms of CO_2 emissions. This saving can be translated in monetary saving through the reduced fuel consumption. A good practical approach is to solve a small VRP made up of three or four TSP because can be shown that further emissions savings can be achieved. Filtering the goodness of the results in relation to adopted emissions model, can be stated that the know-how in terms emissions based route optimisation outlined in this study, is not only interesting in terms of environment care, but also can establish a valuable competitive advantage in the related businesses.

7.3 Future Research

This research has allowed us to work in the emissions route optimisation problem known as the G-VRP. The complexity of the problem, either in its definition or in the variety of approaches used to find an optimal solution open an immense set of research opportunities. Nevertheless, this dissertation thesis aims to provide a piece of academic research where every decision taken was focused both in academic novelty and practical business applications. As presented before, there are paths not completely explored by this work due to several reasons. Therefore, we present a conjunction of possible further improvements for this work that focus both on responding business necessities as well as providing continuous progress in the Operations Research field.

In the context of this study, a necessary improvement can be identified in the benchmarking of the emissions model with real world data. To improve such benchmarks, new emissions models could be proposed to deal with new technologies like hybrid vehicles, alternative fuels, etc. The usage of more up-to-date validated models can lead to better emissions estimations and will support the applications of the best decision when forming the optimal routes.

Moreover, as it has been shown the load weight can have a significant impact on the EF. Gains in emissions can be achieved if a suitable model is applied to a previously calculated (sub-)optimal solution. It would be interesting to investigate how this insight can be exploited to post optimise the final solution, not only in terms of route optimisation but also from an operational point of view. Also, other effects can be used directly in the EFs modelling. Effects like weather, traffic jams, real-time optimisation of routes can drastically change the selection of the best tour to be completed.

Lastly, the development of an algorithm able to deal with asymmetric matrices could increase the realism of the optimisation process. To increase even more the realism of the optimisation problem the business might consider soft boundaries and allow overlap of drivers' regions (instead of solving n TSPs). This would be a much more challenging optimisation problem to solve and would require more constraints and probably more computational time.

Appendix - GA Implementation TSP

We present a snap of the core part of the Genetic Algorithm Implemented in *Python*.

```
success = 0
no_improve = 0
while (success < number_success and no_improve < number_unsuccess):
    current_best = population[0]
    parent_1_candidate_1 = SelectParent(population)
    parent_1_candidate_2 = SelectParent(population)


    if parent_1_candidate_1[2] < parent_1_candidate_2[2]:
        parent_1 = parent_1_candidate_1[1]

    else:
        parent_1 = parent_1_candidate_2[1]

    parent_2_candidate_1 = SelectParent(population)
    parent_2_candidate_2 = SelectParent(population)


    if parent_2_candidate_1[2] < parent_2_candidate_2[2]:
        parent_2 = parent_2_candidate_1[1]
    else:
        parent_2 = parent_2_candidate_2[1]



    child_1,child_2 = OXCrossover(parent_1,parent_2)

    which_child = random.choice([1,2])

    if which_child == 1 :
        child = list(child_1)
    else:
        child = list(child_2)
```

Figure .1: GA Main Phase Part 1 - Python

```
    which_child = random.choice([1,2])

    if which_child == 1 :
        child = list(child_1)
    else:
        child = list(child_2)

    rand = random.uniform(0, 1)

    child_cost = CostCalculation(child,cost_matrix)
    if  rand < mutation_rate:
        child,child_cost = Mutation(child,child_cost,cost_matrix)

    child_fitness = Fitness(child_cost)

    k = random.randint(math.floor(population_size/2.0),population_size-1)

    add = 1
    if BiContains(population,child_cost):
        add = 0

    if add == 1 and child_cost < population[k][2] :
        population[k] =  (child_fitness,child,child_cost)
        success = success + 1
        population.sort()

    if population[0] == current_best:
        no_improve += 1

return population
```

Figure .2: GA Main Phase Part 2 - Python

Glossary

.1 Glossary

arc line which along with two nodes constitutes fundamental units of a graph

branch-and-bound mathematical model used to solve routing problems

chromosome biological structure

crossover biological chromosome process to pass information to offspring

edge line which along with two nodes constitutes fundamental units of a graph

heuristics mathematical model used to solve routing problems

google IT company

google maps online tool for visualising maps

graph mathematical structures used to model pairwise relations between objects

k-d trees computer structure

metaheuristics mathematical model used to solve routing problems

mutation biological process that changes genes structure

node point which constitutes fundamental unit of a graph

python programming language

R statistical package

slope road inclination

xpreso IT Partner company

.2 Acronyms

CVRP Capacitated Vehicle Routing Problem

CW Clarke and Wright heuristic

DVRP Dynamic Vehicle Routing Problem

EF Emission Factor

EM Emission Model

ETA Expected Time of Arrival

EU20-EM Euro 2020 Emission Model

GD-EM Average Speed Gradient Weight Emission Model

GA Genetic Algorithm

GA1 Genetic Algorithm for the TSP - 1

GA2 Genetic Algorithm for the TSP - 2

GA-VRP1 Genetic Algorithm for the VRP - 1

GA-VRP2 Genetic Algorithm for the VRP - 2

G-TSP Green Travel Salesman Problem

G-VRP Green Vehicle Routing Problem

HTTP Hypertext Transfer Protocol

Km Kilometres

Km/h Kilometres per hour

LKH Lin-Kernighan Heuristic

LKH Lin-Kernighan-Helsgaun Heuristic

NN Nearest neighbour heuristic

OR Operation Research

OVRP Open Vehicle Routing Problem

OX Ordered Crossover

PDVRP Pickup and Delivery Vehicle Routing Problem

RI Random Insertion heuristic

SD-EM Average Speed Emission Model

SVRP Stochastic Vehicle Routing Problem

TDVRP Time-dependent Vehicle Routing Problem

URL Uniform Resource Locator

TSP Travel Salesman Problem

VRP Vehicle Routing Problem

VRPTW Vehicle Routing Problem with Time Windows

Bibliography

Baker, B. M. and M. A. Ayechew. 2003. A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.*, **30**(5): 787–800. ISSN 0305-0548.

Baldacci, R., N. Christofides and A. Mingozzi. 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, **115**(2): 351–385.

Baldacci, R., A. Mingozzi and R. Roberti. 2012. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, **218**(1): 1–6.

Balinski, M. L. and R. E. Quandt. 1964. On an integer program for a delivery problem. *Operations Research*, **12**(2): 300–304.

Baškovič, K. and M. Knez. 2011. A review of vehicular emission models.

Blazinskas, A. and A. Misevicius. 2011. Combining 2-opt, 3-opt and 4-opt with k-swap-kick perturbations for the traveling salesman problem. *Information Technologies' 2011 : Proceedings of the 17th International Conference on Information and Software Technologies*, Seiten 27–29.

Boulter, P., I. S. McCrae and T. J. Barlow. 2007. *A review of instantaneous emission models for road vehicles.* TRL Limited.

Cairns, S. 1999. Home delivery of shopping: the environmental consequences. *TSU ref*, **5**.

Christofides, N. and S. Eilon. 1969. An algorithm for the vehicle-dispatching problem. *Or*, Seiten 309–318.

Christofides, N., A. Mingozzi and P. Toth. 1979. The vehicle routing problem. in: Christofides, n., mingozzi, a., toth, p., sandi, c. (eds.), combinatorial optimization. *Vol. 1. Wiley Interscience*, Seite 315338.

——. 1981a. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, **20**(1): 255–282.

——. 1981b. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, **11**(2): 145–164.

Clarke, G. and J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, **12**(4): 568–581.

Cooke, K. L. and E. Halsey. 1966. The shortest route through a network with time-dependent internodal transit times. *Journal of mathematical analysis and applications*, **14**(3): 493–498.

Council of European Union. 1993. Council regulation (EU) no 59/1993.

Dantzig, G. B. and J. H. Ramser. 1959. The truck dispatching problem. *Management Science*, **6**(1): 80–91.

Demir, E. 2012. *Models and algorithms for the pollution-routing problem and its variations*. Dissertation, University of Southampton.

Demir, E., T. Bektaχs and G. Laporte. 2011. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, **16**(5): 347–357.

Donati, A. V., R. Montemanni, N. Casagrande, A. E. Rizzoli and L. M. Gambardella. 2008. Time dependent vehicle routing problem with a multi ant colony system. *European journal of operational research*, **185**(3): 1174–1191.

Eilon, S., C. Watson-Gandy and N. Christofides. 1971. *Distribution management*. Griffin London.

EPA, U. 2014. Greenhouse gas emissions from a typical passenger vehicle. *Technischer Bericht*, US Environmental Protection Agency. Office of Transportation and Air Quality, EPA-420-F-14-040a.

Erdougan, S. and E. Miller-Hooks. 2012. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, **48**(1): 100–114.

Esteves-Booth, A., T. Muneer, J. Kubie and H. Kirby. 2002. A review of vehicular emission models and driving cycles. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, **216**(8): 777–797.

Fisher, M. L. 1994. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, **42**(4): 626–642.

Fisher, M. L. and R. Jaikumar. 1981. A generalized assignment heuristic for vehicle routing. *Networks*, **11**(2): 109–124.

Flood, M. M. 1956. The traveling-salesman problem. *Operations Research*, **4**(1): 61–75.

G. Clarke, J. W. W. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**(4): 568–581. ISSN 0030364X, 15265463.

Gendreau, M., G. Laporte and R. Séguin. 1995. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation science*, **29**(2): 143–155.

Golden, B. L., T. L. Magnanti and H. Q. Nguyen. 1977. Implementing vehicle routing algorithms. *Networks*, **7**(2): 113–148.

Groër, C., B. Golden and E. Wasil. 2011. A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing*, **23**(2): 315–330.

Haghani, A. and S. Jung. 2005. A dynamic vehicle routing problem with time-dependent travel times. *Computers & operations research*, **32**(11): 2959–2986.

Helsgaun, K. 2000. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, **126**(1): 106–130.

Hickman, J., D. Hassel, R. Joumard, Z. Samaras and S. Sorenson. 1999. Meet-methodology for calculating transport emissions and energy consumption. european commission, dg vii. *Technischer Bericht*, ISBN 92-828-6785-4, Luxembourg, 362 p. www. inrets. fr/infos/cost319.

Ichoua, S., M. Gendreau and J.-Y. Potvin. 2003. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, **144**(2): 379–396.

J.M. Bloemhof-Ruwaard, L. H., P. van Beek and L. V. Wassenhove. 1995. Interactions between operational research and environmental management. *European Journal of Operational Research*, **85**: 229,243.

Johnson, D. S. and L. A. McGeoch. 1997. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, **1**: 215–310.

Kamil Rocki, R. S. 2012. Accelerating 2-opt and 3-opt local search using gpu in the travelling salesman problem. *Transportation Research Part E: Logistics and Transportation Review*.

Kara, I., B. Y. Kara and M. K. Yetis. 2007. Energy minimizing vehicle routing problem. In: *Combinatorial optimization and applications*, Seiten 62–71. Springer.

Lacomme, P., C. Prins and W. Ramdane-Chérif. 2001. *A Genetic Algorithm for the Capacitated Arc Routing Problem and Its Extensions*, Seiten 473–483. Springer Berlin Heidelberg, Berlin, Heidelberg.

Laporte, G. 1992. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**(3): 345–358.

Laporte, G., F. Louveaux and H. Mercure. 1992. The vehicle routing problem with stochastic travel times. *Transportation science*, **26**(3): 161–170.

Laporte, G., S. Ropke and T. Vidal. 2014. Heuristics for the vehicle routing problem. *Vehicle Routing: Problems, Methods, and Applications*, **18**: 87.

Li, F., B. Golden and E. Wasil. 2007. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & operations research*, **34**(10): 2918–2930.

Lin, C., K. L. Choy, G. T. Ho, S. Chung and H. Lam. 2014. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, **41**(4): 1118–1138.

Lin, S. and B. W. Kernighan. 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, **21**(2): 498–516.

Little, J. D., K. G. Murty, D. W. Sweeney and C. Karel. 1963. An algorithm for the traveling salesman problem. *Operations research*, **11**(6): 972–989.

Lyon, T. P. and J. W. Maxwell. 2008. Corporate social responsibility and the environment: A theoretical perspective. *Review of environmental economics and policy*, **2**(2): 240–260.

Maibach, M., C. Schreyer, D. Sutter, H. Van Essen, B. Boon, R. Smokers, A. Schroten, C. Doll, B. Pawlowska and M. Bak. 2007. *Handbook on estimation of external costs in the transport sector*. CE Delft Delft.

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

Miller, D. L. 1995. A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, **7**(1): 1–9.

Min, H., V. Jayaraman and R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, **108**(1): 1–15.

Nagata, Y. and O. Bräysy. 2009. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, **54**(4): 205.

Nagy, G. and S. Salhi. 2007. Location-routing: Issues, models and methods. *European Journal of Operational Research*, **177**(2): 649–672.

Nilsson, C. 2003. Heuristics for the traveling salesman problem.

Ntziachristos, L., Z. Samaras, S. Eggleston, N. Gorissen, D. Hassel, A. Hickman *et al.*. 2000. Copert iii. *Computer Programme to calculate emissions from road transport, methodology and emission factors (version 2.1), European Energy Agency (EEA), Copenhagen.*

Ou, W. and B.-G. Sun, 2010. A dynamic programming algorithm for vehicle routing problems. In: *Computational and Information Sciences (ICCIS), 2010 International Conference on*, Seiten 733–736. IEEE.

Palmer, A. 2007. The development of an integrated routing and carbon dioxide emissions model for goods vehicles.

Parragh, S. N., K. F. Doerner and R. F. Hartl. 2008. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, **58**(1): 21–51.

Potvin, J.-Y. 1996. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, **63**(3): 337–370. ISSN 1572-9338.

Prins, C. 2001. A simple and effective evolutionary algorithm for the vehicle routing problem. *COMPUTERS AND OPERATIONS RESEARCH*, **31**: 2004.

Psaraftis, H. N. 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, **14**(2): 130–154.

Sariklis, D. and S. Powell. 2000. A heuristic method for the open vehicle routing problem. *Journal of the Operational Research Society*, Seiten 564–573.

Sbihi, A. and R. W. Eglese. 2007. Combinatorial optimization and green logistics. *4OR*, **5**(2): 99–116.

Soler, D. 2016. Lkh results for soler's atsp instances. (Accessed on 02/08/2016).
URL `"http://www.akira.ruc.dk/~keld/research/LKH/Soler_results.html"`

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

Subramanian, A., E. Uchoa and L. S. Ochi. 2013. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, **40**(10): 2519–2531.

Toregas, C. and C. ReVelle. 1972. Optimal location under time or distance constraints. *Papers in Regional Science*, **28**(1): 133–144.

Toth, P. and D. Vigo, 2001. An overview of vehicle routing problems. In: *The vehicle routing problem*, Seiten 1–26. Society for Industrial and Applied Mathematics.

——. 2002. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, **123**(1): 487–512.

——. 2014. *Vehicle routing: problems, methods, and applications*, Band 18. Siam.

Vidal, T., T. G. Crainic, M. Gendreau, N. Lahrichi and W. Rei. 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, **60**(3): 611–624.

Vidal, T., T. G. Crainic, M. Gendreau and C. Prins. 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, **40**(1): 475–489.

Wang, H. and I. McGlinchy, 2009. Review of vehicle emission modelling and the issues for new zealand. In: *Proceedings of the 32nd Australasian Transport Research Forum*.

Woensel, T., R. Creten and N. Vandaele. 2001. Managing the environmental externalities of traffic logistics: The issue of emissions. *Production and Operations Management*, **10**(2): 207–223.

Zachariadis, E. E. and C. T. Kiranoudis. 2010. A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research*, **37**(12): 2089–2105.