Improving Driver Satisfaction

Exploring cost effects of optimization on workload preference and region consistency in a VRPTW

Q.W.J. van Gulik

Graduation thesis
MSc Applied Mathematics, TU Delft

Delft University of Technology

ORTEC

Improving Driver Satisfaction

Exploring cost effects of optimization on workload preference and region consistency in a VRPTW

by

Q.W.J. van Gulik

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday January 27, 2025 at 10:00 AM.

Student number:	4957075	
Project duration:	March, 2024 – January, 2025	
Thesis committee:	Prof. dr. D. Gijswijt,	TU Delft, supervisor
	Dr. A. Heinlein,	TU Delft
	Dr. L. Berghman,	ORTEC, supervisor
	Drs. T. Bruinink,	ORTEC

TUDelft

Preface

This thesis has been written as part of the MSc programme Applied Mathematics at TU Delft. It has been written from March 2024 to January 2025 at ORTEC in Zoetermeer, under the supervision of Lotte Berghman at ORTEC and Dion Gijswijt at TU Delft.

First and foremost I would like to thank ORTEC as a company for giving me the opportunity to do my graduation project with them. The atmosphere at ORTEC with their knowledgeable and passionate employees has inspired me. I am grateful for the opportunity to have been working among and together with these bright minds. I would like to thank Lotte Berghman and Tom Bruinink for their daily supervision at ORTEC. The weekly meetings kept me on my toes and were of great help throughout the project.

I would like to thank all members of the thesis committee and in particular Dion Gijswijt for the biweekly meetings we have had throughout the project. I especially liked our sometimes technical discussions, and appreciate your sincere interest in the thesis topic. Thank you for your time, guidance and confidence you have given me.

I would like to extend my thanks to all of the other interns at ORTEC during my time there. All of you made the time at ORTEC so much more vibrant. Thank you Oscar for always being positive and for your trust. Thank you Hasan for all your talks about LLMs. Thank you all other interns.

Lastly, I would like to thank Joop van der Heijden. As an employee at ORTEC he has helped me to comprehend and work with the PyVRP software. Without him, the code implementation would have been much more difficult for me. I am thankful for his attentiveness, his accompaniment, his coding tips and his overall support.

Q.W.J. van Gulik
Schiedam, January, 2025

Abstract

This research aims to optimize a VRPTW that incorporates the driver satisfaction factors 'region consistency' and 'workload preference' while not increasing routing costs too much. The developed measures were optimized for using the 'Random Allocation', 'Driver Assignment' and 'Integrated Approach' methods for various weightings in a multi-objective setting. Driver satisfaction was explicitly optimized in the state-of-the-art VRPTW solver called PyVRP developed by ORTEC. The integrated approach outperformed both the driver assignment and random allocation methods on small and medium sized instances, whereas Driver Assignment performed best on large instances.

Contents

Chapter 1

Introduction

The International Road Transport Union (IRU) projected that global shoartage in truck drivers will rise from 3 million in 2023 to 7 million 2028 [15]. In the United States alone, every year more than 100,000 new drivers will need to be hired by the trucking industry [9]. More than half of these hires will be needed to replace retiring truck drivers and 25% will be needed to keep up with industry growth [9]. Besides a lack of truck drivers, high turnover rates also reflect a stronger demand for drivers [9]. As truck driver shortages skyrocket and turnover rates are fierce, the necessity for companies to hold on to their truck drivers is as crucial as ever.

There are different kind of businesses that have delivering customers by using trucks as their core-business. These businesses involve but are not limited to business to business delivery (B2B), parcel delivery and home delivery. This thesis focuses specifically on the home delivery application. One of the most prevalent instances of home delivery is e-grocery, which is the delivery of supermarket items to consumers.

Home delivery distinguishes itself from business to business and parcel delivery. The amount of customers on a given shift seen by a driver working in home delivery is typically between the amount of customers on a given shift for business to business and parcel delivery. In home delivery drivers typically spend a couple of minutes at each customer.

An important point of notice is the following: For the home delivery subbranch, which is a type of less-than-truckload (LTL), the turnover rates (+-18%, year 2019) are much lower than for truckload (TL), which were over 70% in 2019. [9]. This is due to higher pay and more time at home in the LTL sector than in the TL sector. Still, both sectors heavily involve drivers and with driver shortage rising, the LTL sector can also benefit from retaining their employees and attracting new ones. ORTEC aims to incorporate driver satisfaction in their supply chain software to continuously improve and to be innovative.

As demand for drivers increases, trucking companies try to recruit drivers from other carriers [9]. They do this by offering sign-on bonuses, higher pay, newer trucks, and better routes [9]. Part of what makes routes attractive to drivers is consistency. Routes can differ from day to day in service area, imposing a high learning burden on the drivers [30]. This also reduces the service quality for the customers [30]. Consistency in service area reduces the need to learn about new routes and new customers. This way, consistency not only improves employee satisfaction, but also their productivity as they gain experience in the regions they drive in [30]. In addition to better driver satisfaction and productivity, driver satisfaction increases the perceived value of customers, increasing customer satisfaction and therefore their loyalty, resulting in more profitability [17]. All benefits together, larger routing costs due to a focus on consistency are even more than made up for [30].

Many studies consider region consistency purely from an efficiency point of view. Moreover, some studies balance workload, but this is also mostly done from an efficiency point of view. Over the last few years, fairness and equity have been gaining recognition in vehicle routing [39, 46]. In our study, drivers are put central and driver satisfaction is increased.

In this thesis, we incorporate driver satisfaction in a VRPTW. Drivers are central in this study. We consider preferences of drivers related to workload and regions and model experience of drivers in served regions. We aim to optimize a VRPTW that incorporates the driver satisfaction factors ‘region consistency’ and ‘workload preference’ while not increasing routing costs too much. We develop measures for workload preference adherence and region consistency incorporating region preferences and experiences. These measures are optimized for together with conventional routing costs in a multi-objective optimization problem, introduced as the DSVRPTW.

Optimization is done by assigning drivers to routes made on conventional routing costs and using the introduced DSVRPTW in an integrated approach. All methods make use of routing software called PyVRP, which is developed by ORTEC and makes use of a hybrid genetic algorithm. PyVRP is altered to suit our use-case.

What sets this research apart from other research is the combination of region consistency and workload preferences. Drivers are placed centrally in this study. We consider experience of drivers not in the context of efficiency, but in the context of driver satisfaction. Heterogeneous drivers are considered that each have preferred workloads that are adhered to. Deviations from ideal scenarios are spread reasonably well over all drivers to account for fairness. Our research uses real-life instance data of a customer of ORTEC. The customer operates in home delivery. Home delivery is characterized by relatively short distances and by having many clients to serve.

This thesis will be structured in the following way. Chapter 2 will provide theoretical background regarding the VRPTW, genetic algorithms and multi-objective optimization. Secondly, Chapter 3 goes over relevant literature and places our research in context. Chapter 4 describes our problem and gives high level modelling assumptions and their motivation. Using the problem description a model is designed to capture the driver satisfaction factors *Workload preference* and *Region consistency*. Methods to solve our problem are given in Chapter 6. Specifically, the PyVRP software is described and three solution approaches are described: *Random allocation*, *Driver assignment* and *Integrated approach*. In Chapter 7, our specific case is described and analyzed in more detail. Results are presented and analyzed in Chapter 8. Lastly, the conclusions and recommendations are given in Chapter 9 and Chapter 10.

This thesis builds upon the thesis on driver happiness by Bruinink [5].

Chapter 2

Background

This chapter presents background information that will serve as a basis to understanding this thesis and that will be used in later chapters. In Section 2.1 the classical vehicle routing problem (VRP) and vehicle routing problem with time windows (VRPTW) will be described. Section 2.2 describes the philosophy of genetic algorithms and gives a scheme of the working of hybrid genetic search. Lastly, section 2.3 goes over the essential theory on multi-objective optimization.

2.1 The Vehicle Routing Problem

This section will describe the basic vehicle routing problem (VRP) and the VRP variant that forms the basis of this research: the vehicle routing problem with time windows (VRPTW). Notation is used from the software available at ORTEC, called PyVRP, and its documentation [44] in order to be consistent throughout this thesis.

2.1.1 The Classical Vehicle Routing Problem (VRP)

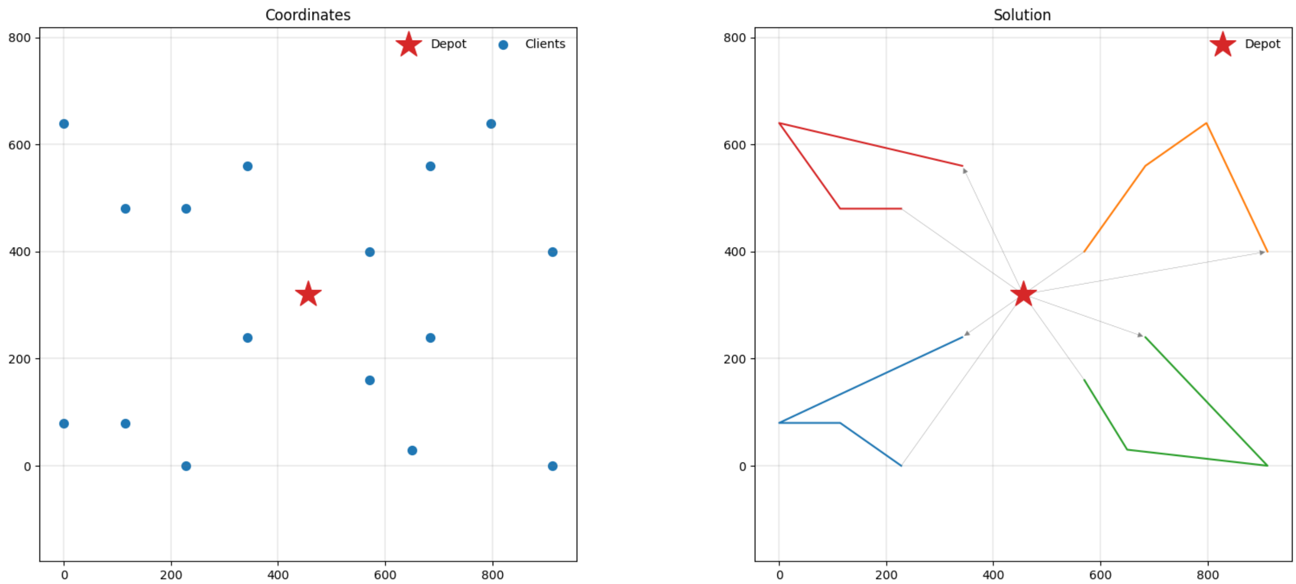
The vehicle routing problem describes the problem of how to create efficient routes in order to serve customers. Efficiency commonly translates to driving the least amount of distance or time, while serving all customers present in the problem. Other objectives are also used such as minimizing the amount of vehicles used, driving such that the environmental impact is minimized, among (many) others.
Although vehicles are the most important use-case of VRPs, they can be possibly used in any scenario in which items have to travel to destinations. An example of such a scenario is planning internet traffic through servers. In this thesis, a vehicle routing problem will be used in the context of vehicles delivering parcels to customers.

To build up to the VRP formulation, we will first introduce some notation. We consider a complete directed graph $G = (V, A)$. The vertex set V is partitioned into $V = \{0\} \cup V_c$. Here 0 represents the depot which is the starting and final location for any given route. The set of n customers to be served is denoted by $V_c = \{1, 2, \dots, n\}$.
Each arc $(i, j) \in A$ is assigned a weight $d_{i,j} \geq 0$. This weight can represent either distance, time or any other measure. Any route that uses this arc is attributed this weight as a form of costs. A fleet K is a set of vehicles k that are available in the vehicle routing problem. In the VRP, every vehicle is driven by exactly one driver and every driver drives exactly one route.
A route is defined to be a path in G that starts and ends at the depot. A path is a walk that visits each node in that walk at most once. A route is denoted as $r = (i_1, i_2, \dots, i_s)$, where $S = \{i_1, i_2, \dots, i_s\} \subseteq V_c$. The fact that a route begins and ends at the depot is assumed implicitly, that is, $i_0 = i_{s+1} = 0$. The set of all routes r is denoted by R.

We follow Wen [56, pp. 32, 33] and Kallehauge et al. [23, p. 70] and formulate the vehicle routing problem in formulation (2.1). Here, the $x_{i,j}^k$ are decision variables that are set equal to one when vehicle k uses arc $(i, j) \in K$ and zero otherwise.

$$
\begin{aligned}
f(\mathbf{x}) = \quad \min \quad & \sum_{k \in K} \sum_{(i,j) \in A} d_{i,j} x_{i,j}^k && && (2.1a) \\
\text{s.t.} \quad & \sum_{k \in K} \sum_{j \in V \setminus \{i\}} x_{i,j}^k = 1 && \forall\, i \in V_c, && (2.1b) \\
& \sum_{j \in V_c} x_{0,j}^k = 1 && \forall\, k \in K, && (2.1c) \\
& \sum_{i \in V \setminus \{0\}} x_{i,0}^k = 1 && \forall\, k \in K, && (2.1d) \\
& \sum_{i \in V \setminus \{h\}} x_{i,h}^k - \sum_{j \in V \setminus \{h\}} x_{h,j}^k = 0 && \forall\, h \in V_c, \forall k \in K, && (2.1e) \\
& x_{i,j}^k \in \{0,1\} && \forall\, (i,j) \in A, \forall k \in K && (2.1f)
\end{aligned}
$$

The objective function (2.1a) sums the costs of all arcs used. Constraints (2.1b) make sure that each client is served exactly once. Constraints (2.1c, 2.2d) ensure that each vehicle begins and ends its trip at the depot. Constraints (2.1e) make sure that flow is preserved. In other words, whenever a vehicle arrives at a customer, that vehicle must also leave that customer and vice versa. Lastly, constraints (2.1f) make sure that for any driver and any arc, that driver either drives that arc or not. An example of a VRP instance and its solution is given in figure (2.1).



(a) VRP instance data.

(b) Solved VRP instance.

Figure 2.1: Toy example of a VRP.

A solution $\mathbf{x}$ is called feasible if it complies to all the constraints of the VRP formulation (2.1b) –(2.1f). The set of all feasible solutions will be called F. The notation F will be overloaded for different problems, but it will be clear from the context which feasible set F represents. A solution that is not feasible is called infeasible. Infeasible solutions can for example not deliver to certain customers, or deliver to customers more than once, not begin or end at the depot or drive routes that do not form closed loops.

The vehicle routing problem is not easy to solve. As the number of customers increases, the number of different routes a vehicle can drive to serve all these customers is given by the factorial of the number of customers. Brute-forcing a solution is therefore not feasible.
In fact, the vehicle routing problem is a NP-hard problem [34]. The traveling salesman problem (TSP), which is NP-hard [24], can be easily reduced to the vehicle routing problem. This makes the vehicle routing problem NP-hard as well. This means that no polynomial time algorithm exists to solve it, except if P = NP, and that it is at least as hard as the hardest problems which can be verified in polynomial time.

To solve an arbitrary vehicle routing problem, we cannot feasibly rely on exact algorithms, as these would take too much time to solve large instances of the problem. Therefore we must resort to finding upper and lower

bounds for the objective function (2.1a).
Finding lower bounds can be done through relaxations of the problem: The vehicle routing problem is made simpler in a way that any feasible solution to the original vehicle routing problem is also a feasible solution to the relaxation as the relaxation has fewer constraints to comply with. If $\mathbf{x}$ is an optimal solution to the relaxation, then $f(\mathbf{x})$ is a lower bound for $f(\mathbf{x}^*)$, the objective value of the optimal solution for the non-relaxed problem.
Finding upper bounds can be done by applying heuristics. Heuristics are pragmatic methods to get good feasible solutions. These good solutions are not necessarily optimal. Any feasible solution $\mathbf{x} \in F$ gives an upper bound for the best solution, that is $f(\mathbf{x}^*) \leq f(\mathbf{x})$. The better a heuristic performs, the tighter the gap between the best found solution and the true optimal solution $\mathbf{x}^*$.

There is a wide variety of solution methods to solve VRPs. These include exact methods and heuristics. An overview of these algorithms is given by Zhang et al. [62], see figure (2.2). In this research, a genetic algorithm is used and adjusted (see Chapter 5 and Section 6.1 in particular).

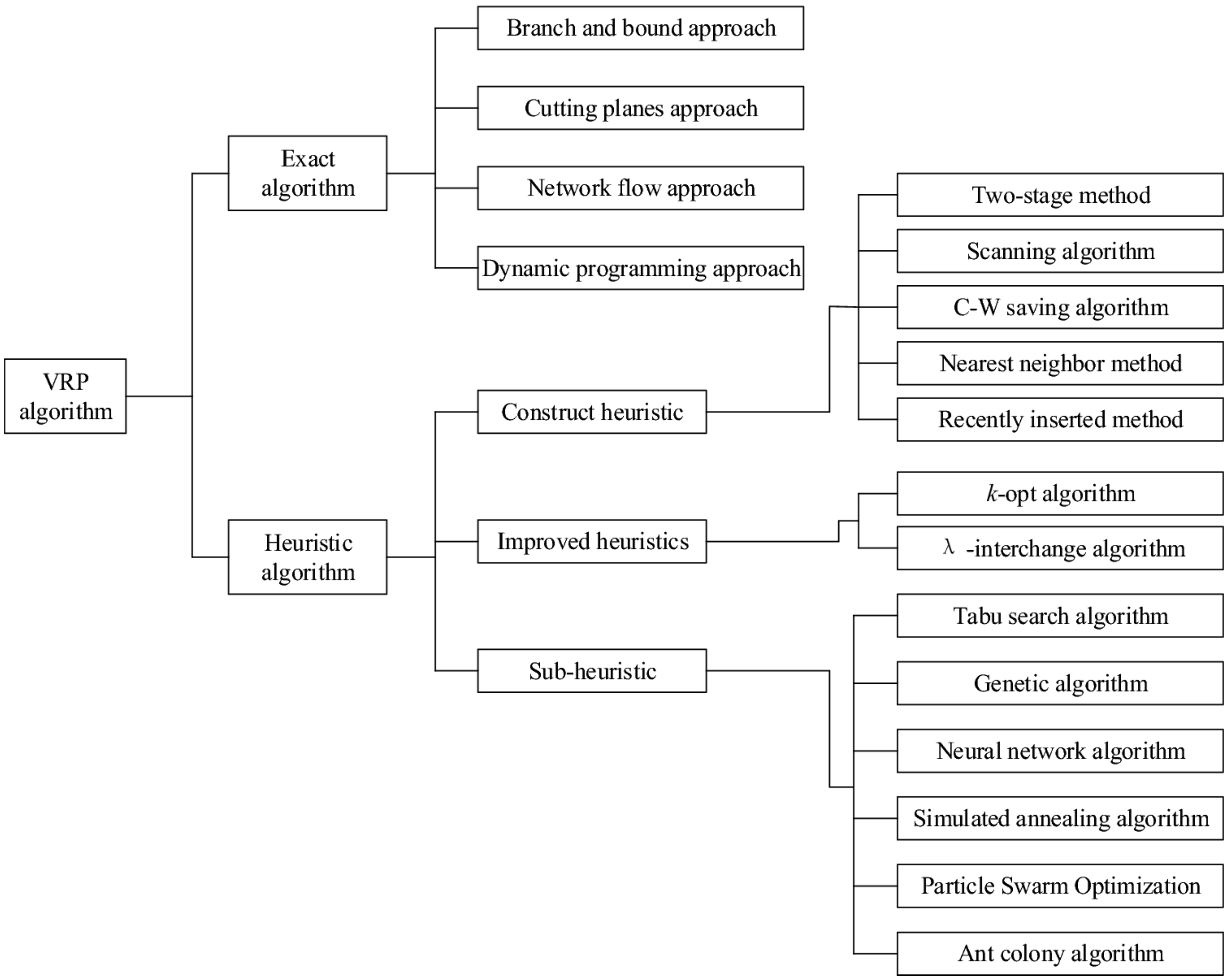


Figure 2.2: Exact and heuristic algorithm classification for solving a vehicle routing problem. [62, p. 198]

Over the years many variants of the classic VRP have been studied [62, 51]. Examples of such variants are the capacitated consistent vehicle routing problems (ConVRP), the vehicle routing problems with time windows (VRPTW), the open vehicle routing problems (OVRP) and the vehicle routing problem with stochastic demand (VRPSD), but many more are studied. VRPTW is one of the most prevalent and well studied of all these variants. The VRPTW forms the basis for the VRP model that has been researched and will be described in Chapter5.

2.1.2 The Vehicle Routing Problem with Time Windows (VRPTW)

The VRPTW, as the name suggests, makes use of time windows. A time window $[e_i, l_i]$ is a window in which a customer $i \in V_c$ can be served. Next to time windows, each customer i is assigned a demand q_i and a service time s_i. The depot is said to have a demand of 0, i.e. $q_0 = 0$. The capacities of all vehicles are assumed to be the same and are denoted by Q. Each arc $(i, j) \in A$ is additionally assigned a travel time $t_{i,j}$.

Service of a customer i can only start between e_i and l_i. A vehicle is however allowed to arrive at a customer i before e_i, but has to wait until e_i to start servicing that customer. The depot also has a time window, $[0, H]$. For the sake of feasibility it is assumed that $0 \leq e_i \leq l_i \leq H$ for any customer $i \in V_c$.

There are multiple ways to formulate the VRPTW (see the work by Kallehauge for an overview [22]). The following formulation follows the formulation also used by Wen [56, pp. 32, 33] and Kallehauge et al. [23, p. 70]. For convenience, we list all parameters, sets and decision variables.

- $d_{i,j} \in \mathbb{R}_{\geq 0}$: Distance between nodes i and j
- $q_i \in \mathbb{R}_{\geq 0}$: Demand of node i
- Q: Vehicle capacity
- $t_{i,j}$: Travel time between nodes i and j
- e_i: Start time of time window at node i
- l_i: End time of time window at node i
- s_i: Service time at node i
- K: Set of all drivers
- A: Set of all arcs between nodes
- V: Set of all nodes, consisting of all customers and the depot.
- V_c: Set of all customers

The decision variables and model formulation are then the following:

$$x_{i,j}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels from node } i \text{ to node } j. \\ 0 & \text{otherwise.} \end{cases}$$

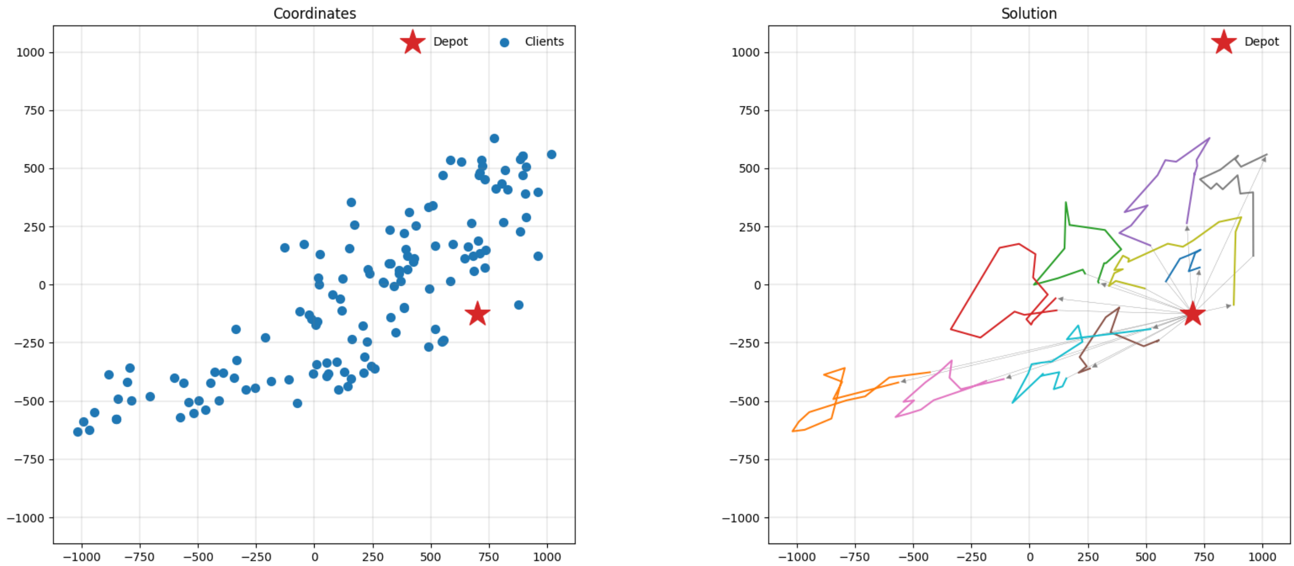
$$w_i^k \in \mathbb{R}_{\geq 0} : \text{time at which vehicle } k \text{ starts serving customer } i.$$

$$\begin{aligned}
f(\mathbf{x}) = \min \quad & \sum_{k \in K} \sum_{(i,j) \in A} d_{i,j} x_{i,j}^k && && \text{(2.2a)} \\
\text{s.t.} \quad & \sum_{k \in K} \sum_{j \in V \setminus \{i\}} x_{i,j}^k = 1 && \forall\, i \in V_c, && \text{(2.2b)} \\
& \sum_{j \in V \setminus \{0\}} x_{0,j}^k = 1 && \forall\, k \in K, && \text{(2.2c)} \\
& \sum_{i \in V \setminus \{0\}} x_{i,0}^k = 1 && \forall\, k \in K, && \text{(2.2d)} \\
& \sum_{i \in V \setminus \{h\}} x_{i,h}^k - \sum_{j \in V \setminus \{h\}} x_{h,j}^k = 0 && \forall\, h \in V_c, \forall k \in K, && \text{(2.2e)} \\
& \sum_{(i,j) \in A} q_i x_{i,j}^k \leq Q && \forall\, k \in K, && \text{(2.2f)} \\
& e_i \leq w_i^k \leq l_i && \forall\, i \in V, \forall k \in K, && \text{(2.2g)} \\
& w_j^k \geq w_i^k + s_i + t_{i,j} - M(1 - x_{i,j}^k) && \forall\, i,j \in V_c, \forall k \in K, && \text{(2.2h)} \\
& x_{i,j}^k \in \{0,1\} && \forall\, (i,j) \in A, \forall k \in K, && \text{(2.2i)} \\
& w_i^k \geq 0 && \forall\, i \in V, \forall k \in K. && \text{(2.2j)}
\end{aligned}$$

The formulation inherits all the constraints of the VRP formulation (2.1b - 2.1f) as well as the objective function (2.1a). Constraints (2.2f, 2.2h, 2.2g and 2.2j) are newly added to the problem formulation.
Constraints (2.2f) form the capacity constraints. Note that the total capacity that a route $r = (i_1, i_2, \ldots, i_s)$ takes on equals $q(S) := \sum_{i \in S} q_i$, where $S = \{i_1, i_2, \ldots, i_s\} \subseteq V_c$ and where q_0, that is the demand of the depot, is assumed to be zero. Route r is made up of arcs $((0, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{s-1}, i_s), (i_s, 0))$. Together, the selection of the customers these arcs originate from form constraints (2.2f).

The capacities could also depend on the vehicle (type) used —mathematically q^k —to allow for a heterogeneous fleet. Sometimes, VRPTW is assumed to not have capacity constraints. Capacity constraints are explicitly stated in the given formulation. The VRP formulated in Section 2.1 is called the capacitated vehicle routing problem (CVRP) when capacity constraints (i.e. constraints of the form of (2.2f)) are added.

Constraints (2.2g) make sure that the deliveries all start within their respective time window. Constraints (2.2j) ensure that customers can only be served after the depot has opened.
Constraints (2.2h) ensure that the next customer j can only be served after the previous customer i is served and the vehicle has travelled from the previous customer to the next customer. The constant M is a big enough number such that the constraint is always true whenever an $x_{i,j}^k$ equals 0. In that way, the constraint is only effective when $x_{i,j}^k$ equals 1, that is when arc (i, j) is driven on by vehicle k, such that the term $M(1-x_{i,j}^k)$ disappears.



(a) VRPTW instance data.

(b) Solved VRPTW instance.

Figure 2.3: VRPTW problem data and solutions. The instance comes from the used dataset. Coordinates are normalized to fit into an 1000x1000 grid as to not give away coordinates of clients of our case company.

Figure (2.3) gives an instance of the VRPTW as used in this thesis. We notice that the solution given does not seem optimal at first. For example, the green route and red route cross each other and the orange route is not the shortest possible. This has to do with the time windows. Solutions for VRPTWs indeed give different solutions than for VRPs.

2.2 Genetic algorithms

Genetic algorithms are metaheuristics that can be used to solve a variety of problems, including the Vehicle Routing Problem (VRP) and some of its variants. In the last decade, genetic algorithms have boomed in solving VRPs and its variants [38]. The intuitive understanding of them, their adaptability and potential for customization as well as their effectiveness has made them more popular and sprouted many variants of genetic algorithms, as for example hybrid genetic search.

2.2.1 General evolution

As genetic algorithms are inspired by the concept of evolution, they also inherit much of the jargon used in the study of evolution. To set the scene and to describe the working of genetic algorithms better, we will first briefly describe the working of evolution.
Consider a population. The population produces offspring using genes. Genes code for the characteristics of an individual. By having certain genes, they have a certain level of success in creating offspring in a given environment. This level of success in creating children in a given environment is called the fitness of an individual. When creating offspring, children receive a combination and possibly adaptation —called a mutation —of the genes of their parents. Therefore, the fitness of a child can be different from, but is related to, the fitness of their parents.

The probability of creating offspring increases with the fitness of the individual, thereby selecting which individuals will reproduce and which individuals will die not having created any offspring. This phenomenon is called selection. The process of selectively passing down genes based on the fitness of individuals is called evolution. Over time, the genes that code for good fitness will get more and more prevalent, while genes that code for worse fitness will become more sparse.

2.2.2 Evolution in genetic algorithms

As in evolution, genetic algorithms keep track of a population (also called pool). This population, in the context of genetic algorithms, is a set of solutions to a mathematical problem. Each member of the population has genes. For a solution to a mathematical problem these genes are some sort of characterization of the solution.
In each iteration of a genetic algorithm a new 'generation' of mathematical solutions is 'born'. Two parent solutions will combine and possibly adapt their genes —characteristics —and pass them down to their child solution. In this way, a new generation is formed. The function defining how offspring are produced from two parents is called the crossover operator.
In order to let the population evolve, their fitness to the environment must be measured. The environment is the mathematical problem at hand and the fitness of a solution is the evaluation of the objective function for that solution. The better the objective function of a solution, the better its fitness, hence the larger the probability that solution will pass down its genes to the next generation. Worse solutions have a smaller probability of passing down their characteristics. This will cause them to die out. By simulating evolution in this way good solutions for the mathematical problem at hand are tried to be found.

Genetic algorithms are highly customizable: How large should a population be? What characteristics of the solutions are captured in genes? What is the probability of getting children given a certain fitness? How is fitness measured? How are genes combined and possibly modified when getting children?

2.2.3 Hybrid Genetic Search

Hybrid genetic search (HGS) is a genetic algorithm that combines global search, in the form of a genetic algorithm, with local search methods. By using the best of both worlds, HGS leads to high-quality solutions. The steps taken in a hybrid genetic algorithm are the following:

1. Choose parents
2. Apply crossover operator
3. Apply local search procedure
4. Resize population

These steps are applied iteratively. One cycle of these steps is called an iteration. The best solution found at any point in time is stored. Figure 2.4 presents these ideas graphically.

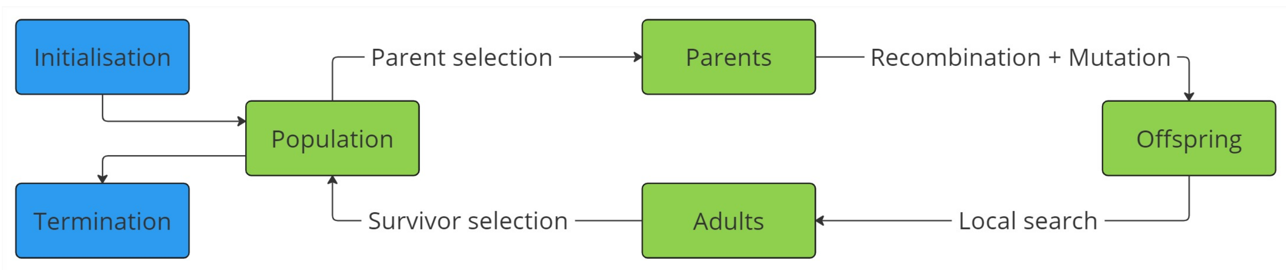


Figure 2.4: The general scheme of a hybrid genetic search algorithm as a flowchart. Adapted from [12, p. 27].

HGS methods use a k-ary tournament selection in the selection of two parents. Individuals are ordered from best fitness to worst fitness, where the first individual is individual 1. The algorithm first selects k individuals (which is the tournament size) from a population at random. Then, the algorithm selects an individual from the ordered tournament by selecting member i with probability $p(1-p)^{i-1}$. The probability p is chosen, where a larger value of p steers towards selecting more fit individuals.

A more elaborate pseudocode on HGS is given in the PyVRP documentation [1] and is displayed in Algorithm 1.

input: initial solutions $s_1, \ldots, s_n$
output: the best-found solution s^*
Set s^* to the initial solution with the best objective value
while *Stopping criterion is not met:* **do**
  Select two parent solutions (s^{p_1}, s^{p_2}) from the population using k-ary tournament.
  Apply crossover operator XO to generate an offspring solution $s^o = XO\left(s^{p_1}, s^{p_2}\right)$.
  Improve the offspring using a search procedure LS to obtain $s^c = LS\left(s^o\right)$.
  Add the candidate solution to the population
  if *s^c has a better objective value than s^*:* **then**
    $s^* \leftarrow s^c$
    quit
  end
  if *Population size exceeds maximum size:* **then**
    Remove the solution with the lowest fitness until the population is at minimum size
    quit
  end
end
return S^*

Algorithm 1: Hybrid Genetic Search

2.3 Multi-objective Optimization

Multi-objective optimization, as the name states, is the optimization of multiple objectives. Multi-objective optimization problems are all around us. Maximizing quality while simultaneously minimizing costs when buying something is one such example. Another example of optimizing on multiple objectives is maximizing the taste of a meal, maximizing the nutrition of a meal and minimizing the cost of a meal. The last example contains three objectives.

Often these objectives are not in each others interest. As an example, consider return per cost. When maximizing solely on return, the costs likely go up. When minimizing solely on costs, the return likely goes down as well. In multi-objective optimization, a best trade-off between the different objectives should be striven for.

This section loosely follows the books by Coello [8] and Branke et al. [4]. After introducing multi-objective optimization, notation and the concepts of domination and Pareto optimality are presented.

2.3.1 Pareto front

The objectives $(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ live in an objective space. The objective space is a vector space of k dimensions, which represent the different objectives. The objective functions form a vector function

$$\mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\right)^\top.$$

The objective vectors in the codomain of $\mathbf{f}$ live in the objective space $\mathbb{R}^k$. Without loss of generality, we can assume that every objective is to be minimized (if an objective were to be maximized, adding a minus sign in front of the objective function makes it a function to be minimized). The objective space is different from the decision space $\mathbb{R}^n$ in which all decision variables

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

live. The set of decision variables $\mathbf{x}$ that give feasible solutions $\mathbf{f}(\mathbf{x})$ is denoted as F. As multi-objective optimization is similar to linear or integer linear programming, just with multiple objectives, the notions of constraints and feasibility carry over.

As solutions have multiple objectives, a solution can perform better on some objectives and worse on other objectives compared to another solution. A solution is truly better than another solution if it is better in at least one objective and it is at least as good in all other objectives. This is the notion of domination. Formally, a feasible solution $\mathbf{x}_1$ is said to dominate another feasible solution $\mathbf{x}_2$ if for every $i \in [k]$, $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ and there exists $i \in [m]$ such that $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$. Domination is denoted as $\mathbf{f}(\mathbf{x}_1) \preceq \mathbf{f}(\mathbf{x}_2)$. If all inequalities are strong,

$\mathbf{x}_1$ is said to strongly dominate $\mathbf{x}_2$, denoted as $\mathbf{f}(\mathbf{x}_1) \prec \mathbf{f}(\mathbf{x}_2)$.

Whenever a solution is not strongly (weakly) dominated by any other feasible solution, it is called weakly (strongly) Pareto optimal. Typically, any multi-objective problem (MOP) has multiple Pareto optimal solutions. The set of all strongly Pareto optimal solutions is called the Pareto optimal set $P(F)$. The corresponding set in the objective space is called the Pareto front, which we will denote by $\mathbf{f}(P(F))$. Formally, given an objective vector $\mathbf{f}$ and a set F of feasible solutions of a MOP, the Pareto optimal set is defined as $P(F) := \{\mathbf{x} \in F \mid \nexists\, \mathbf{y} \in F : \mathbf{f}(\mathbf{y}) \preceq \mathbf{f}(\mathbf{x})\}$. Formally, $\mathbf{f}(P(F)) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in P(F)\}$.

All solutions in the Pareto optimal set can be considered optimal, just for different trade-offs. As the trade-off between all objectives can be different for different use-cases, the whole Pareto front is often striven to be discovered. The easiest way to construct the Pareto front is to weight the different objective functions and to optimize this weighted sum of objectives as a single objective function. This method is called the scalarization method or weighted sum method and is widely used in multi-objective optimization (see e.g. [25, 40]). Without loss of generality, this sum of weights can equal one, that is, $\min \lambda_1 f_1 + \cdots + \lambda_n f_n$, where $\lambda_1 + \cdots + \lambda_n = 1$. Trying out different weights will construct part of the Pareto front.
A drawback of the traditional scalarization method is that for non-convex Pareto fronts, not the whole front can be discovered [20, 25]. A Pareto front is said to be non-convex if there exists a convex combination of points on the Pareto front that dominates a point on the Pareto front. Using the scalarization method will not pose an issue in this thesis.

Chapter 3

Literature Review

This chapter aims to present and analyze relevant literature. Section 4.2 presents the need for incorporating driver satisfaction into vehicle routing problems and describes the most important driver satisfaction factors to consider. As the first objective of this thesis is on the incorporation of workload preference and region consistency into the routing problem model, Section 3.2 covers workload preference measures and Section 3.3 covers literature related to ways to use regions in the context of driver satisfaction.

To compare implementation methods, a central measure must exist on which methods can be compared. This central measure must characterize workload preference and region consistency explicitly. Moreover, the central measures for both workload balance, region consistency and routing costs must be compared through a multi-objective measure. Therefore, multi-objective optimization is also part of this literature review and is treated in Section 3.4. Our contribution is presented in Section 3.5.

3.1 Driver satisfaction

Drivers leave their companies often and are often dissatisfied with their job. This section examines how pressing driver shortage and turnover in the trucking industry is, and what makes drivers satisfied in their job.

3.1.1 Driver shortages and turnover rate

A distinction is often made between truckload (TL) and less-than-truckload (LTL). In truckload shipping the truck is filled entirely with only one type of truckload. In less-than-truckload the truck is filled with truckload of multiple customers. Drivers in TL often drive longer hours and have further destinations to travel to than drivers in LTL.

The turnover rate in truck load drivers is enormous. Driver turnover rate of truckload drivers in the United States has consistently been over 80% annually in the period from 2012 to 2018 [9]. To compare, turnover rate for LTL drivers in the same period averaged around 10%.

Besides a high turnover rate, there is a shortage of truck drivers in general. All types of drivers combined, there is a shortage of an estimated 233.000 drivers in Europe (7% of all truck driver jobs) and a staggering approximate 2.2 million drivers in China, accounting for 12% of driver jobs unfilled [15]. The number of drivers needed in the future is only expected to increase [9].

3.1.2 Driver satisfaction factors

Much research is done to what makes drivers satisfied [33, 57]. As driving TL is more demanding and turnover rates are far higher, driver satisfaction research is primarily focused on this group of drivers. For truckload, the lifestyle, loneliness, image, long hours, stress, poor pay and health-related issues draw people from trucking [9, 14, 57]. Next to the absence of these factors, drivers value good condition of the equipment [11], freedom in their work and good general working conditions [33]. However, these sources focus mainly on truckload and do not focus specifically on what makes drivers satisfied regarding their routing.

The need for incorporating other factors than costs was already suggested over half a century ago. Kirby and McDonald criticize routes that "would be quite unacceptable to any transport manager" [26, p. 305] such as routes that cross themselves, cross other routes multiple times or too many short routes [26]. They suggest to take other factors into account: "In our opinion these difficulties call into question the normally accepted criterion of 'optimality' and suggest that there is a place for a subjective factor in assessing optimality". [26, p. 306] In the PhD dissertation by Allison [2], next to Kirby and McDonald, Mole [41] and Waters [55] are cited stating similar ideas.

Bruinink concluded that the two most important factors contributing to driver satisfaction, in terms of routing itself, are their workload preference as well as their region consistency [5]. Experience in driving a region is particularly important as this experience lessens the cognitive workload required to drive there [43]. Interviews by Bruinink and internal communication also acknowledge the value of experience considering driver satisfaction. Other than this explorative study, in which drivers were interviewed directly and asked for their preferences, and suggestions made by Kozyreff, Meerbergen, and Zobiri [31], little research has been done in what exactly attributes to drivers' satisfaction in their routing.

To incorporate driver satisfaction explicitly into a routing algorithm, two kinds of measures must be known. First of all it must be known what a driver satisfaction concept means for a particular driver. The mathematical modeling of a driver satisfaction factor is called it's (dis)utility measure $\mathbf{u}$ [46, 52]. The utility measures of all the drivers must be compared in a certain way. The way in which this comparison between drivers is done is called the (un)fairness measure $\phi(\cdot)$ [46, 52].

3.2 Workload preference measures

Bruinink concludes from his interviews with drivers that regarding working times and the nature of their route, the primary focus of drivers is on their own work hours, hence concluding that route duration is a grounded measure for driver satisfaction [5]. In this section we will go over proposed measures and theories considering workload preference.

There are a lot of fairness measures. Tsang and Shehadeh [52, p. 2] state that "quantifying fairness is challenging because there is no best definition or measure of fairness that is universally accepted". They have unified the different fairness measures in a new framework and give an overview of commonly used deviation-based fairness measures as can be seen in Table 3.1. Let $\mathbf{u}$ represent the numerical outcomes of N subjects of interest, such as the workload of drivers. For $N > 3$, Tsang and Shehadeh proved the following relations between the presented measures: $\phi^{(i)}(\mathbf{u}) = \phi^{(iii)}(\mathbf{u})$ and $\phi^{(vii)}(\mathbf{u}) = N\phi^{(vi)}(\mathbf{u})$. This leaves six commonly used deviation-based fairness measures. Matl, Hartl, and Vidal also give such a table [40].

Table 3.1: Existing deviation-based fairness measures. Here $\mathbf{u}$ is a vector of length N and $\overline{u}$ is the mean over all entries of $\mathbf{u}$. [52, p. 3]

Index	Measure	Name
i.	$\max_{i\in[N]} u_i - \min_{i\in[N]} u_i$	Range
ii.	$\sum_{i=1}^{N}\sum_{j=1}^{N} \lvert u_i - u_j\rvert$	Gini deviation
iii.	$\max_{i\in[N]}\max_{j\in[N]} \lvert u_i - u_j\rvert$	Maximum pairwise deviation
iv.	$\sum_{i=1}^{N} \lvert u_i - \overline{u}\rvert$	Absolute deviation from mean
v.	$\left[\sum_{i=1}^{N}(u_i - \overline{u})^2\right]^{1/2}$	Standard deviation
vi.	$\max_{i\in[N]} \lvert u_i - \overline{u}\rvert$	Maximum absolute deviation from mean
vii.	$\max_{i\in[N]}\sum_{j=1}^{N} \lvert u_i - u_j\rvert$	Maximum sum of pairwise deviation
viii.	$\sum_{i=1}^{N}\max_{j\in[N]} \lvert u_i - u_j\rvert$	Sum of maximum pairwise deviation

All measures in Table 3.1 are theoretically good candidates for fairness measures. All these measures satisfy continuity, normalization, symmetry, Schur convexity, translation invariance and positive homogeneity properties making them convex fairness measures [52]. Convex fairness measures generally pose less optimization burden than non-convex fairness measures [52].

Janssens et al. [20] give two measures to balance workload for heterogeneous drivers. They do however not balance workload over a time horizon. Both methods use a standard deviation based method (see v. in Table 3.1). They normalize v. by dividing by N inside the square root. First they propose to balance workload separately for all different values of expected workload. This is useful in practice if the fleet of vehicles is large and the amount of different expected workloads is small (e.g 8 hours and 4 hours). In other cases, this method is too

restrictive. Second, they propose for u_i to normalize effective workload by expected workload. This way the imbalance in relative deviation from expected workload is minimized. This method is similar to the method proposed by Bender, Kalcsics, and Meyer [3].
While many studies have focused on a homogeneous crew of drivers, Bender, Kalcsics, and Meyer balance workload for a heterogeneous crew [3]. Each driver in this heterogeneous crew has different contract hours. Through using the relative working times, drivers working times can still be compared to each other even though their contract hours vary. This way potential overtime is also fairly balanced over the drivers.
To balance the workload for a heterogenous crew of drivers, Bender, Kalcsics, and Meyer use an equity measure on the relative work hours. u_i is given as the ratio of realized work time to the contract hours. They use maximum absolute deviation from mean (vi in Table 3.1) as their fairness measure. This way, the largest deviation of relative workload to the mean relative workload is minimized. Deviations in shorter work times weigh heavier than deviations in longer work times.

A totally different approach is taken by Linfati, Yáñez-Concha, and Escobar [37]. They aim to balance workload by making compact routes. They consider the mean distance of customers to the centroid of the route, where the centroid is the geographic mean of the customers in that route. They minimize the sum of these in-compactness measures over all vehicles k. Besides that, they also minimize a maximum absolute deviation from mean, where the mean is the mean of the realized worktimes (driving time and waiting time).
This approach has as benefit that drivers can be compared to each other even though they have different expected working times. A drawback is that relative overtime explodes when the expected workload is small (close to zero). This makes the method less reliable and difficult to interpret.

In the interviews performed by Bruinink [5], preference is given to overtime being spread out over multiple days instead of having a single day with all the overtime. As an example Bruinink states that drivers prefer to work an hour extra on four different days than to work four hours extra on a given day. Therefore, the measure proposed is minimizing the variance of the workload. This can either be done per day [3, 20] or over an extended time period. Bruinink made the assumption that all drivers prefer to work the same amount of hours every day.
Dileepan [10, p. 79] also prefers the sum of squared deviation (which is similar to the variance except for a normalization factor) as a measure for workload imbalance. In comparison to the sum of absolute deviation and the range of workload, Dileepan argues that the sum of squared deviations is preferred. They state three reasons: In a pairwise exchange, the sum of squared deviations can be easily computed, large deviations are heavily penalized and more information is taken into account than purely the extreme values.

3.3 Region consistency

In the evaluation of driver satisfaction regarding the regions in which a driver drives, Bruinink concludes that both their preference as well as their experience are factors to consider [5]. This section presents literature on how experience can be forced upon drivers and how it can be measured. Literature on strategies incorporating preferences is also reviewed.

3.3.1 Introduction

Consistency in the VRP is a broader term that can be evaluated in different ways. Three key-terms are region consistency, customer consistency and driver consistency. Customer consistency is the extend to which drivers deliver to the same set of customers each time they drive. Region consistency is the extend to which drivers drive in the same region each time they drive. Lastly, driver consistency is the extend to which customers are visited by the same driver each time they are visited. Driver consistency was introduced by Groër, Golden, and Wasil [16] and has grown into its own VRP variant, known as the ConVRP (see e.g. [49, 29, 61, 35]). For a formulation of ConVRP see for example [35].

Because the motivation for our study is the large turnover rate of drivers, the focus in region consistency is from the drivers perspective. That means no attention is given to the ConVRP explicitly. However, when striving for driver-centric region consistency, customer-centric region consistency will also rise [21]. Indeed, when drivers drive more in the same areas, the customers in that area will likely be delivered more by that driver. This establishes a personal connection between drivers and customers [16], resulting in increased customer satisfaction.

Many papers approach the goal of consistency via a two-stage approach: First, regions (or master routes) are created on a strategic level. The making of these regions is called the districting phase [48]. When districts (master routes) are created, drivers are assigned to these districts (master routes). This offers a lot of consistency at the price of not being flexible. In the second stage, routes are created on an operational level (see e.g. [3, 48]), but based on the regions (master routes) made in the first stage. In this routing stage, routes are created on a daily basis. By routing on a daily basis flexibility is allowed. This combination of a strict, consistent and strategic districting with a flexible operational routing allows for consistent yet cost-efficient routes.

3.3.2 Strategic regions

There are many different ways to make regions. Regions can be described either by customers that are inside the region or by a geographical area (see e.g. [3, 7]). Christofides [7] writes that these problems are ultimately the same.
On any given day, customers fall into one of the made regions. Following a strict region assignment, solving the VRP variant ends up solving a subproblem for each driver: Given the customers assigned to the driver, how can the driver best traverse these customers. This is a far easier and smaller problem to solve than the original VRP variant.
It is important for regions to be compact, as drivers will have a high degree of flexibility traversing customers in the region without increasing distance too much [3]. This is especially important in the presence of time windows in which some traversals are less favourable or even infeasible [3]. A high region familiarity can be achieved while not increasing travel costs too much [49].

Master routes

Next to defining regions, region consistency can also be achieved through the use of master routes (tactical plan) [28]. Master routes are made using customer data and represent typical routes that drivers take. The idea is to assign each driver to a master route. In the operational planning, the deviation from the master route is to be minimized somehow. This can be done through distance-based measures [50] and/or through route similarity measures [31].

When creating master routes you ideally do not deviate from them too much. Therefore, master routes are most effective when clients are stable over time. This works well in a business to business model in which customers can be stable over months and have a clear pattern in when they are to be delivered.
In a home delivery setting, typically, the customers are more random. A customer can be delivered on certain days of the week, but can also be delivered on a totally different day of the week, or time or with a different demand. This variety in customer behaviour over time makes using master routes a less feasible strategy in our home delivery use-case.
In VRPTW problems, master routes do not work well. Time windows can make it impossible to arrive at some customer during its time window. Quality of the operational routes will be poor, or, operational routes will be very dissimilar to the master routes, invalidating the need for master routes in the first place.

3.3.3 Flexible regions

The biggest drawback of holding on to strictly assigned territories is that the routing becomes static. Drivers will deliver every single customer that is within their assigned region. Softening the hard constraints of strictly assigned territories facilitates a more adaptive response to daily demand fluctuations. Softening also enables potential to conform better to driver satisfaction objectives.

Another disadvantage of territory assignment is that the routes are expected to stay somewhat consistent over time. In home delivery, new customers can come and go and demands can be different over different days. Having flexible regions can therefore be beneficial.

Instead of having fixed regions to which drivers are assigned, Bender, Kalcsics, and Meyer allow these fixed regions to be adapted to daily demand, making the territory assignment soft [3]. They show that soft territory assignment is effective for region consistency and workload balance, while not increasing routing costs too much. That is, drivers must be able to switch between territories on a daily, operational, basis. They used a heterogeneous set of drivers.

Another way to introduce flexibility in route making is to only attribute a subset of the total number of customers to the drivers, initially. Such a method can be viewed as creating partial districts. Zhong, Hall, and Dessouky [63] put the most representative customers of a route together into what they introduced to be a 'core region'. Furthermore they introduced the terms 'exclusion zone', which is an area close to the depot, and 'flex zone', which is area that is not in a core region nor in the exclusion zone. In contrast to core regions, flexible regions are not tied to a driver. Flexible regions are often times smaller than strictly assigned districts [20, 36, 49, 63]. They assign every driver a core region. All customers inside a core region must be served by that driver. All customers in the flex zones can be attributed to certain drivers, which can differ each time a route schedule is made. This way, they try to find a balance between consistency for driver familiarity on the one hand and flexibility for optimal routing on the other hand.

Janssens et al. [20] go even further, eliminating the districting steps altogether. They do not use core-areas and rely entirely on microzones. Although no regions are made, a tactical plan still exists.

Other papers study the effects of driver experience (e.g. [6, 36, 45, 49, 63]). All these studies consider experience to affect delivering speed or the travel time between customers as drivers can more easily find paths that are efficient based on their experience in the region. As far as we are aware no study, except for the thesis by Bruinink [5] and the measures proposed by Kozyreff, Meerbergen, and Zobiri [31], considers driver experience in the context of driver satisfaction.

Learning curves and forgetting curves

Besides preference regions, driver satisfaction can also be captured in driver experience [5]. To take into account driver experience, it is important to add a dynamic element into the route making. Such a dynamic element can be captured through the use of learning and forgetting curves, which model the experience of workers based on the tasks they do. Learning and forgetting curves can be classified in classes of models. The classes are log-linear, exponential, and hyperbolic models. We refer readers that wish to know more about experience models to the bundled work edited by Jaber [19]. In particular, the paper by Fogliatto and Anzanello [13] provides a good basis.

To the best of our knowledge, Zhong, Hall, and Dessouky were the first to introduce the concept of learning and forgetting elementary components of routes, introduced as cells or microzones, based on the experience a driver has in driving or delivering in that cell [63]. They used this experience to adjust the delivery time of a driver to the customers in that particular region by a factor. The model they used was the log-linear model by Salah [47]. Smilowitz, Nowak, and Jiang [49] adopted the concept and usage of cells in combination with learning customers as well as regions. Their philosophy is to increase customer and region familiarity, of which the latter, in a dynamic setting, can be interpreted as driver satisfaction.

3.3.4 Measures

Kozyreff, Meerbergen, and Zobiri propose a measure in which a satisfaction rate is assigned to the edge between any two customers [31]. The satisfaction rate for a given edge is taken as the mean over all feedback given by drivers for that edge. This measure differs from the learn-and-forget experience-based measure in the fact that this measure is equal for all drivers, links instead of cells are considered and the measure is adaptive through feedback instead of through experience. The satisfaction measures can be further tailored towards the drivers by adding driver specific parameters [31].

Driver satisfaction can be obtained by either directly or indirectly solving for it, or through a combination of the two [5, 31]. A decision can be made based on the additional computational costs, where direct implementation has more computational cost than a combination of direct and indirect solving, which has more computational cost than indirect solving. The indirect case is also called 'driver assignment' by Bruinink [5].

In driver assignment, driver satisfaction measures are taken into account only after the routes have already been made. The drivers are assigned to the routes such that the minimum satisfaction is maximized. Any other metric can be considered as well. In the direct approach, driver satisfaction factors are already considered when optimizing the routes. When making the routes it is already known which driver will drive which route. Incorporating driver satisfaction factors directly into the route making can bring extra computational costs. Combining a direct and indirect method can potentially be less computationally heavy than the direct approach, while also yielding better results than the driver assignment.

Region consistency can be measured by counting the number of different drivers that serve a customer, called customer consistency. Yang, Ni, and Song optimized on customer consistency. When the same number of customers are used, they state that having one driver deliver most times and all the others only a few times is preferred over having all drivers deliver that customer the same amount of times. In their corresponding measure, delivering to the same customer more often is therefore rewarded. This reward is however always in the context of percent-wise deliveries, not in the absolute amount of deliveries to that customer. Still, this reward can be interpreted as drivers gaining experience delivering to that customer and indirectly rewarding driver satisfaction.

3.4 Multi-objective optimization in driver satisfaction

Across different studies, significant improvements in driver satisfaction can be made by only increasing the routing costs by at most a few percent [5, 40, 49]. Matl, Hartl, and Vidal [40] state that the trade-off structure between different objectives does not depend much on the equity function, but does depend on what type of balance is considered. In the context of balancing distance/duration, many different Pareto-optimal solutions are found, whereas in the context of balancing the number of stops the found Pareto-optimal points are less diverse in the number of stops.

Size of Pareto front

The size of the solution set is a lot different for different equity measures [40]. Solely using a maximum decreases the size of the Pareto front by a factor 5 in smaller instances and by a factor 7 in larger instances in the use-cases and optimization methods of Matl, Hartl, and Vidal [40]. Standard deviation is one of the best measures to use when considering the size of the Pareto front obtained.
Although the chosen equity measure does not influence the trade-off between different objectives much [40], Matl, Hartl, and Vidal [40] state that many studies have shown that the choice of equity objective does have a significant impact on the structure of the solutions. In terms of the hypervolume measure, which is a way to compare different Pareto fronts, and considering distance balancing, the range, mean absolute deviation, standard deviation and Gini index are very similar. They summarize stating that "the Pareto-optimal solutions generated by different equity functions are seldom identical, but their quality is often similar". They also state that solutions typically exist that respect many different equity measures. Lastly, in their extensive study on the Pareto fronts related to incorporated driver satisfaction aspects, they conclude that specific areas of the Pareto front typically contain a certain solution structure.

These conclusions by Matl, Hartl, and Vidal are valid for balancing route duration (and load and stops for that matter). A question is whether these conclusions also hold valid for balancing deviation from driver specific preferred workloads over their working days.

Robustness of Pareto front

Matl, Hartl, and Vidal [40, p. 126] state that "Most multi-objective methods —and particularly genetic algorithms —implicitly assume that Pareto-optimal solutions share common characteristics, especially if those solutions have similar objective function values." They have researched the connection between solution similarity in the objective space and similarity in the decision space in the context of bi-objective optimization of costs versus route durations, number of stops (customers) and load. Solutions found on the Pareto front are found to be reasonably similar in the decision space, having a median similarity in edges of around $55 - 65\%$. When comparing solutions next to each other in the found Pareto optimal set, the similarity in the decision space increases, reaching as high as a median similarity in edges of around $75\% - 80\%$.
Next to showing robustness of the Pareto front, this suggests that a decision maker can indeed reliably choose a piece of the Pareto curve that seems interesting. Choosing between schedules on a particular piece of the Pareto curve should result in reasonably similar schedules, making it easier for the DM to choose a planning.

The PyVRP software available at ORTEC is a hybrid genetic search algorithm. The result by Matl, Hartl, and Vidal [40] on the fitness landscape gives more motivation of using a genetic algorithm; Solutions on a similar location on the Pareto curve typically have a similar underlying structure, that is, solutions are similar on the decision space, meaning that evolution of the algorithms is more reliable.

3.5 Contribution

This research focuses on increasing driver satisfaction by tailoring routing to the preferences and experiences of drivers while not increasing routing costs too much. To this end workload will be balanced for a heterogeneous set of drivers over a time horizon. Drivers preferences for regions and experiences in regions will also be considered. Driver satisfaction will be examined over a real-life case study.

Workload balance has been researched quite a lot already. In most cases, drivers are homogeneous. For homogeneous drivers, all papers considered balancing workload over all drivers. All routes are steered to have the same length, resulting in all drivers working close to the same amount of hours. In heterogeneous crews, relative workload is balanced. Deviations from smaller workloads are deemed more important than deviations from larger workloads.

In this study, preferred workloads are central, no matter whether this corresponds to contract hours, free schedules or anything else. Deviations from preferred workloads are all equally important, no matter the preferred workloads. For balancing, we also spread out deviations over multiple days, using a time horizon of 30 days.

The central measure for workload preference is inspired on the standard deviation. This is because all of the data is involved in this measure, because deviations are penalized more if they are concentrated on fewer days as opposed to spread out over more days and because the Pareto front will be bigger. We will extend a workload measure to account for heterogeneous drivers.

There has been done a lot of research in region consistency, most often with driving consistently in regions as the sole objective. Drivers are assigned regions that are designed for drivers to be the most productive, not acknowledging preferences drivers might have for regions. And even if their assignment to a region was based on their preferences, drivers would consistently drive precisely these regions, not evaluating their experiences in neighbouring regions. Part of the papers that research region consistency apply learning and forgetting curves to evaluate experience. They do this from an efficiency point of view: Driving in similar areas makes drivers deliver faster.

We will use driver learning in the context of driver satisfaction. We acknowledge drivers' preferences to drive in certain regions as well as the experience they obtain when serving customers in regions. Delivering customers in preference regions and regions drivers have experience in will be rewarded.
By using driver experience, the strategic and operational steps might be able to be performed together. Using driver experience has the potential to naturally form districts. As no districts are set in stone and optimization happens daily, experience naturally engages with daily demand fluctuations. Besides the dynamic forming of regions, experience in itself adds to the satisfaction of drivers.

Chapter 4

Problem description

The problem in question revolves around a vehicle routing problem with time windows (VRPTW). In this vehicle routing problem customers have to be served by vehicles. The classical VRPTW will be extended to also optimize for the driver satisfaction factors: workload preference and region consistency. The goal of this thesis is to find improvements in both workload preference as well as region consistency during the optimization process, while not increasing routing costs too much.

In this problem description first some assumptions related to the VRPTW variant at ORTEC will be stated and motivated in Section 4.1. Then the incorporation of driver satisfaction in our problem is described at a high level (Section 4.2). Then, the assumptions and decisions made for workload preference as well as for region consistency to form our problem will be stated and described in Section 4.3 and Section 4.4 respectively.

4.1 VRPTW related assumptions

For every day, an instance of the VRPTW (see formulation (2.2)) is solved that is adapted to include the driver satisfaction factors. The VRPTW takes a set of drivers and customers as input and outputs a set of routes. In this subsection, routes, drivers and customers will be briefly described.

4.1.1 Routes

Each route begins and ends at the depot, of which there is only one. Between each pair of consecutive customers, the shortest route, given by mapping software at ORTEC, is implicitly assumed to be driven. Drivers only deliver to customers, load is only picked up at the depot.

4.1.2 Drivers

For any day, the scheduled drivers drive exactly one vehicle. The vehicles are assumed to be homogeneous, which means that they have the same capacity and speed. A vehicle is driven on exactly one route. Although the vehicles are assumed to be homogeneous, the drivers that drive the vehicles are not considered to be equal.

The drivers carry characteristics related to their satisfaction. Each day, a subset of the set of drivers is scheduled to work. This set of drivers can differ for each day. The work schedule is assumed to be given. In this thesis, no flexible drivers or call-in drivers will be considered.

The capabilities of drivers are all the same. Hence no distinction is made between experienced and inexperienced drivers, young and old drivers etc. in terms of driving and delivery pace. That is, seniority is in no way affecting the modeling decisions. Drivers are also all assumed to cost the same, that is, their routing costs, which consist of the total time driven (including needing to wait when a time window is not yet opened) and the total distance traveled, are calculated in the same way.

4.1.3 Customers

Every customer has a location, which is given by an x and y coordinate. Additionally, each customer has a time window in which a driver may arrive. The time windows are assumed to be known and to be certain, that is, the time windows are not probabilistic in nature. If a driver arrives too soon at a customer, the driver will wait until the time window for that customer opens. If a driver arrives later at a customer than at the end of their time window, the solution is infeasible.

4.2 Driver satisfaction

In addition to optimizing on routing costs, consisting of total driving time and total distance driven, this thesis aims to effectively and efficiently optimize on driver satisfaction as well. Two driver satisfaction factors are considered:

- Workload preference
- Region consistency,

which will be described in more detail in the following sections.

These two driver satisfaction factors, together with routing costs, lead to a multi-objective optimization problem. The objectives are not necessarily agreeing; Optimizing for one of the objectives is paid for by the other two. Good balances between the three objectives are sought after.

We can think of driver satisfaction in multiple ways. Driver satisfaction can be evaluated on an individual basis or can be balanced among the different drivers. The latter notion is called fairness. In this research, driver satisfaction is both measured on an individual basis as well as balanced between drivers.
Driver satisfaction could be described on single VRPTW instances. However, satisfaction can be better described over longer periods of time. Consistency is an emergent property that is obtained when solving multiple VRPTW instances. Fairness is also better described over longer periods of time. Moreover, results obtained are more reliable when obtained over a longer period of time. The time period considered will be called the time horizon.

Not all VRPTW instances to be solved will require the same number of vehicles, and hence drivers. A work schedule for the entire time horizon is assumed to be known on day 1. We assume routes are made on a daily basis. On any given day t, the routes and driver assignment to these routes of days 1 up to and including day $t-1$ are known. These routes and related data are referred to as the history available on day t. This history will be used in the optimization of day t. We iteratively solve every day up to and including the time horizon.

We aim to enhance driver satisfaction over a time horizon of 30 days, following Bruinink [5]. There are two reasons for this. First, we only use the history available on day t and the work schedule when optimizing day t. Routes are not made in advance for the entire time horizon. This makes finding good solutions more difficult, and a reasonably long time horizon will help to find good solutions. Second, fairness and consistency can be better described for longer time horizons.
A time horizon than is longer than 30 days is less realistic as the work schedule is less likely to be available that many days in advance. Besides that, having an even longer time horizon would increase computational costs, while not adding significantly more information. A thirty-day time period is also realistic for planning purposes as thirty days constitute a month.

We note that routes will not be elongated just for the sake of workload preference or region consistency. All else equal, a driver will not drive any extra time, delivering no one, just to achieve their preferred workload or to longer be in suitable regions.
Region consistency is two-fold in this research. On one hand, drivers are more satisfied with region familiarity. The goal is to maximize the experience that drivers have in the regions they visit. On the other hand, drivers are more satisfied with preferred regions. The goal here is to maximize the number of times that a driver visits a preferred region and to minimize the number of times that a driver visits a disliked region.

A thirty day period is evaluated on the performance of all thirty days combined.

4.3 Workload preference

Drivers are assumed to all have preferred workloads. A preferred workload is a time in seconds that that driver wishes to work on a single trip from and to the depot. In practice, such a preferred workload could also represent contract hours or a more elaborate system of combinations of preferences by the employer and employee. The preferred workloads can differ between the days and between the drivers.

A motivation for using workload preference are the interviews done by Bruinink, indicating that drivers do not care that much about the workload of others, but mainly care about themselves [5]. Another motivation is the fact that not all drivers work the same amount of hours every month, work on the same days or work the same amount of hours on any given day.

This is a key difference between this thesis and the thesis by Bruinink [5]. We consider heterogeneous drivers, whereas Bruinink considered homogeneous drivers instead [5]. They assumed all drivers are the same in that they all want to work the same amount of hours. Their objective is to have the mean workloads of drivers over the time horizon as close to each other as possible. This is why they called this driver satisfaction factor 'workload balance'.
In our heterogeneous case, our objective is two-fold. The objective is to have the realised workloads as close to the preferred workloads as possible while spreading workload deviations over multiple days. Moreover, we want to distribute workload deviations reasonably fair over all drivers. Since we are dealing with preferred workloads in a heterogeneous setting, we call this workload related driver satisfaction factor 'workload preference'.

4.4 Region consistency

For region consistency, both preference as well as experience are considered important [31]. Together, they will form the region consistency measure. By including experience, routes When drivers display preference in regions, they will be rewarded when driving in these regions. Moreover, drivers gain experience when driving regions, which is rewarded when drivers drive in these regions. When drivers do not drive in a region, they lose experience. Similar to workload preference, region consistency will be evaluated on an individual level as well as between drivers.

Chapter 5

Model design

This chapter aims to present a detailed model design for the driver satisfaction factors. To this end, Section 5.1 builds towards the workload preference measure and Section 5.2 builds towards the region consistency measure. Both measures will be combined into a single objective function using the scalarization method in Section 5.3.

5.1 Workload preference

Drivers are assumed to all have preferred workloads, in seconds, that driver wishes to work on a single trip from and to the depot. The preferred workloads can differ between the days and between the drivers.
As each driver has their own preference hours, a comparison with the mean amount of work hours over all drivers is insufficient. Even for a single driver, comparing the work hours to the mean preference hours is insufficient as even for one driver, their preference hours may differ over different days.
Each driver is assumed to want to work a set amount of time on each day. For driver d, their preference hours are $w_d^{\text{pref}} = (w_{d,1}^{\text{pref}}, w_{d,2}^{\text{pref}}, \ldots, w_{d,T}^{\text{pref}})$, where T is the amount of days that are considered for the plan horizon (by default, we set $T = 30$). It is assumed that all preference hours of drivers are known before optimization for any given day that takes place.

Interviews done by Bruinink concluded that drivers prefer to spread out overtime over multiple days instead of being realized on a single day. For example, drivers prefer to work one hour of overtime on four different days over having four hours of overtime in just a single day. For drivers it is important to spread out deviations as much as possible. This also enhances workload predictability as drivers cannot easily be confronted with much overtime on a single day.
In an ideal world where all workload preferences are met, we want a workload preference related measure to equal zero and vice versa. This way it is more easy to interpret workload preference costs and scaling can be done more effectively. Moreover, we would like a measure to improve whenever a solution is better. Compare e.g. a standard deviation based measure to a min-max measure.
Because of these reasons, and reasons stated in the literature review (Chapter 3) supporting standard deviation based measures, we propose the following elementary measure on which we will build further:

$$\sum_{\tau=1}^{t} \left(w_{d,\tau} - w_{d,\tau}^{\text{pref}} \right)^2,$$

where t is the day that is optimized for and $w_{d,\tau}$ is the realized workload of driver d on day τ.

In order to compare different drivers, this measure must be suitably normalized. All drivers are considered equal, so we divide by the number of days a driver has worked. We obtain the following on standard deviation based measure:

$$u^{\text{wp}}(d,t) := \frac{\sum_{\tau=1}^{t} \left(w_{d,\tau} - w_{d,\tau}^{\text{pref}} \right)^2}{N_T^{\text{days}}(d)}, \tag{5.1}$$

where $N_T^{\text{days}}(d)$ is the number of days driver d drives in total up until the time horizon T. This measure provides spread of deviations from preferred work hours between all work days for a single driver. This measure also allows for comparison between different drivers.

We could have divided by $N^{\text{days}}(d,t)$, where $N^{\text{days}}(d,t)$ would be the number of days driver d has driven up to and including day t. This method has a clear advantage: By dividing by $N^{\text{days}}(d,t)$, one would only need to know the number of days the driver has worked up to then, instead of having to know future schedules. This

method would also have some disadvantages, which are advantages of (5.1).
If we would divide by $N^{\mathrm{days}}(d,t)$, the first few days would be more important than the latter days from the perspective of the optimizer. On any day, the value of $u^{\mathrm{wp}}(d,t)$ would roughly be the same as we average over the number of days. However, the part of this measure that can be influenced on the day that is solved shrinks every day. Indeed, on any given day t, the history up to that day is fixed. The absolute value of $u^{\mathrm{wp}}(d,t)$ does not matter for the solver, only the differences that the local steps make. Therefore, dividing by $N^{\mathrm{days}}(d,t)$ would unfairly prioritize the earlier days over the latter. Dividing by $N_T^{\mathrm{days}}(d)$ solves this issue.

A more elaborate method was also considered that uses a convex combination over drivers of squares of differences of the current day's preferred and realized workloads. The values of the convex combination can be based on history and serve to reach fairness among drivers. This method would also value each day the same in the optimization process. Added benefits of this method are that not a whole schedule —including future days —is needed to calculate the measure. Moreover, perfecting a single day would result in the measure being equal to zero, as every term revolves around the present day. This opposes (5.1), where perfecting a single day does not necessarily nullify the measure. A drawback of this measure is that it does not scale with the measured workload preference deviations. It does not measure precisely the detriment we feel not complying to the workload preferences. It measures the relative adherence to workload preference between drivers. Therefore, we chose not to use it.

Balancing workload equally among the drivers does not work as drivers have different preferred workloads. Balancing $u(d,t)$ between all drivers also does not work: a highly undesirable schedule in which all deviations from the preferred workloads are equal, but very large, would render a zero. This could potentially steer into more deviation from preferred workloads, which is undesirable. The ideal is not to have the same relative deviations from preferred workloads as other drivers; the ideal is for all drivers to have their realized workloads equal their preferred workloads.
As for any driver the ideal deviation from their workloads equals zero, we take the square deviation from this ideal scenario and result in a measure based on standard deviation. The history up to day t is used in the unfairness measure. The resulting measure is in accordance with the results obtained by Matl, Hartl, and Vidal [40] and Dileepan [10]: computational feasibility. The following workload preference unfairness measure will be used throughout the thesis:

$$\phi^{\mathrm{wp}}(t) := \frac{1}{\#D} \sum_{d \in D} \left(u^{\mathrm{wp}}(d,t)\right)^2 . \tag{5.2}$$

By normalizing over the number of drivers, this measure becomes invariant over the problem size (in terms of number of drivers). When this measure gets multiplied with a scalar that is related to the problem size, a relative importance scalar in the scalarization method can be relatively constant. This benefits ease of use of this measure. In practice, scalars used in the scalarization method would ideally be estimated based on the problem instance at hand.

The preferred workload might not exactly represent the total workload for a single day. However, this is not a problem, as first of all, the real workload for any given day is not known before optimizing and estimations for workload are also imprecise a month in advance in the context of home delivery. And second of all, as workload is not increased just for the sake of workload preference, having the workload preferences potentially be higher than realized workloads poses no problem. Modeling the preference hours roughly, but not exactly, to the real workload will also make the results in this thesis more robust and the conclusions stronger.

5.2 Region consistency

In order to quantify what are considered preference regions for a driver and what are regions a driver has experience in, the distribution area will be discretized. The distribution area of a home delivery company is the geographical area in which customers can be served. The distribution area will be split in several regions r_i that make up the set of regions R. The regions form a partition of the distribution area.

This section aims to build towards a region consistency measure. To this end, preference regions are considered (Subsection 5.2.1), experience regions are considered (Subsection 5.2.2) as well as their combination (Subsection 5.2.3).

5.2.1 Preference regions

Drivers consider certain regions preferable over other regions. In order to increase the job satisfaction of drivers, these preferences are taken into account. Drivers can choose whether a region is preferred or not. For a driver d, the preference and neutral regions are denoted as R_d^1 and R_d^0 respectively. Preference regions of drivers are assumed to be known. It is assumed that every driver chooses the same number of preference regions.

When a region is preferred by a driver, visiting a customer in that region will result in a reward of 1. These rewards are normalized for the number of customers a driver visits on a given day. We notice that preference regions are, although not equal in spirit, mathematically similar to an assumed tactical plan in [20]. This results in the following preference adherence reward:

$$f^{\text{pr}}(d,t) := \frac{1}{N^{\text{cust}}(d,t)} \sum_{c_i \in R_d^1} x_{d,c_i,t}, \tag{5.3}$$

where $x_{d,c_i,t}$ is a decision variable that equals 1 if driver d delivers customer c_i on day t and 0 otherwise. $N^{\text{cust}}(d,t)$ is the amount of customers driver d has delivered to on day t. The value of $f^{\text{pr}}(d,t)$ is between 0 and 1, where a value closer to 1 indicates better performance on preference regions.

5.2.2 Experience regions

In addition to preference regions, we acknowledge the importance of familiarity with regions. Familiarity in our context has not to do with the speed at which drivers serve their customers. As drivers learn new routes and become familiar with these regions and with the customers, they enjoy their work more [31]. The experience a driver has delivering a region is therefore tied to their satisfaction. This driver satisfaction factor will be taken into account in the multi-objective optimization process.

Drivers learn to drive in a region when they have visited at least one customer in that region on a given day. In contrast to, for example, Zhong, Hall, and Dessouky [63], a driver does not have to deliver every customer in a region. In fact, by delivering only one customer in a region, the driver gains familiarity with the entire region [49]. The level of experience a driver has with a particular region was termed customer access costs (CAC) by Smilowitz, Nowak, and Jiang [49], but we will simply call this experience. The experience of all drivers on all regions is put into the objective function.

Experience can be modeled in a variety of ways. Common model classes are the log-linear, hyperbolic, exponential, and multivariate models (Fogliatto and Anzanello [13]). All these models model variation in service rates based on experience. In decades of making new models, the log-linear model by Wright [59] is still one of the most widely accepted [18]. We refer readers that wish to know more about experience models to the bundled work edited by Jaber [19]. In particular, the paper by Fogliatto and Anzanello [13] provides a good basis.

The functions we will use to denote the experience of a driver are based on the work by Zhong, Hall, and Dessouky [63] and are slightly adapted to suit our usecase, which is the modeling of driver satisfaction instead of service rates. Zhong, Hall, and Dessouky make use of learning and forgetting curves following the log-linear model and use them to model service rates in VRPs. When a driver delivers to at least one customer in a region, this region is learned. Vice versa, when a driver does not deliver at least one customer in a region, this region is forgotten. The learning and forgetting curves that we will use are given by:

$$L(t) = \min\left(E_1 t^l, E_\infty\right) \tag{5.4}$$

$$F(t) = \max\left(E_\infty t^{-f}, E_1\right). \tag{5.5}$$

In these functions, E_1 denotes the experience a driver has the first time they serve a customer in a particular region (i.e. they had no experience yet), whereas E_∞ denotes the limit of experience (i.e. the experience after '∞' days). $L(t)$ denotes the experience a driver has after t consecutive days of serving at least one customer in a particular region, given that the driver did not have any experience in that region prior to learning. $F(t)$ has an equivalent meaning, but considers forgetting: $F(t)$ denotes the experience a driver has after t consecutive days of not serving any customer in a particular region, given that the driver had the maximum experience in that region prior to forgetting. l and f are parameters between 0 and 1, which denote the learning and forget rate respectively. The larger the values of l and f are, the quicker a region is learnt c.q. forgotten. In the context of driver experience, drivers typically learn a particular region far quicker than they forget a region. Therefore $0 < f << l < 1$, which contrasts the assumption by Zhong, Hall, and Dessouky, who assumed l to be equal to f.

The experiences considered for all drivers on all regions attain values between E_1 and E_∞. Bounding experiences makes sure that optimizing on routing costs remains the priority. When a driver learns a region by visiting that region on a given day, their experience gets updated based on the learning curve. The current experience value determines where on the learning curve that region currently is. The associated time t on the learning curve for a given experience value can be calculated by solving (5.4) for t (disregarding the min and max). When a driver learns, their experience will be updated based on the increase in value of the learning curve between that point t and the point $t+1$. Forgetting happens in the exact same way, but with the forgetting curve instead of the learning curve.

The iterative update procedure is as follows:

$$E_{dj}(t) = \begin{cases} \min\left\{ E_1 \cdot \left(\left(\frac{E_{dj}(t-1)}{E_1} \right)^{1/\ell} + 1 \right)^{\ell}, E_\infty \right\} \text{if driver } d \text{ visits at least one customer in region } j \text{ on day } t \\ \max\left\{ E_\infty \cdot \left(\left(\frac{E_{dj}(t-1)}{E_\infty} \right)^{-1/f} + 1 \right)^{-f}, E_1 \right\} \text{if driver } d \text{ does not visit region } j \text{ on day } t. \end{cases} \tag{5.6}$$

Here $E_{dj}(t)$ is the experience a driver d has in region j on day t. In this recursive function,

$$\left(\frac{E_{dj}(t-1)}{E_1} \right)^{1/\ell}$$

and

$$\left(\frac{E_{dj}(t-1)}{E_\infty} \right)^{-1/f}$$

give day $t-1$ for the learning curve and forgetting curve respectively on which these curves evaluate to $E_{dj}(t-1)$. This can be seen by solving (5.4) and (5.5) for t (disregarding the min and max). Adding 1 for the next day, this value acts as t in (5.4) and (5.5). In other words, given a prior experience value, we calculate day $t-1$ corresponding to this experience on either the learning curve or the forgetting curve. After that, the new experience is calculated based on t. In fact, $E_{dj}(t)$ is calculated from $E_{dj}(t-1)$. This results in the final recursive formula (5.6).

Similar to equation (5.3), we define the total experience a driver interacts with during their route, normalized for the amount of customers they serve as follows:

$$f^{\text{ex}}(d,t) := \frac{1}{N^{\text{cust}}(d,t)} \sum_{c_i} x_{d,c_i,t} E_{dj_{c_i}}(t). \tag{5.7}$$

5.2.3 Combining driver preference and driver experience

The driver preference and driver experience are combined into the region consistency measure. That is,

$$f^{\text{rc}} := \lambda f^{\text{pr}} + (1-\lambda) f^{\text{ex}}.$$

f^{rc} takes values between 0 and 1, because both f^{pr} and f^{ex} take values between 0 and 1. Similar to (5.1), we relate f^{rc} to the ideal value, which is 1 for all drivers. This results in the following formula

$$u^{\text{rc}}(d,t) := \frac{\sum_{\tau=1}^{t} \left(f^{\text{rc}}(d,\tau) - 1 \right)^2}{N^{\text{days}}(d,t)}. \tag{5.8}$$

Similar to the unfairness measure for workload preference (see (5.2)), we obtain an unfairness measure related to region consistency:

$$\phi^{\text{rc}}(t) := \frac{1}{\#D} \sum_{d \in D} \left(u^{\text{rc}}(d,t) \right)^2. \tag{5.9}$$

5.3 Combining costs, workload preference and region consistency

For feasible solutions, the solver developed by ORTEC called PyVRP evaluates costs for distance (f^{dist}) and duration (f^{dur}), which we call routing costs. Costs for workload preference (ϕ^{wp}) and region consistency (ϕ^{rc}) are added, which we call driver satisfaction costs.

Similar to the work by Bruinink [5], driver satisfaction costs are interesting to analyse when compared to the routing costs. When optimizing solely on routing costs, let the best solution found be solution s^* with costs c^*.

The driver satisfaction objectives are scaled by c^* to compare the driver satisfaction costs to the standard costs more easily. To this end, the VRPTW must first be solved purely on routing costs. The obtained c^* will then be used to solve the same VRPTW instance again, but modified to including the driver satisfation factors.

The final objective to be optimized is:

$$f := f^{\text{dist}} + f^{\text{dur}} + c^* \left(\alpha \phi^{wp} + \beta \phi^{rc} \right). \tag{5.10}$$

This objective function, together with the VRPTW constraints from problem formulation (2.2) form the problem to be solved on any given day. For convenience, we will name this formulation the DSVRPTW (Driver Satisfaction Vehicle Routing Problem with Time Windows) throughout this thesis. The VRPTW without driver satisfaction factors, which consists of f^{dist} and f^{dur}, will be called VRPTW.

Chapter 6

Methods

This chapter describes three different solution approaches to solve the DSVRPTW: random allocation, driver assignment and an integrated approach. All three methods make use of routing software available at ORTEC called PyVRP. PyVRP makes use of a hybrid genetic algorithm. We adapted the software to support the three solution approaches.

As PyVRP is used in all three solution approaches it will be described in Section 6.1. Then the discretization method is presented in Section 6.2. After that, the three solution approaches are described in Section 6.3. After these methods to solve the DSVRPTW are described, the hyperparameter tuning of our learn and forgetting curves is presented in Section 6.4.

6.1 PyVRP software

The PyVRP software is based on a state-of-the-art open-source genetic algorithm for the capacitated vehicle routing problem [53]. This algorithm is extended to support time windows. Additional construction heuristics and a crossover operator called selective route exchange (SREX) [42] were implemented [27]. The local search procedure is inspired by the SWAP* neighbourhood introduced by Vidal et al. [54] and is intensified in PyVRP [27].

In this section will be described how PyVRP works. First, PyVRP will be initialised. In the subsequent subsections each step of the hybrid genetic search (parent selection, recombination and mutation, local search and survivor selection) will be described in more detail and in the context of PyVRP. This section follows the steps done in the code-base sequentially and makes use of the documentation of PyVRP to explain certain steps in more detail [44]. For a more in depth description of the PyVRP software we refer to the papers by Kool et al. [27] and Wouda, Lan, and Kool [58] as well as the documentation of PyVRP, which is available at `https://pyvrp.org/`.

6.1.1 Initializing PyVRP

When PyVRP is ran, it is initialised with a data instance containing the following data:

- Routes
  - Route ID
  - Earliest start time
  - Latest stop time
  - Capacity Q_i
  - Maximum driving duration
- Tasks
  - Task ID
  - Address
  - Demand q_i
  - Time window $[e_i, l_i]$
  - Service time s_i
- Depot
  - Address.

The data can be made more sophisticated and heterogeneous. All heteregeneous data that are homogeneous in our use-case are omitted in the description of the input for simplicity. Besides the data instance, the PyVRP reads optional arguments (flags) and a set of hyperparameters of which some are described in more detail in the documentation [44]. PyVRP uses this user input and the hyperparameters to construct objects such as the population, local search, solution and genetic algorithm.

6.1.2 Parent selection

The population gets initialised with random solutions. By default 25 solutions are in the initial population, although support exists for user-made initial solutions.
Every iteration one child will be added to the population. No selection procedure is done. However, the algorithm holds a maximum population size. Note that a solution here is a solution to the complete problem. A solution can (and will) have multiple routes.

The new child will be made through k-tournament selection in which two individuals will be drawn. By default, k equals 2. The selection pressure is $p = 1$, resulting in a deterministic tournament. So in any tournament the best solution is always chosen. In the default case of $k = 2$, i.e. a binary tournament, effectively the best of two randomly chosen solutions is selected.

In order to keep the population diverse, a diversity measure is taken into account. The diversity measure makes sure that the amount of difference between the two chosen solutions is between two threshold values. This is done because a diverse population will result in faster conversion. A population can also become too diverse, resulting in solutions of less quality. Drawing a parallel to evolution in real life: A population that is heavily inbred has low potential for evolution. On the other hand, a population that is too diverse will not survive well; For example, a bird survives well having wings and a mammal can best have two or four legs.
If this diversity criterion is not met, the second solution will be substituted for another randomly chosen solution. If the diversity measure has failed 10 times, the second solution is accepted anyhow.

The diversity is measured by the symmetric broken pairs distance (BPD). This measure calculates for every customer whether the preceding customer is the same for both solutions as well as whether the following customer is the same for both solutions. Formally,

$$BPD(s_1, s_2) = \frac{\sum_{i=1}^{n} \mathbf{1}_{p_{\text{sol}_1}(i) \neq p_{\text{sol}_2}(i)} + \mathbf{1}_{s_{\text{sol}_1}(i) \neq p_{\text{sol}_2}(i)}}{2n},$$

where sol_1 and sol_2 are two solutions and where $p_{\text{sol}}(i)$ and $s_{\text{sol}}(i)$ represent the preceding customer (or depot) and succeeding customer (or depot) respectively [44]. A score closer to 1 indicates more diversity whereas a score closer to 0 represents less diversity.

6.1.3 Crossover operator

A crossover operator is the recombination of parents and possible mutation of solutions to obtain the next generation of solutions. Selective route exchange [42], as the name suggests, selectively chooses routes to exchange. It carefully selects routes from the second parent (parent B) that can be exchanged with routes from the first parent (parent A). Parent A is dominant in both children as the routes that will not be exchanged will all be from parent A in both children.

For illustration purposes, imagine that solutions A and B consist of many routes and for simplicity imagine that only one depot is used. The planned routes will roughly form circle segments in the geographical space (see Figure (6.1) [32] for an illustration). This fact is used by the SREX algorithm. Let the polar angle of a route be the polar angle of the geographical center of the customers (and depot) served in that route with respect to the geographical center of all customers and depots in the problem instance combined (i.e. the geographical center of all locations $i \in V$). We say a 'circle segment' for a solution consists of a set of routes such that no other route exists in that solution that has their polar angle in between any of the polar angles of the routes in the circle segment. In other words, a circle segment of routes is a set of routes for which their polar angles are in a row when ordered.

To effectively combine two parents to make a child, routes from both parents have to be chosen that can most easily be exchanged. This is where the circle segment representation comes in. Both circle segments, say $A_{\text{circ}} \subset A$ and $B_{\text{circ}} \subset B$ consisting of a selected number of routes in solutions A and B are turned such that they overlap each other as good as possible. That is, the amount of tasks in $A_{\text{circ}} \setminus B_{\text{circ}}$ is to be minimized. The number of routes considered in the circle segments is chosen uniformly by default.

To this end, first the routes in solutions A and B are ordered by ascending polar angle. A number of moved routes is given as input. Circle segment A_{circ} is moved either counterclockwise or clockwise by one route. The same is done for segment B_{circ}. The difference in size of $A_{\text{circ}} \setminus B_{\text{circ}}$ is evaluated for all four movements. The movement that decreases $A_{\text{circ}} \setminus B_{\text{circ}}$ the most is performed. This process is iterated until no further improvement can be found.

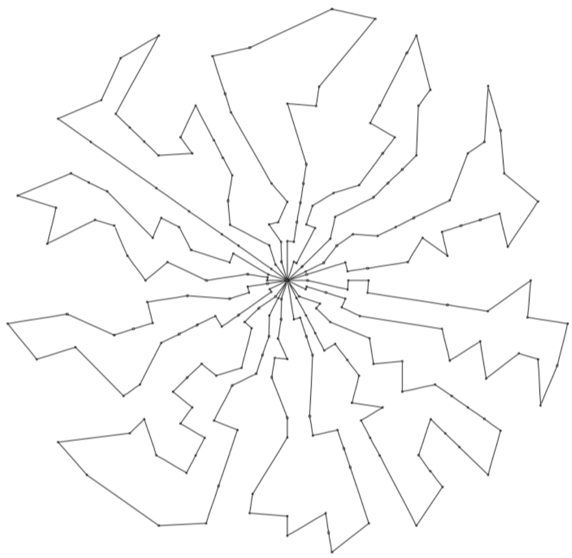


Figure 6.1: Planned routes will roughly form circle segments in the geographical space. [32, p. 1319]

It can not be guaranteed that $A_{\text{circ}} \setminus B_{\text{circ}}$ can be empty. However, by making $A_{\text{circ}} \setminus B_{\text{circ}}$ as small a possible, most of the routes are kept intact for the children. The broken routes are repaired in the local search procedures.

To describe the children of parents A and B more effectively, let A_{circ} be the set of routes in the circle segment of A and let A_{circ}^C be the set of routes in A not in the circle segment of A (i.e. the complement). Similarly, let B_{circ} be the set of routes in the circle segment of B and let B_{circ}^C be the set of routes in B not in the circle segment of B. The children of parents A and B are the following:

- Child 1: $B_{\text{circ}} \cup \left(A_{\text{circ}}^C \setminus B_{\text{circ}}\right)$
- Child 2: $(A_{\text{circ}} \cap B_{\text{circ}}) \cup A_{\text{circ}}^C$,

where $A_{\text{circ}}^C \setminus B_{\text{circ}}$ takes routes in A_{circ}^C and throws out customers that are in B_{circ} and $A_{\text{circ}} \cap B_{\text{circ}}$ takes routes in B_{circ} and keeps all customers that are also present in A_{circ}. The child with the best objective function, in which penalties are also taken into account, is selected.

Offspring always inherit their vehicle type from parent A. Notice that this is not problematic for our use-case, as the routes that are swapped between the two parents are similar. These routes are in roughly the same region. The routes can however have a totally different length, but will converge over time to be comparable between parents.

6.1.4 Local search procedure

Several local searches are performed in PyVRP. These include node operations as well as route operators.
The node operators are of the form (α, β). Such an operator selects α consecutive nodes from the first route and β consecutive notes from the second route. These node segments are swapped between the routes. The most elementary exchange operators have a special name: (1,0)-exchange is called relocation, as one single customer gets relocated to another route. (1,1)-exchange is called a swap as two customers of different routes are swapped. The routes operators use information from the whole route when performed. This makes them more expensive than node operators. One can either swap two routes from the very first customer after leaving the depot or swap two customers and inserts them in the best place in the other route [53].

6.1.5 Concluding remarks

Throughout the optimization process solutions can be infeasible. PyVRP has a mechanism in place that lets the percentage of infeasible solutions in the population be roughly constant throughout the optimization process. PyVRP achieves this by adjusting the penalty for being infeasible [27].

Right now we have established a good view of the software, PyVRP, that will be used in this research. All steps of the hybrid genetic algorithm as used in PyVRP are explained in detail. PyVRP can be explained in even more detail, but this would extend to much on the necessary information needed to comprehend the inner workings of

PyVRP. Moreover, this would rub against what can reasonably be written without breaching company secrets.

In this section not only did we describe the core steps performed by PyVRP, we also discussed how some of these steps are altered to serve the driver satisfaction objectives. Again, readers that want to know more about the inner workings of PyVRP are referred to its documentation [44] and to the papers by Wouda, Lan, and Kool [58] and Kool et al. [27].

6.2 Discretization of the map

To facilitate driver learning and prioritize specific areas, cells (regions) are created. Having cells keeps the freedom to schedule drivers freely, but also rewards drivers for driving in areas consistently. Especially in our case where time windows are considered, having the flexibility to use part of another area is useful. The time and distance saved by serving customers that would be just outside a strict area outweigh the reduction in region consistency caused by serving in another area. However, rewarding driving in the same (smaller) area adds consistency while not giving up too much flexibility.

To make regions, the map of the served area is discretized. This is similar to the methods by Zhong, Hall, and Dessouky [63], Janssens et al. [20], Bruinink [5] and Li et al. [36]. Instead of using postal codes [63, 5], we adapt the idea by e.g. Li et al. [36] and Janssens et al. [20] who defined cells as identical squares which make up the service region. We will make regions by discretizing the map in hexagons.
Using hexagons over squares when discretizing the map has several benefits. First, hexagons pose fewer edge effects than squares. This is because the hexagon is the figure with the smallest edge to area ratio that can be used to fully tile a surface. Second, hexagons are more compact than squares as the mean distance to the nearest center of a hexagon is smaller than the mean distance to the nearest center of a square given the same area. Moreover, all neighbouring cells are equidistant. Squares have neighbours either directly or diagonally adjacent to them. The center of a square is closer to the center of squares directly adjacent to them than to squares diagonally adjacent to them. Hexagons, however do not possess this issue, making them more regular than squares.

To discretize the service area in hexagons, the H3 library in Python is used. The H3 library in Python sees a hexagon as an object. A hexagon is precisely described by its position and size (rotations are fixed and depend on the size of the hexagon), which also encodes a labeling to index the made cells. The sizes of the hexagons can have 16 values, where a size of 0 is the biggest (≈ 2.6 million km^2, around a fourth of the United States or Europe) and size 15 is the smallest ($\approx 0.9\text{m}^2$). Each hexagon has seven child hexagons within the parent, so that the area of a hexagon of size i is seven times smaller than that of a hexagon of size $i-1$.

A reasonable size of the hexagons has to be chosen. Cells that are too large do not allow for distinction between areas. Cells that are too small fail to define an area to be learned. Moreover, we prefer most regions to contain some clients each day. The hexagons that we will use have size 6, which have an area of $\approx 36.13\text{km}^2$. An example of this grid is given in figure (6.2). For illustration purposes artificial customers are added.

6.3 Solution approaches

Three solution approaches are described: Random allocation, driver assignment and an integrated approach. These solution approaches are similar to the solution approaches by Bruinink [5].

6.3.1 Random allocation

As a baseline approach random allocation will be used to assess driver satisfaction. First, routes will be made without taking driver satisfaction into account. This is done for all 30 days. Since driver satisfaction is not taken into account, history is also not taken into account in the optimization process. After all 30 days are optimized on routing costs, drivers are assigned to routes on a daily basis at random. The random allocation method does not take driver satisfaction into account and serves as a benchmark for more clever approaches.

6.3.2 Driver assignment

Similar to the random allocation, drivers are assigned to predetermined routes. Unlike random allocation, the driver assignment approach assigns drivers to predetermined routes in a clever way. Drivers will be assigned to routes such that a given convex combination of driver satisfaction objectives is optimized. As drivers are assigned to routes that were made solely optimizing on routing costs, no additional costs for driver satisfaction are made.

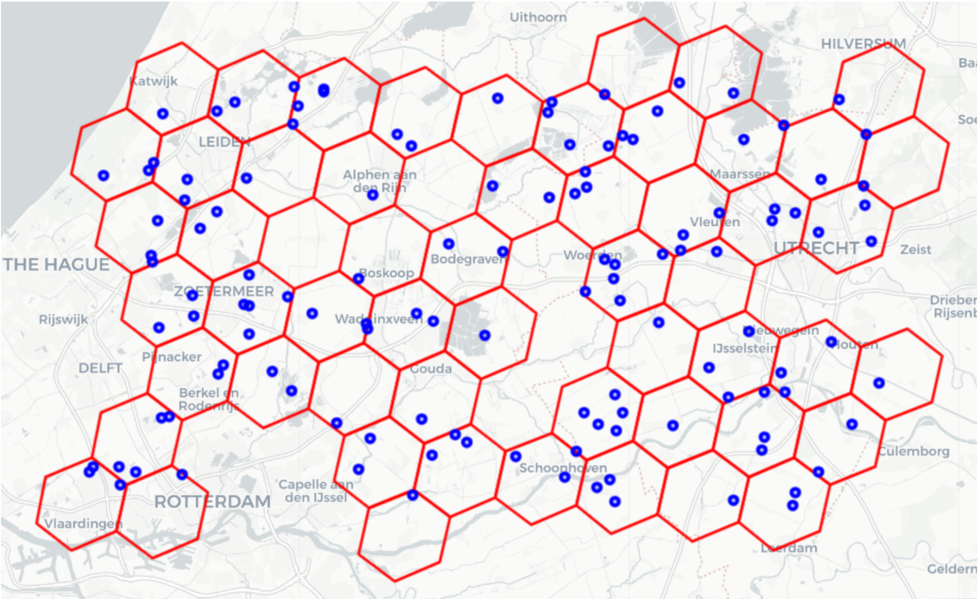


Figure 6.2: Example discretization of the map. Clients are fictional. Client density is comparable to the data used in this research.

Given any α and β, an assignment of drivers to routes is made that minimizes

$$\alpha \phi^{\mathrm{wp}(t)} + \beta \phi^{\mathrm{rc}(t)}. \tag{6.1}$$

This is done iteratively over the days. Only knowledge about assignments of drivers to routes of previous days is used to optimize expression (6.1). As a driver assignment is done for a particularly day t, the experiences are updated. These new experiences are used when assigning drivers to routes on day $t+1$.

The driver assignment is done using the commercial solver CPLEX. The constraints used in driver assignment are given below:

$$f(\mathbf{x}, t) = \min \quad \alpha y + \beta z \tag{6.2a}$$

$$\text{s.t.} \quad \sum_{k \in K} x_{k,r} = 1 \qquad \forall r \in R_t, \tag{6.2b}$$

$$\sum_{r \in R} x_{k,r} = 1 \qquad \forall k \in K_t, \tag{6.2c}$$

$$\frac{1}{|K_t|} \sum_{r \in R_t} \sum_{k \in K_t} x_{k,r} \left(u^{\mathrm{wp}}(k,t)\right)^2 \leq y \quad , \tag{6.2d}$$

$$\frac{1}{|K_t|} \sum_{r \in R_t} \sum_{k \in K_t} x_{k,r} \left(u^{\mathrm{rc}}(k,t)\right)^2 \leq z \quad , \tag{6.2e}$$

$$x_{r,k} \in \{0,1\} \quad \forall r \in R_t, \forall k \in K_t, \tag{6.2f}$$

$$y, z \in \mathbb{R}. \tag{6.2g}$$

Here, K_t is the set of drivers that work on day t. R_t is the set of routes made on day t. We note that $|K_t| = |R_t|$ on any day t.

Driver assignment can be performed much faster than the integrated approach. The restriction in computing power is hence not on driver assignment. Therefore, many more different ways can be tried in driver assignment as opposed to the integrated approach. 300 different weight pairs are used as (α, β). We notice that the values of α and β do not matter in the absolute sense: only their fraction determines their relative importance. The fractions α/β used in any driver assignment are between 10^{-35} and 10^5 and are evenly distributed on a log-scale. The most extreme fractions have been chosen in such a way that no better solutions in either of the two driver satisfaction factors can be established. To only solve for either of the two driver satisfaction factors, $(\alpha, \beta) = (1, 0)$ and $(\alpha, \beta) = (0, 1)$ are solved for as well. In any result that considers driver assignment, only the Pareto optimal set of found solutions are displayed.

6.3.3 Integrated approach

The integrated approach optimizes on driver satisfaction as well as the routing costs (distance and duration). The way this is done is similar to the approach of Bruinink [5]. In contrast to the random allocation and driver assignment approach, the integrated approach takes driver satisfaction into account when making routes. The routes are assigned to the drivers at the very start of the route making process. Routes are made to explicitly also optimize on the driver satisfaction factors. The driver assignment to the routes is embedded.

This approach will use the PyVRP software available at ORTEC, which uses a hybrid genetic search algorithm (see section (6.1)). This software is adapted to optimize on driver satisfaction as well. The structure of the PyVRP software remains untouched. That is, the types of search procedures, the parent selection, the crossover operators and the resizing of the population remain untouched. Moreover, the hyperparameters stay tuned as they were when driver satisfaction was not considered.

PyVRP is altered to also optimize explicitly on the driver satisfaction factors. The objective function is changed as to incorporate workload preference and/or region consistency. Although the inner structure of PyVRP remains unchanged, all four hybrid genetic search steps are altered through the change in the objective function to be optimized.

We state the final objective of the DSVRPTW (Driver Satisfaction Vehicle Routing Problem with Time Windows) given as (5.10) in Section (5.3) again:

$$f := \left(f^{\text{dist}} + \psi^{\text{wp}} + f^{\text{otdur}}\right) + c^* \left(\alpha\phi^{wp} + \beta\phi^{rc}\right). \tag{6.3}$$

The routing costs consist of the total distance driven and the total route duration. These objectives are respectively denoted by f^{dist} and ψ^{wp}. The workload preference costs are given by ϕ^{wp} and the region consistency costs are given by ϕ^{rc}. Depending on whether workload preference and/or region consistency is taken into account, α and/or β are unequal to zero.

The workload preference and region consistency measures are normalised so that they give values in the same order of magnitude no matter the number of drivers or number of days considered. To relate these costs to the routing costs, these measures are scaled by the best costs found when optimizing solely on the routing costs. This value is given by c^*. First, an instance has to be solved purely on routing costs to obtain c^*, which in turn is used when solving DSVRPTW.

Similar to the integrated approach, several values of α and β are taken. Using this scalarization method a variety of solutions will be obtained. These solutions will form the heuristically obtained Pareto front. By fitting a line (or plane) through the points making this Pareto front and by assuming this Pareto front is convex, an approximation of the best result obtained by using other values for α and β can be obtained. It is up to a decision maker (DM) to select priorities between the objectives that align with their preferences.

Every single day-instance is ran for 60 seconds. The c^* needed for DSVRPTW is obtained after running only 2 seconds for small instances and 5 seconds for medium sized and large instances. Running PyVRP for 60 seconds using driver satisfaction roughly translates to a few hundred iterations (hitting a couple thousand iterations for small instances and for the first few days in which not much history is used and hitting 200 iterations for large instances and for the last few days in which much history is used).

6.4 Hyperparameter tuning

Ideally there exists a function that perfectly captures the satisfaction of a driver, particularly related to the routes they drive. To the best of our knowledge, such a function does not (yet) exist, if possible at all. Moreover, such a function would be dependent on the driver, of whom we do not have data related to how their satisfaction can best be modeled.

To interpret our numerical results, we link experience values to behavior in serving regions. This way, numerical values of experience let a decision maker get more insight in the underlying region consistency. To this end, links between numeric experience values for regions and their visit frequency are given in Table 6.1. The hyperparameters are set such that the numeric values given by learning and forgetting a region correspond to the visiting frequencies given in this table.

Table 6.1: Experience perception and numeric values

Experience perception	Visiting frequency	Experience value
Perfect	every day	1
Excellent	every other day	0.8
Fair	once a week	0.5
Poor	once a month	0.3
Non-existent	never	0

We need to limit the time horizon of our simulations, because of both runtime and the amount of data available. To get a good simulation of the experience, the hyperparameters are chosen in such a way that it is easier to jump from one experience level to the next. In other words, the heat of the experience is set high. The resulting hyperparameter values for the learning and forgetting curve are given in Table 6.2. Note that in contrast to Zhong, Hall, and Dessouky [63], we set the learn rate considerably higher than the forget rate as we consider driver satisfaction.

Table 6.2: Hyperparameters for the learning and forgetting curves

Hyperparameter	Value
E_1	0.3
E_{inf}	1
l	0.8
f	0.4

.
To get a feeling for the behaviour of the experience function given the common visit frequencies from Table 6.1, Figure 6.3 is presented below. These values allow drivers to learn very quick in our limited time horizon. Periodic behaviour that is interpretable is achieved over time.

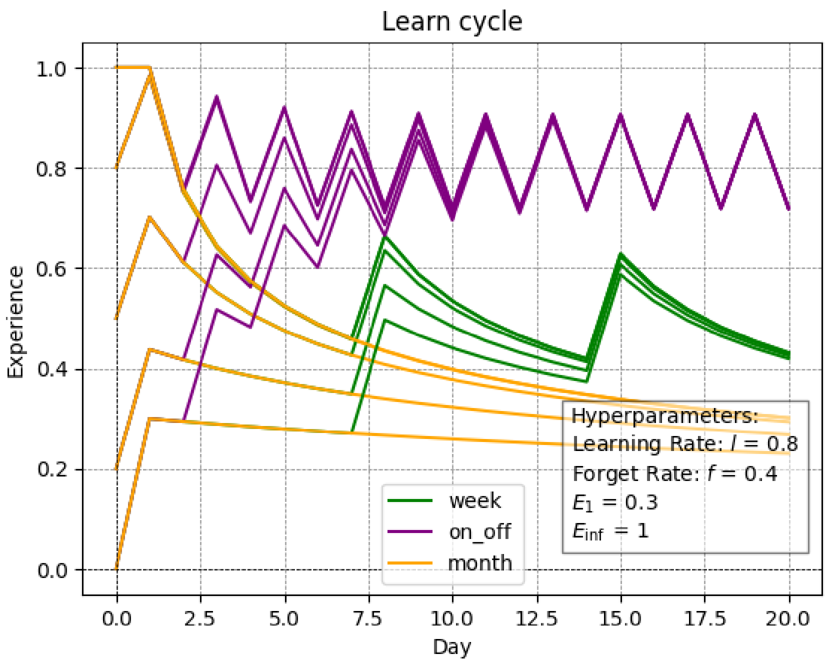


Figure 6.3: Experience curve under various visit frequencies. A region is served every other day, once a week and once a month. Initial values plotted are: 0, 0.2, 0.5, 0.8, 1.

Chapter 7

Case study

The data that is used in this research comes from one of the customers of ORTEC, which we will call company X. This company works in home delivery. The company has many different depots from which they serve their customers. Each customer is always delivered from a single depot.

In practice, not all routes for a given day are made at the same time. As convenience for the customers is important in home delivery, goods are often delivered on the next day or even on the same day as ordering. The loading of trucks is done at the latest moment in order to give the customers the most time to still order. The latest an order can still be made for a given round of delivery is called the cutoff point. A round of delivery is called a wave. The customers of company X are delivered in waves.

The cutoff point is assumed to be before the start of a wave. This means that when the routes are formed all information is available. Route optimization can take place between the cutoff point and the start of loading the trucks. A wave hence equals a single routing instance. In practice, companies can have two, four or even up to six waves per day. Although our data consists of waves, not full days, we will still call these waves days.
This section first goes over instance selection in Section 7.1. Subsequently, the instances data will be described and key characteristics are presented in Section 7.2. Lastly, Section 7.3 describes how artificial data is generated that complements the data that is available in our instances.

7.1 Instance selection

Every day, each depot has anywhere between four and twenty-four drivers working that day. To ease analysis, the data is split into small, medium and large instances. Small instances have anywhere between four and six drivers working on any day, medium instances have between eight and twelve drivers and large instances have between eighteen and twenty-four drivers. Every instance size contains 10 depots, which in turn each consist of 30 instances, all of the same size. Each day, the DSVRPTW has to be solved, resulting in a set of routes. Every instance considered contains a feasible solution and all customers that are scheduled to be served on a given day are served.

We do not have enough day-instances to arbitrarily choose them. We have to make a selection of which sizes (i.e. the number of drivers working) to use. To make sure every depot can be filled with thirty day-instances, the instances are not chosen uniformly over the number of drivers required for that instance. More data is available that require a number of drivers closer to the mean number of drivers required for the given depot size when considering small and medium instances. For large instances, probabilities depend on the number of day-instances (still) available. This also differs from depot to depot. The probabilities that are used for sampling are given in Table 7.1. The mean number of drivers used for every depot is always the same, given the depot size. For input sizes small, medium and large, the mean number of drivers used for a depot is respectively 5, 10 and 20.

7.2 Instance data

In Table 7.2 we give an overview of the most important characteristics of the instances used. Again, instances are split in small, medium and large sized instances. The mean, standard deviation, min and max are taken over all 30 instances over all 5 depots.
The mean number of customers per route statistic is first calculated per day. Out of these means, the mean, standard deviation, minimum and maximum are taken. This is done in this way, because routes have not been

Table 7.1: Probabilities that a day instance is used in a simulation based on the amount of drivers needed on that day.

Problem size	# drivers working on a day										
	4	5	6	...	8	9	10	11	12	...	18 –24
Small	0.25	0.5	0.25								
Medium					0.1	0.2	0.4	0.2	0.1		
Large											custom

made to give these key characteristics.

The capacity of vehicles is always the same for all 30 days for a particular depot with small instance sizes. In other words, vehicles are homogeneous with regards to capacity when considering single depots. For example, the depot vehicle capacity of 90 is the vehicle capacity of all 7 vehicles on all 30 days of the corresponding small depot. For any instance size, depot and day there is always enough capacity to form feasible solutions.

The number of customers per route does not change much over the instance sizes. We only see a slight increase over the instance sizes. Overall, the number of customers per route has low variance.
The time windows of these customers shorten a bit as the instance size increases. This does not give the full picture. The standard deviation of these time window lengths is large across all instance sizes. All instance sizes contain (a lot) of time windows that are only an hour long.
The load per customer and the service duration per customer are highly similar across all three instance sizes. The mean and standard deviation of these statistics are almost exactly the same across all three instance sizes.

Moreover, the cost attributes considered for routes are necessary to mention. An arbitrary route endures the following costs:

- For every kilometer driven: 2
- For every hour driven: 60

We will work with these values as these are the values also used by ORTEC for the given client.

7.3 Generation of artificial data

Drivers are assumed to work five days a week on average. This means that for small, medium and large instance sizes, the fleet sizes are 7, 14 and 28 respectively. The drivers that have to work on any given day are chosen at random. The drivers will work $(5/7) \cdot 30 = 21.43$ days on average. Some drivers will drive more days and some drivers will drive fewer days. First, the generation of workload preferences is presented in Subsection 7.3.1. After that, Subsection 7.3.2 presents the generations of regions, the initialization of region preferences and experience and gives some key characteristics of our instances related to the made regions.

7.3.1 Workload preference

We do not have any information about contract hours or work schedules of drivers, let alone preference hours. In practical settings, preference hours are related to contract hours, which in turn are related to route lengths. Therefore, we will base the generation of artificial data on the route lengths obtained when solving the VRPTWs without driver satisfaction. Typical route lengths of days can be seen in Table 7.3. We notice that the route lengths are typically shorter than 8 hours.

To do this, we first solve all thirty days used in a depot on routing costs. The lengths of all routes on all days are taken. Then, preference hours of drivers are taken from a normal distribution where the mean is the mean route duration of the made routes and the variance is the variance of the route duration of the made routes. Preference durations are not allowed to fall outside the range of the durations of the made routes. Preferences generated that do are clipped to the range of obtained route durations.

To create more consistency over the time horizon and to make the artificial data more realistic, we assume drivers' preference hours to be consistent over the weeks. Driver workload preferences are generated for seven days. These preferences are copied until 30 days are reached. For each depot, the same schedules are used over

Table 7.2: Key characteristics of used problem instances. Numbers are either rounded to whole numbers or to two decimal points.

SMALL				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Number of routes (drivers)	5	0.68	4	6
Load per customer	6.07	2.49	2	32
Number of customers	53.52	14.42	24	91
Mean number of customers per route	10.66	2.29	5.60	15.17
Capacity per vehicle	160	40.50	90	200
Time window length (s)	9672	2855	2700	27600
Service duration of customers (s)	540	190	120	2160
MEDIUM				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Number of routes (drivers)	10	1.11	8	12
Load per customer	6.09	2.51	2	65
Number of customers	128.61	22.09	83	186
Mean number of customers per route	12.85	1.55	8.64	16.91
Capacity per vehicle	141.27	38.82	90	200
Time window length (s)	9074	3740	2700	27600
Service duration of customers (s)	555	203	120	4440
LARGE				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Number of routes (drivers)	20	1.69	18	24
Load per customer	6.05	2.41	2	62
Number of customers	277.81	35.36	208	367
Mean number of customers per route	13.89	1.37	11.26	18.35
Capacity per vehicle	128.69	43.34	90	200
Time window length (s)	8506	4298	3240	24000
Service duration of customers (s)	559	205	180	4320

the different weight distributions in the scalarization method in order to be consistent and to be able to draw conclusions more easily.

Comparing route durations with the number of customers per route, we observe that their fraction is almost constant. This is to be expected.

7.3.2 Region consistency

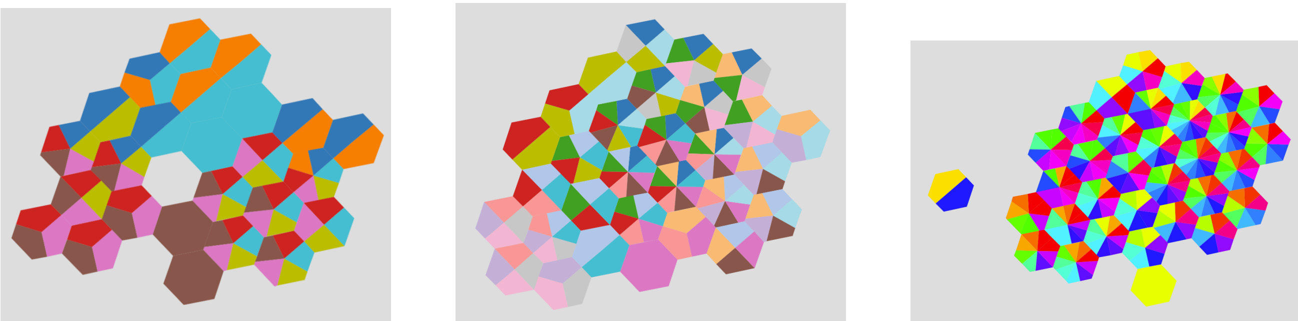
For region consistency we will initialize preferences and experience of drivers. They are both initialized as clusters of cells. It is realistic to assume that the driver preference areas are larger than single cells. The geographical compact clusters also grant flexibility in the order in which drivers deliver their customers, especially in the presence of time constraints [3].

We take the set of all possible regions that have at least one customer within the time horizon of 30 days. Out of all possible regions, two regions are chosen at random. These hexagons, along with the six hexagons surrounding this hexagon are the preference regions of this driver. The preference regions of drivers are also initialized with

Table 7.3: Route durations of used problem instances. The route durations are obtained when solely optimizing on the routing costs.

Instance size	Mean	Standard Deviation	Minimum	Maximum
SMALL	11712	3589	464	21233
MEDIUM	12910	4765	508	28195
LARGE	14577	5447	586	28797

full experience, having value 1. Here we assume that drivers that prefer to work in a region have already driven in that region a lot, resulting in the maximum experience. This procedure is followed no matter the instance size. Figure 7.1 shows what such an initialization can look like for the three different problem sizes. No map is drawn such that client data of ORTEC remains confidential. As mentioned earlier, the hexagons that we will use have size 6 in the H3 library, which corresponds to an area of $\approx 36.13\text{km}^2$.



(a) SMALL (b) MEDIUM (c) LARGE

Figure 7.1: Preference regions of drivers. Each color corresponds with a driver. Small instances have 7 drivers, medium 14 drivers and large 28 drivers. Shared preferences for preference regions are presented by splitting a hexagon among these drivers. These drivers still prefer the whole hexagon.

Table 7.4 gives an overview of the most important characteristics of the instances regarding region consistency.

Table 7.4: Key region consistency characteristics of used problem instances.

SMALL				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Customers per region	2.92	2.86	1	23
Regions per day	18.31	4.61	9	29
Region occurrence over 30 days (days)	15.78	10.35	1	30
Total number of distinct regions occurring	34.8	18.86	17	67
MEDIUM				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Customers per region	5.29	5.11	1	41
Regions per day	24.3	7.62	13	48
Region occurrence over 30 days (days)	20.48	10.54	1	30
Total number of distinct regions occurring	35.6	16.47	21	64
LARGE				
Statistic	Mean	Standard Deviation	Minimum	Maximum
Customers per region	6.48	7.89	1	74
Regions per day	42.85	22.30	17	87
Region occurrence over 30 days (days)	23.04	9.06	1	30
Total number of distinct regions occurring	55.8	31.39	26	104

The small and large instances have similar delivery area sizes as seen from the total number of distinct regions occurring. However, large instances cover a much larger area, just over 1.5 times as large. Although small and medium-sized instances cover similar area sizes, the medium-sized instances use more of this area on a daily basis. Regions occur on more days and the amount of regions occurring per day is larger for medium instances. In addition to medium instances using more of their delivery area on a daily basis, the density of customers in this area is also significantly larger for medium instances (5.29 customers per region) than for small instances (2.92 customers per region). All instance sizes have a lot of variety in the number of customers per region. This is in part caused by some regions having a lot of customers on certain days. We can classify these extremes (23 for small instances, 41 for medium instances and 74 for large instances) as outliers as they do not occur often.

Chapter 8

Results

This chapter presents the results. In Section 8.1 the Pareto plots obtained from Random Allocation, Driver Assignment and Integrated Approach are presented as well as KPI's describing the quality of solutions. The experience behavior will be investigated in more detail in Section 8.2. Finally, Section 8.3 treats region consistency in more detail.

8.1 Pareto fronts and KPI's

We begin by presenting the most important plot. Figure 8.1 shows the Pareto fronts of three representative depots of all three different problem sizes. The axes represent the normalized driver satisfaction measures 5.2 and 5.9. By normalization we mean taking the fourth root of each of the two methods to subvert the two squares. Note that after normalization by taking the fourth-root, the Pareto set stays the same, as taking the fourth root is a strictly monotone function.

The orange dot represents the Random Allocation method. Drivers are randomly assigned to predetermined routes. The blue dots represent the Driver Assignment approach. Driver Assignment is ran for 300 α/β-values that are evenly distributed on a log scale ranging from 10^{-35} to 10^{5}. Additionally, but proven superfluous, $(\alpha, \beta) = (1, 0)$ and $(\alpha, \beta) = (0, 1)$ are ran as well. The Pareto plot only plots the Pareto optimal solutions. Black dots represent the Integrated Approach. The (α, β)-values for which the Integrated Approach is ran are presented in Table 8.1. The integrated approach makes routes solving not only on routing costs, but also on the introduced driver satisafaction measures. Routes made are hence not optimal in routing costs. The percentwise routing cost increases of results of the integrated approach are plotted next to the black dots in figure 8.1.An overview of the Pareto fronts of all depots can be seen in Appendix A.1.

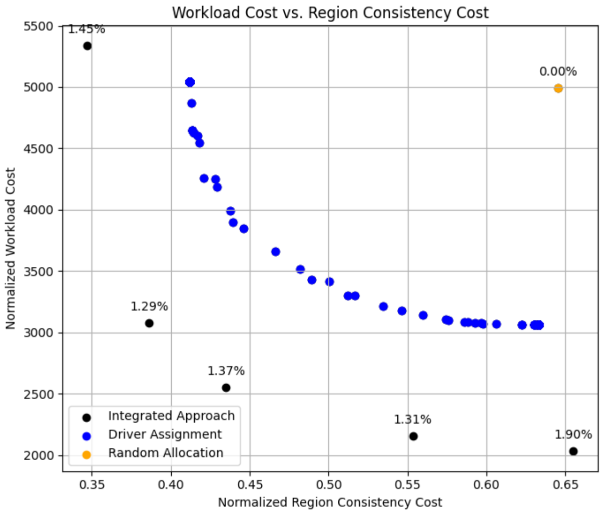
Table 8.1: (α, β)-pairs for which the integrated approach is ran.

Instance size	(α, β)-pairs ran
SMALL	$(0,0)$, $(5 \cdot 10^{-14}, 0)$, $(2 \cdot 10^{-14}, 2)$, $(5 \cdot 10^{-15}, 10)$, $(2 \cdot 10^{-15}, 20)$, $(5 \cdot 10^{-14}, 50)$
MEDIUM	$(0,0)$, $(2 \cdot 10^{-15}, 0)$, $(5 \cdot 10^{-15}, 2)$, $(2 \cdot 10^{-16}, 10)$, $(5 \cdot 10^{-17}, 20)$, $(5 \cdot 10^{-14}, 30)$
LARGE	$(0,0)$, $(5 \cdot 10^{-16}, 0)$, $(2 \cdot 10^{-16}, 0.5)$, $(5 \cdot 10^{-17}, 2)$, $(5 \cdot 10^{-14}, 5)$

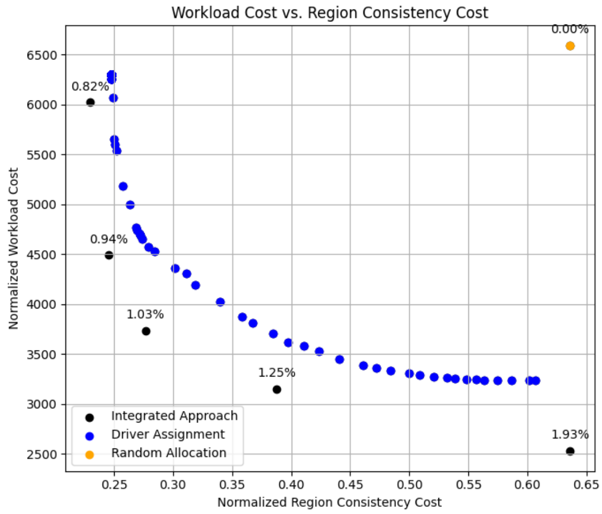
When plotting the Pareto curve, the drivers' preferences are initialized only once. This initialization is used for all the different weightings of the objectives. In other words, each dot in Figure 8.1 has the same initialization. In this way, the comparison between the different objectives is fair.

It is worth mentioning that the selected (α, β)-pairs do not necessarily give similar relative routing cost increases. The same (α, β)-pair could give relative routing cost increases of anywhere between 0.5% to 11% for small depots (where we also consider depots of company X that were not treated in this study). Most relative cost increases for small instances under the chosen (α, β)-pairs are between 1% and 3.5%.

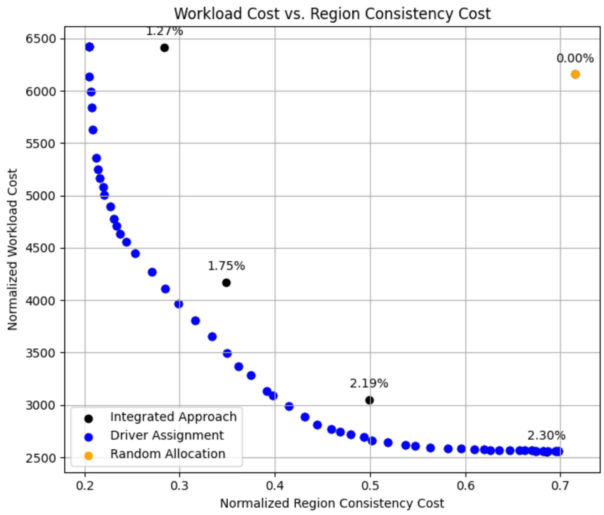
The Pareto figures look like squares. This is not exactly true. We could have focused even more on either of the driver satisfaction factors when focusing on only one of the two. This would lead to the other factor performing worse than the random assignment approach. The costs for such solutions would be even higher than the relative cost increases given in the Pareto plots. These relative cost increases are unrealistic in a practical setting.



(a) SMALL



(b) MEDIUM



(c) LARGE

Figure 8.1: Pareto plots of representative depots of all three problem sizes.

From these plots it is immediately clear that the integrated approach greatly outperforms Driver Assignment and Random Allocation for small instances considering cost increases of at most 2%. For medium instances, the integrated approach also outperforms the driver assignment approach, but margins are slimmer. It is up to a decision maker what cost increases are acceptable for what increase in driver satisfaction. In large instances, Driver Assignment greatly outperforms the integrated approach. Driver assignment does not only perform better on the driver satisfaction measures, but is does this without incurring additional costs and being much faster to solve than the integrated approach (cf. +- 2 seconds to +- 30 minutes). The integrated approach has potential to outperform driver assignment by, next to clever implementations, running the program for longer.

Table 8.2, Table 8.3 and Table 8.4 give KPIs for small, medium and large instances respectively. In these tables, 'WL deviation' is the deviation of realized workloads from preferred workloads. 'Driver exp' is the mean experience of a driver over all 30 days and all customers served. 'prv' and 'pcv' stand for 'preference region visits' and 'preference customer visits' respectively and are also taken over all 30 days. There is a small difference in total prv/pcv and driver prv/pcv: Total prv/pcv is taken over all regions/customers and drivers and days together. Driver prv/pcv takes for every driver the mean over all regions/customers and days. 'std' means standard deviation.
Now follows an example that illustrates the difference between prv and pcv. Suppose a driver drives for one day in 2 regions, one of which is their preference region and the other is not. The preference regions visit percentage (prv %) is 50%. If the preference region has 7 customers and the non-preference region has 3 customers, then the preference customer visit percentage (pcv %) is 30%.

As mentioned in Subsection 6.3.2, only the fraction of α and β defines the driver assignment settings, not their individual values. The α's and β's given in tables Table 8.2, Table 8.3 and Table 8.4 are chosen such that they align the most with the values generated for (α, β) as described in Subsection 6.3.2. When plotting these points in the same way as in Figure 8.1, they are roughly in the place one would expect; the blue points closest to the line drawn from the orange point to one of the black points.

Part of the normalized workload cost in Figure 8.1 is explained by being close to preferred workloads. The other part of this improvement is explained by fairness among drivers. For example, suppose we have two drivers, each having a route with a duration that is 50 minutes of their preferred duration. Summing the squares of both numbers gives 5000. Suppose, the deviations were 10 and 80 minutes, giving 6500 after summing the squares. The first scenario of both drivers deviating 50 minutes from their preferences is preferred as this adds to fairness, even though the total deviation from preferences is worse than in the (10, 80)-case. This is known as the power mean inequality. Spread in deviations from preferred workloads under multiple days and under multiple drivers adds to the workload measure ϕ^{wp}. This is the reason that the normalized workload costs are always larger than the workload deviation mean in Table 8.2, Table 8.3 and Table 8.4. The standard deviation of the workload deviations seems to scale quadratically with the workload deviation mean. Schedules with less workload deviation are therefore also more fair.

Table 8.2: KPIs of the Random Allocation method, the Driver Assignment method and the Integrated Approach method for various (α, β)-pairs. The KPI's are calculated on an aggregate of 5 solutions consisting of 30 days each. The mean values are taken of the results of the 5 depots.

SMALL											
KPI	Random Allocation	Driver Assignment					Integrated Approach				
		(α, β)									
		$(5 \cdot 10^{-14}, 0)$	$(2 \cdot 10^{-14}, 2)$	$(5 \cdot 10^{-15}, 10)$	$(2 \cdot 10^{-15}, 20)$	$(0, 50)$	$(5 \cdot 10^{-14}, 0)$	$(2 \cdot 10^{-14}, 2)$	$(5 \cdot 10^{-15}, 10)$	$(2 \cdot 10^{-15}, 20)$	$(0, 50)$
Routing cost ↑ %	0	0	0	0	0	0	2.479	1.618	1.433	1.574	1.833
WL deviation mean	3760	2198	2225	2481	2906	3902	1420	1576	1964	2328	3869
WL deviation std	797	357	361	386	425	943	147	165	185	267	854
Driver exp mean	0.435	0.444	0.470	0.543	0.584	0.614	0.446	0.499	0.584	0.611	0.644
Driver exp std	0.052	0.057	0.046	0.064	0.073	0.079	0.051	0.059	0.063	0.069	0.080
Total prv %	34.8	35.1	36.9	41.1	47.1	46.8	35.3	38.0	44.9	48.2	52.0
Driver prv % mean	34.8	35.4	37.1	41.5	48.0	47.8	35.4	38.3	45.6	49.1	52.9
Driver prv % std	7.6	6.7	6.8	6.8	8.9	8.9	7.8	6.5	7.9	7.9	9.1
Total pcv %	39.7	39.6	48.1	59.4	64.5	67.3	38.7	54.1	65.7	69.9	73.7
Driver pcv % mean	39.7	39.4	48.0	59.1	64.3	67.2	38.7	53.8	64.6	69.0	73.2
Driver pcv % std	15.3	16.0	15.9	13.8	13.2	13.9	16.6	15.7	13.3	11.6	11.6

Table 8.3: KPIs of the Random Allocation method, the Driver Assignment method and the Integrated Approach method for various (α, β)-pairs. The KPI's are calculated on an aggregate of 5 solutions consisting of 30 days each. The mean values are taken of the results of the 5 depots.

MEDIUM											
KPI	Random Allocation	Driver Assignment					Integrated Approach				
		(α, β)									
		$(2 \cdot 10^{-15}, 0)$	$(5 \cdot 10^{-16}, 2)$	$(2 \cdot 10^{-16}, 10)$	$(5 \cdot 10^{-17}, 20)$	$(0, 30)$	$(2 \cdot 10^{-15}, 0)$	$(5 \cdot 10^{-16}, 2)$	$(2 \cdot 10^{-16}, 10)$	$(5 \cdot 10^{-17}, 20)$	$(0, 30)$
Routing cost ↑ %	0	0	0	0	0	0	1.947	1.458	1.831	1.956	2.138
WL deviation mean	5128	2349	2857	3648	4470	5032	1966	2531	3146	4005	4744
WL deviation std	1015	305	381	565	937	1333	221	299	411	644	970
Driver exp mean	0.408	0.401	0.572	0.626	0.649	0.658	0.405	0.565	0.650	0.674	0.691
Driver exp std	0.052	0.047	0.063	0.071	0.077	0.078	0.047	0.053	0.059	0.067	0.070
Total prv %	32.7	32.8	46.2	51.7	52.5	53.2	32.2	44.6	52.8	56.2	58.1
Driver prv % mean	32.8	32.9	46.8	52.8	54.1	54.4	32.2	45.2	53.8	56.8	58.6
Driver prv % std	8.8	8.4	9.4	10.2	10.6	11.3	7.6	8.8	10.1	10.3	11.6
Total pcv %	35.9	33.9	67.4	72.9	74.5	75.3	33.9	66.6	77.2	82.4	83.3
Driver pcv % mean	35.8	34.2	67.8	73.4	75.3	75.9	34.5	67.0	77.7	82.3	83.6
Driver pcv % std	18.4	15.5	15.8	14.4	13.6	14.1	16.8	13.1	12.0	10.4	10.3

Table 8.4: KPIs of the Random Allocation method, the Driver Assignment method and the Integrated Approach method for various (α, β)-pairs. The KPI's are calculated on an aggregate of 5 solutions consisting of 30 days each. The mean values are taken of the results of the 5 depots.

LARGE									
KPI	Random Allocation	Driver Assignment				Integrated Approach			
		(α, β)							
		$(5 \cdot 10^{-16}, 0)$	$(2 \cdot 10^{-16}, 0.5)$	$(5 \cdot 10^{-17}, 2)$	$(0, 5)$	$(5 \cdot 10^{-16}, 0)$	$(2 \cdot 10^{-16}, 0.5)$	$(5 \cdot 10^{-17}, 2)$	$(0, 5)$
Routing cost ↑ %	0	0	0	0	0	3.026	2.805	2.493	2.396
WL deviation mean	5384	1826	2419	3530	5457	1997	2478	3477	5396
WL deviation std	926	261	328	691	1718	237	258	541	1313
Driver exp mean	0.330	0.335	0.537	0.604	0.657	0.341	0.440	0.550	0.615
Driver exp std	0.047	0.047	0.067	0.074	0.067	0.047	0.051	0.061	0.066
Total prv %	24.1	24.6	38.8	44.2	49.2	23.7	31.2	39.2	45.5
Driver prv % mean	24.1	24.7	39.4	45.1	50.0	23.7	31.8	39.4	46.2
Driver prv % std	7.9	8.7	9.5	11.2	13.4	8.7	9.5	9.8	10.0
Total pcv %	24.3	24.6	59.1	68.1	73.3	24.2	48.4	65.5	74.3
Driver pcv % mean	24.6	24.9	59.4	68.4	73.8	24.5	48.9	65.9	74.4
Driver pcv % std	15.2	14.7	17.1	15.4	14.6	15.3	17.3	14.9	13.5

Larger increases in routing costs are possible for the integrated approach. However, larger routing cost increases are unrealistic as no decision maker will choose to increase their travel costs with such high percentages. If we would increase routing cost percentages, then the integrated approach will perform better and will outperform the driver assignment, even in large instance sizes.

We see that the region consistency measure and the workload preference measure capture the KPI's well. Performing well on the region consistency (workload preference) measure translates into doing well on region consistency (workload preference) related KPI's and vice versa.

Interestingly, when considering Driver Assignment in small instances, the preference region percentage does not increase changing (α, β) from $(2 \cdot 10^{-15}, 20)$ to $(0, 1)$, i.e. focusing only on driver satisfaction. This suggests there are better strategies to optimize on region consistency. In particular, such strategies are non greedy, meaning that they will not optimize on region consistency on every single day, but will rather strategize a solution over 30 days to optimize region consistency over the entire 30-day period.

These results are even achieved in a setting in which workload preference was quite naively initialized. When obtaining and using workload preference of actual drivers, the preferred workloads will be more in line with the total travel time of days. Lower values for workload preference deviation can then be attained and we expect percentage-wise improvements in workload preference deviations to be even better.
The best possible workload deviation mean is not known. Workload preferences are initialized based on obtained route lengths when optimizing solely on routing costs. Although the mean of all workload preferences over all days for a single depot equals the mean of all route lengths over all days for a single depot, the means for specific days are not said to be equal. For Random Allocation and Driver Assignment, this means a workload deviation mean of 0 is highly likely to be unachievable, even if the workload preference measure is optimized over the 30 days. For the integrated approach, a workload deviation mean close to zero is theoretically possible if the total workload preference for a day is at least as big as the total realized workloads obtained for that day when solely optimizing on routing costs and if the weight on the workload preference measure is considerably higher. However, the resulting routes would not be desired as the routing costs would be much higher. The achieved reduction in workload deviation mean is therefore even better, considering that the best workload deviation mean is larger than 0.

Interestingly, the total preference customer visit percentage increases more than the total preference region visit percentage in both absolute value and relative value when comparing results that focus little on region consistency with results that focus much on region consistency. This means more customers are visited per preference region than per non-preference region. This happens because experience is rewarded per delivered customer, not per delivered region.

8.2 Experience behavior

Figure 8.2 gives the experiences of all drivers throughout the 30 day horizon for a representative small depot. Subfigure 8.2a is made only solving for routing costs. Subfigure 8.2b is made using the integrated approach with $(\alpha, \beta) = (0, 50)$. Immediately, we see an increase in average experience. For this depot, the average experience increased from 0.622 to 0.744. When using the integrated approach, regions get learned sooner than using the random allocation. Between both solution methods, there is no big difference in fairness regarding experience.
Drivers are initialized with preference regions. Their experience is initialized in the same way as their preferences. Indeed, Figure 7.1 can be interpreted as the preference regions of drivers, but also as the initialized experiences of drivers.

Over time, drivers learn and forget regions. The initial experience will diffuse into other regions. Figure 8.3 and Figure 8.4 gives the experiences of all drivers throughout the time horizon for a small depot.
The behavior of the evolutions of the drivers' experiences is clearly different when steering on our region consistency measure. When solely focusing on routing costs, experiences disperse over the complete service area. Most regions have most drivers have experience. Drivers do not necessarily drive in their preference regions and experience is gained arbitrarily.

When steering towards our region consistency measure, we see a clear pattern in the evolution of drivers' experiences. Clearly, drivers gain experience on only a selection of the service area. Still, experience is gained to some extend in a large portion of the delivery area, but this is necessary for several reasons. First, all customers have to be delivered. Second, routing costs cannot increase too much. And third, not all drivers work on all days. If customers have to be delivered on a day on which the driver with the most experience in the concerned regions does not work, then these customers have to be served by less experienced drivers.

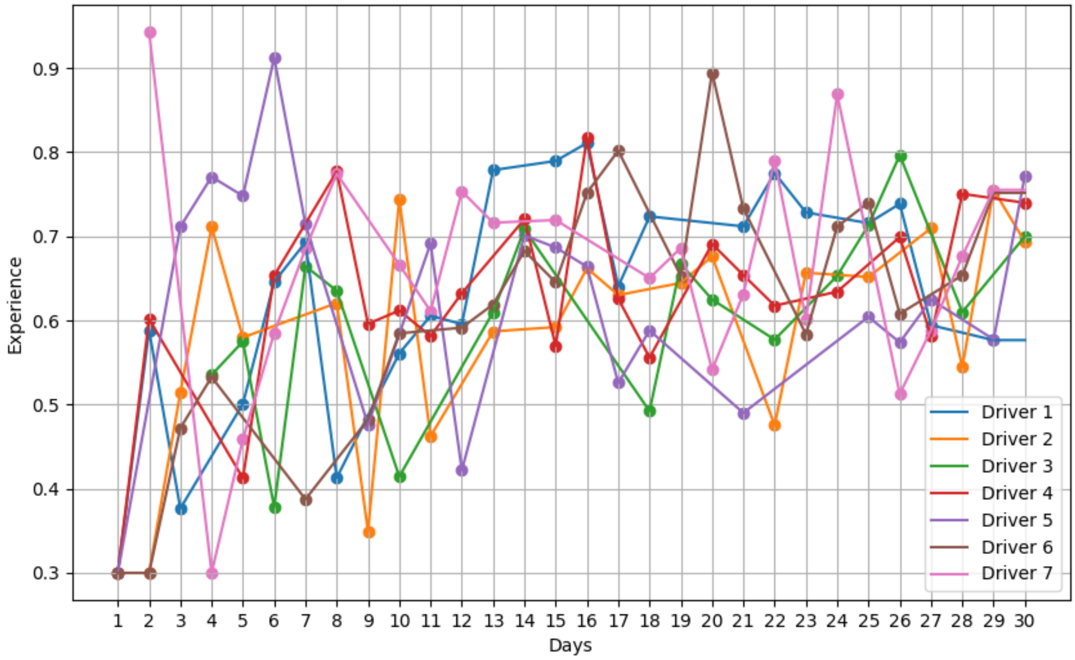
We see drivers mainly getting experience in their preference regions and the regions surrounding their preference regions. Besides the experience itself, this has another benefit. Consider the case in which two drivers have the same preferred regions. They cannot drive both in that same region. In our model, drivers have gained experience surrounding their preferred regions in an organic way. In the case of conflicting preference regions of drivers, routes are made that let drivers share their shared preference regions and that lay inside the regions that drivers have experience in surrounding their preferred regions.
In conventional, soft, assignments of drivers to regions, conflicts are also resolved by drivers driving part of their assigned regions and part of the regions surrounding them (e.g. [63]). In our model, driver satisfaction is considered. Drivers drive in the surrounding areas of their preference regions that they have most experience in. This gives more structure to the routes and even increases experience over time in a subset of the boundary regions of preferred regions. This brings more consistency and elevates driver satisfaction.

8.3 Region consistency

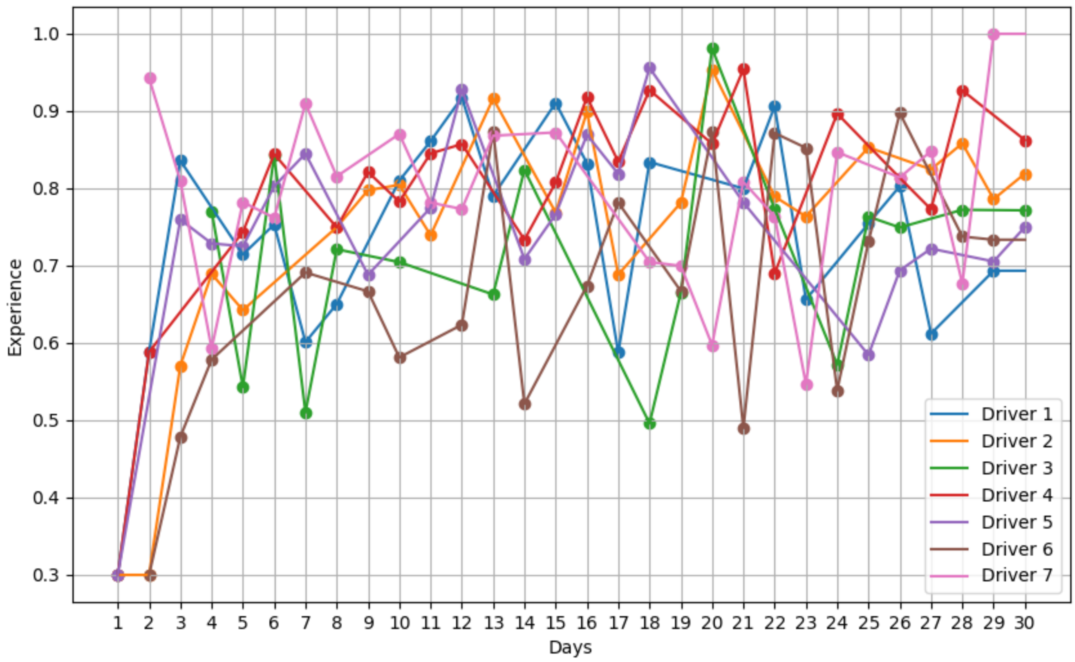
In the previous section we have investigated the experience of drivers. One can view experience as inherently adding to driver satisfaction. An increase in experience directly adds to driver satisfaction. Another interpretation of experience is experience being a proxy to help solve for region consistency.

Another reason we want to investigate region consistency in its own right is because an experience function is only an approximation of satisfaction felt by drivers. The experience function used does not accurately model satisfaction per se and is dependent on the chosen model and hyperparameters.
Region consistency in itself is a measure for experience. Region consistency does not depend on chosen models or chosen hyperparameters. By evaluating region consistency in itself, we can draw conclusions on experience while not depending on our experience model. This does not disregard the need for the experience model, as discussed in the methods chapter (Chapter 6).
In Figure 8.5 we present the region frequencies obtained via the random allocation and integrated approach ($(\alpha, \beta) = (0, c)$, where c depends on the instance size). Whenever a driver visits a region n times throughout the 30-day period, this visit multiplicity n is counted once. We call the total number of times any driver visits any region n times the frequency of this visit multiplicity. When considering a visit multiplicity of n with frequency f, the total number of visits to regions accounted for is $n \cdot f$. The visit multiplicities times the frequencies are plotted on the y-axis. The total area under the graph is equal to the total number of region visits throughout the 30-day period for all subfigures of Figure 8.5.

A good region consistency entails large visit multiplicities. Moreover, we would like large visit multiplicities to have a high frequency. In addition to region consistency in itself, we would like this consistency to be in preference regions as much as possible. For Figure 8.5, this translates into green areas in the right-most part of the histograms.
This is exactly what we see happening in Figure 8.5. We see more green in general, green in regions with a large visit multiplicity and large visit multiplicities in general. This all means preference regions are better respected when focussing on region consistency. Moreover, regions are visited more consistently, irrespective of the intepretation of the experience model.
The histogram plots of all other weightings considered can be seen in Figure B.1. The frequency histogram plots of the random allocation and the (α, β)-pair that focuses only on workload preference are highly similar. This is to be expected, given that the region consistency costs are similar for these two methods, as can be observed in Figure 8.1. All (α, β) configurations between the one focusing solely on workload preference and the one solely focusing on region consistency gradually transform the one into the other.
The more we focus on region consistency, the more regions are visited more frequently. Not only are regions visited more frequently, but these regions are also more often preference regions. Experience in regions served also increases. This can be seen in Figure 8.5.

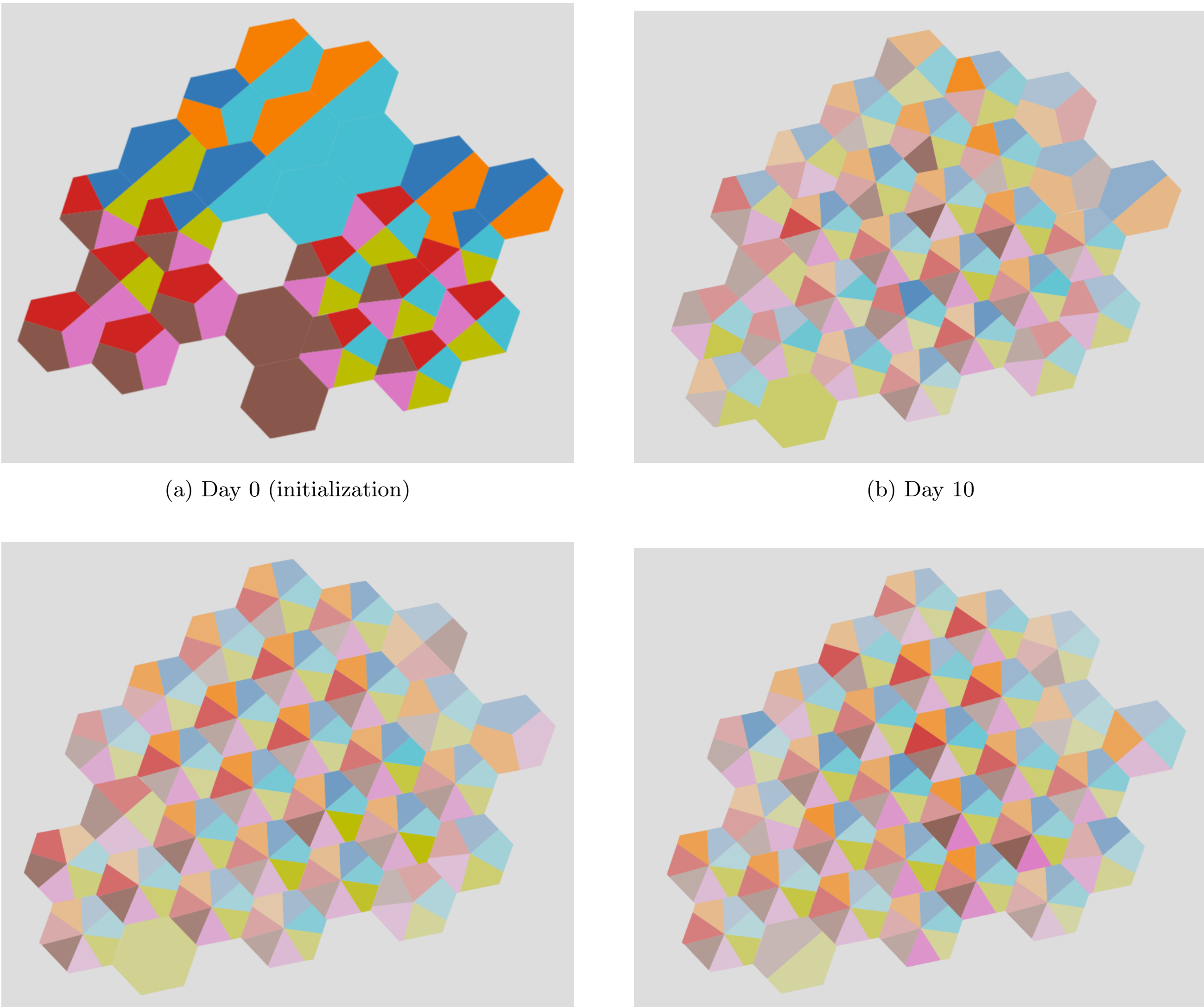


(a) SMALL, Random Allocation



(b) SMALL, Integrated Approach $(\alpha, \beta) = (0, 50)$

Figure 8.2: Mean driver experience over served regions per day over days. For every day a driver works, the mean experience value over all regions served at least once on that day is given. A pair of days that embrace a number of days off are connected via linear interpolation.



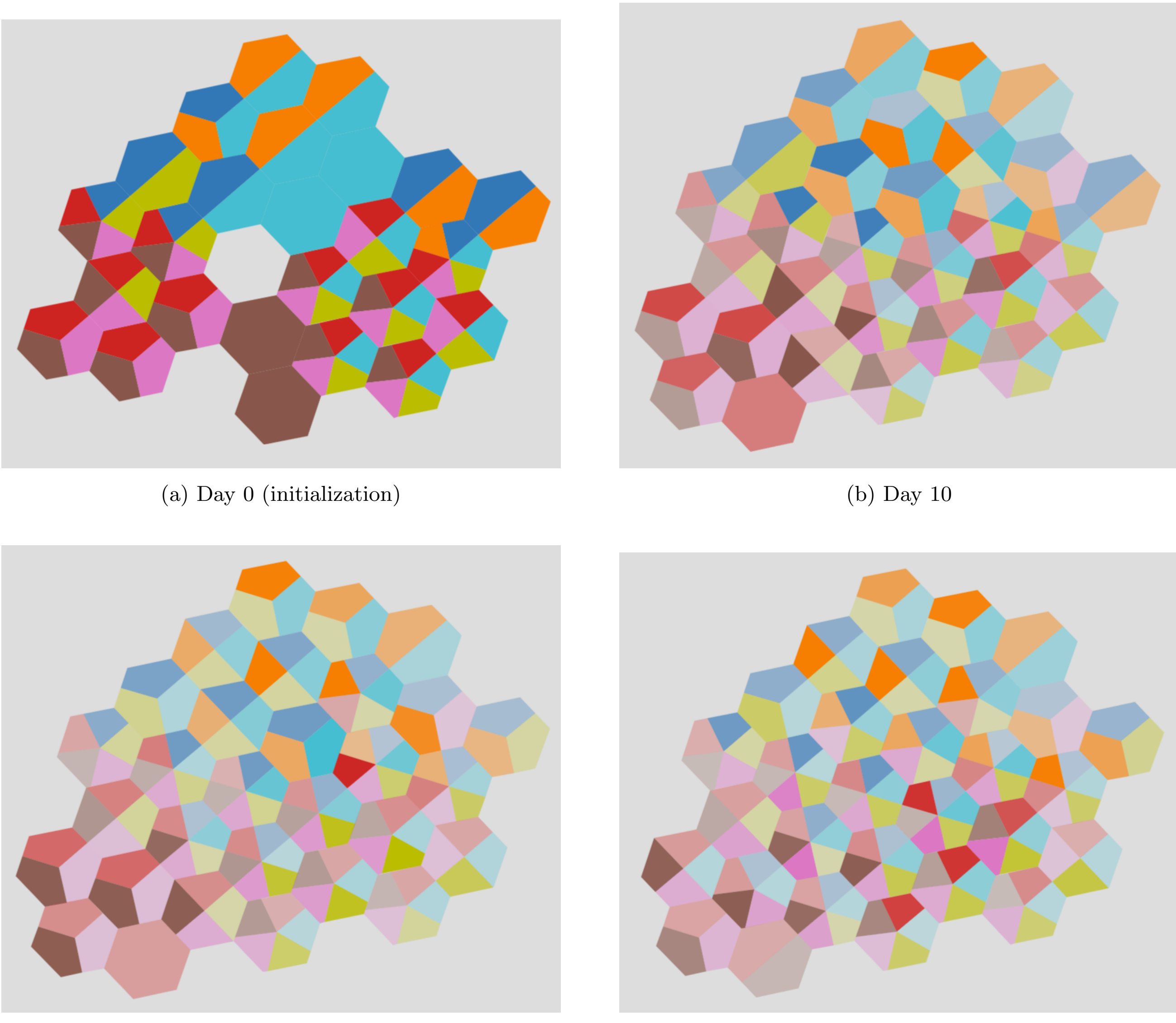
(a) Day 0 (initialization)

(b) Day 10

(c) Day 20

(d) Day 30

Figure 8.3: Experience behaviour of drivers throughout the 30 days. A single small depot is considered. Results are generated solely solving for routing costs. The opacity of regions is given by the experience of drivers. The more experience a driver has in a region, the more opaque that region. Some regions are added on later days. This is because no driver had experience in that region prior to that day.



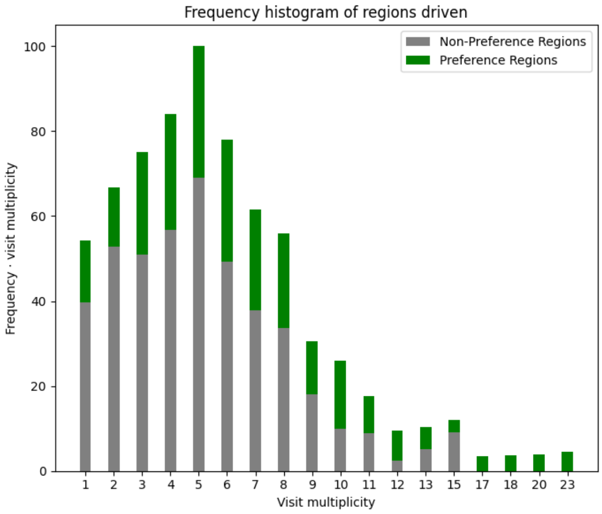
(a) Day 0 (initialization)

(b) Day 10

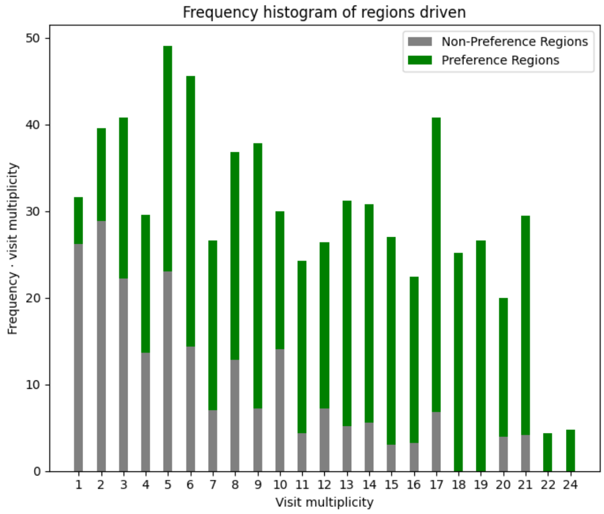
(c) Day 20

(d) Day 30

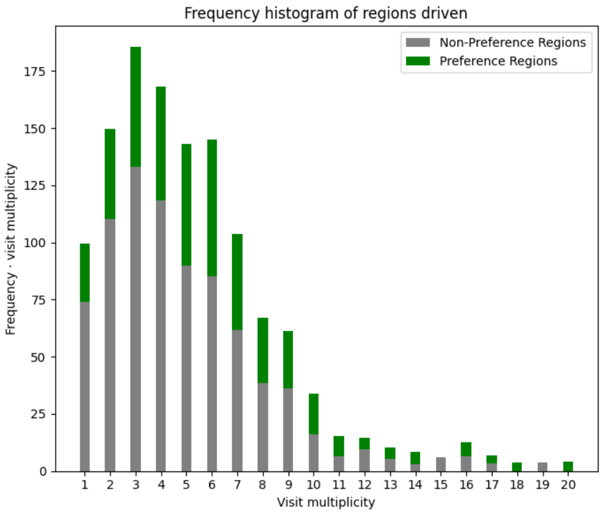
Figure 8.4: Experience behaviour of drivers throughout the 30 days. A single small depot is considered. Results are generated for $(\alpha, \beta) = (0, 50)$.



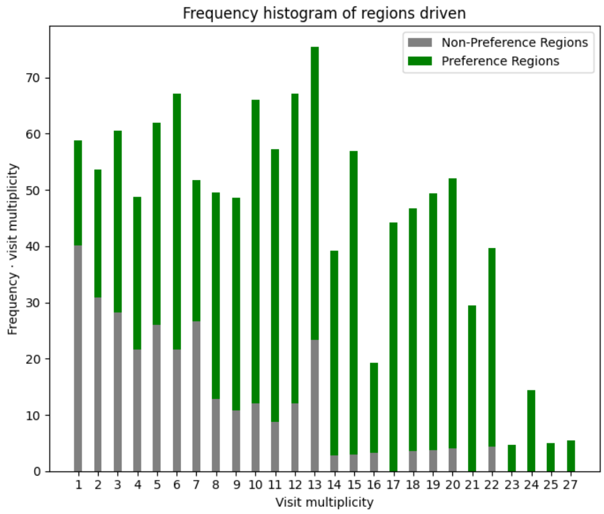
(a) SMALL, Random allocation



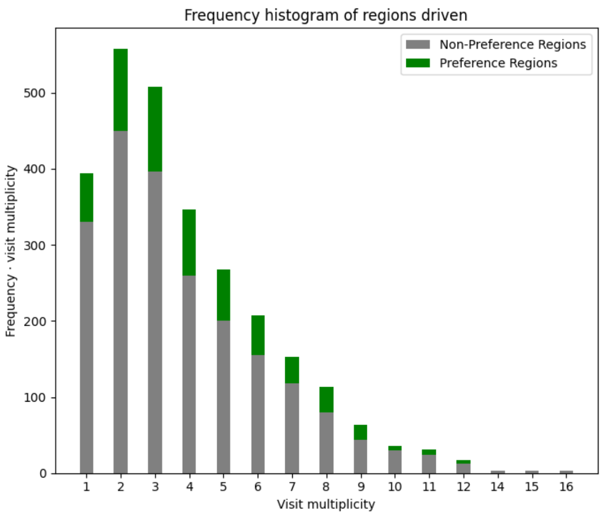
(b) SMALL, Integrated Approach $(\alpha, \beta) = (0, 50)$



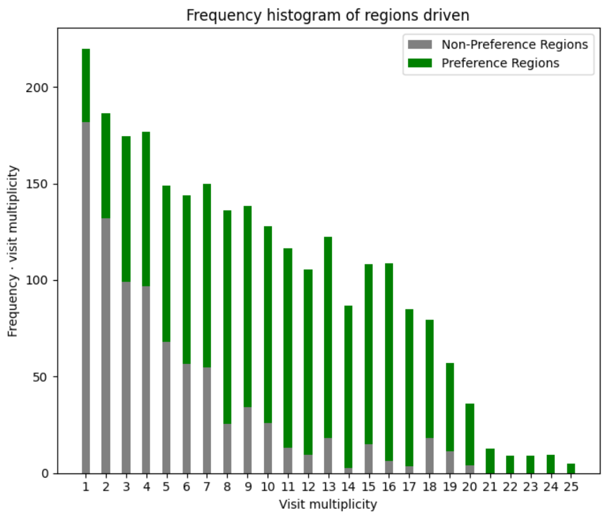
(c) MEDIUM, Random allocation



(d) MEDIUM, Integrated Approach $(\alpha, \beta) = (0, 30)$



(e) LARGE, Random allocation



(f) LARGE, Integrated Approach $(\alpha, \beta) = (0, 5)$

Figure 8.5: Frequency histograms of regions driven. Graphs are the mean of results of all 5 depots considered.

Chapter 9

Conclusion

The purpose of this thesis was to analyze how the driver satisfaction factor ‘region consistency’ and ‘workload preference’ could be modeled and optimized for while not increasing routing costs too much. We have incorporated both driver satisfaction factors in the PyVRP software developed by ORTEC. To do so, three methods were implemented, ran and analyzed: ‘Random Allocation’, ‘Driver Assignment’ and ‘Integrated Approach’. All three methods were used to solve problem instances of small, medium and large sizes.

Workload preference is modeled using preference workloads and quantified. Region consistency is modeled using preferred regions and using a log-linear experience function. The log-linear experience function not only rewards experience attained by drivers, but also serves as a proxy to get better region consistency. It does this by steering towards preference-neighbouring regions that a driver has gotten experience in in the event of conflicting preference regions.

Driver Assignment performs considerably better than random allocation. Driver Assignment either halves the mean workload deviation of drivers or comes close to doubling the preference region visit percentages and almost tripling the driver preference customer visit percentage for large instances. Balancing between workload preference and region consistency, Driver Assignment still greatly outperforms random allocation, almost halving the mean workload deviation and improving driver preference region visit percentages by 50% and improving preference customer visit percentages by a factor 2. The improvement obtained from using Driver Assignment instead of Random Allocation increases as the problem sizes increase.

Both the integrated approach and driver assignment achieve good solutions and are able to achieve fair routes. Integrated Approach outperforms Driver Assignment for small and medium sized instances on all different weightings of our workload preference and region consistency measures for a routing cost increase of less than 2%. The improvement obtained from using the integrated approach instead of Driver Assignment decreases as the problem sizes increase. For large instance sizes (20 drivers on average), Driver Assignment outperforms the integrated approach on all different weight settings if only routing cost increases of at most 3% are permitted. Larger increases in routing costs are possible, and the integrated approach would outperform Driver Assignment, but such large routing cost increases are unrealistic as no decision maker will choose to increase their travel costs with such high percentages.

Chapter 10

Recommendations

We state the following recommendations and ideas for further research.

- Regions are made arbitrarily in this study. We have discretized the delivery area and have assigned drivers preference regions. Research to what constitutes a preference region, what their shape, location and other characteristics are would be highly beneficial.
- Whenever scalars were chosen for region consistency and workload preference, these scalars were used throughout the whole time horizon. The scalars in the multi-objective can however be changed during the optimization process. Using this freedom, better solutions might be found.
- In our study, all drivers were known. In practice, drivers will come and leave the company. Researching modeling the introduction of a new driver in the system can be worthwhile as this makes the model more realistic and more suitable for practical implementation.
- Different additional heuristics to solve the DSVRPTW can be developed. For example, drivers could get a shadow preference that is large in regions that no driver has a preference for. Using such a shadow experience, an algorithm would be more likely to schedule this driver in that region. That driver gets experience in that region, which adds to driver satisfaction. If shadow preferences were not used, drivers would perhaps take turns driving that region and none of them is rewarded for having build experience in that region. This does neither add to region consistency nor driver satisfaction.
- Some papers have researched region consistency in the context of driver efficiency: Driving in the same area time after time makes drivers more efficient. This type of experience can be combined with the satisfaction gained from experience. Those two measures could support each other and amplify performance.
- Experience functions could be dependent on drivers and on regions (compare e.g. city vs rural areas). Drivers can also be categorized in groups that share the same experience function or initialization (e.g. junior vs senior drivers).
- Our chosen learning and forget hyperparameters are chosen such that regions can quickly be learned and forgotten. This is because we only consider a time horizon of 30 days, so a model with more heat was suitable. In more realistic modeling, drivers can get experience in regions over many months or even years. Further research can investigate whether the described behavior, results and conclusions still hold up in more long term settings. Additionally, a study towards experiences, or even satisfaction levels, gained by driving or not driving in regions would be of great use.
- Our chosen experience function is not the only experience function that can be used to model driver satisfaction via experience. Although it is almost impossible to do a study to the perceived satisfaction levels drivers get when getting experience in regions, research can be done on which experience function best fits driver satisfaction related to experience.
- When many routes are considered it is more difficult for the algorithm to schedule drivers in their preferred regions. It can be beneficial to make initial solutions based on the history. Moreover, steering the algorithm towards preference and experience regions by adding ghost customers in those regions, that will later be removed, can help the algorithm to find good solutions quicker.
- In the driver assignment method as well as in the integrated approach, optimization was performed based on current and historical data. It is reasonable to assume workload preferences are known in advance. Optimization can then be done by also taking future preferences into account. Incorporating future preferences in driver satisfaction optimization can be worth researching.

- As much driver satisfaction related input as possible should be taken from industry to make the model more realistic.
- In this thesis, only preferred regions were considered. In future research, disliked regions can also be considered.
- Using a measure that is independent from a time horizon is preferred. The measure that makes use of a convex combination over drivers could be preferred. Performance of the algorithm can be evaluated on the kpi's and not on the measure itself. It would be interesting to see the differences in kpi's obtained when using this measure over the one used in this thesis.
- We could bring more structure in which driver works when. Right now we have chosen the drivers that have to work arbitrarily for any given day. This could make the model more realistic and could potentially make the model perform better. Having drivers work more regularly over the days gives structure, which can benefit region consistency.
- The neighbourhoods in which is searched are not adjusted. Adapting such a neighbourhood could be computationally ineffecient. Estimators to adapt the neighbourhoods in a computationally cheap way could be researched. Nonetheless, neighbourhoods will keep relying heavily on distances and durations as these will still be the most important objectives.
- What is obviously very good to do is to take the driver assignment solutions and parse them as initial solutions to PyVRP. The costs as well as driver satisfaction factors are good for the driver assignment solutions. Especially in large instances this could save a lot of computation time searching for solutions that are good on all measures. The way this would work is as follows: First, all days are solved on routing costs. The first day will be solved via driver assignment to get the best objectives in the driver satisfaction factors. Then, the first day will be optimized in the integrated approach, using a couple of the driver assignment solutions that perform well on the given weighing of the driver satisfaction factors. The implicit driver assignment in the integrated approach is taken from the driver assignment of the driver assignment approach. The first day is then solved and performs at least as good as the driver assignment for day 1. Driver assignment for day 2 will be done, taking into account the history made by the integrated approach. After driver assignment is done for day i, the integrated approach is done with a couple of the driver assignment solutions of day i. This is done until 30 days are solved for.
  We note that this approach does not guarantee a better solution on 30 days than either of the driver assignment approach or integrated approach would. It is however very likely that both methods will be outperformed. Further research can investigate such a solution method.
- Rewarding experience helps when preferences of drivers are conflicting or when certain areas are preferred by no driver. It would be interesting to still let drivers choose their preference regions, but assign all regions that are not preferred to drivers in a dynamic way. This way experience is more forced. It does not matter that the region is forced upon a driver as no driver had that region as preference region.
- There have been depots outside of the scope of this study that had a considerably larger relative routing cost increase using the same driver satisfaction factor weights as other depots of the same size. This proves similar weights for the driver satisfaction factors do not yield similar results over different depots. Further research can investigate what characteristics of depots and their instances ask for what driver satisfaction factor weightings.

For more recommendations, see also the the thesis by Bruinink [5] and the paper by Kozyreff, Meerbergen, and Zobiri [31].

Bibliography

[1] *A brief introduction to HGS*. URL: https://pyvrp.org/setup/introduction_to_hgs.html (visited on 06/17/2024).

[2] Jerry Denver Allison. "Workload balancing in vehicle routing problems". PhD thesis. Oklahoma State University, 1986.

[3] Matthias Bender, Jörg Kalcsics, and Anne Meyer. "Districting for parcel delivery services – A two-Stage solution approach and a real-World case study". In: *Omega* 96 (Oct. 2020), p. 102283.

[4] Jürgen Branke et al. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-88908-3. URL: http://dx.doi.org/10.1007/978-3-540-88908-3.

[5] Tom Bruinink. *Beyond Efficiency: Exploring the Cost Effects of Prioritizing Driver Happiness in Vehicle Routing*. Master's thesis. Rotterdam, The Netherlands, Oct. 2023.

[6] Xi Chen, Barrett W. Thomas, and Mike Hewitt. "The technician routing problem with experience-based service times". In: *Omega* 61 (2016), pp. 49–61.

[7] Nicos Christofides. "Fixed routes and areas for delivery operations". In: *International Journal of Physical Distribution* 1.2 (Feb. 1971), pp. 87–92.

[8] Carlos A Coello Coello. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[9] Bob Costello and Alan Karickhoff. *Truck driver shortage analysis 2019*. Tech. rep. American Trucking Associations, July 2019.

[10] Parthasarati Dileepan. "Delivery planning problem". PhD thesis. University of Houston, 1984.

[11] Kathryn Dobie, James P Rakowski, and R Neil Southern. "Motor carrier road driver recruitment in a time of shortages: What are we doing now?" In: *Transportation Journal* 37.3 (1998), p. 5.

[12] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg, 2015.

[13] Flavio S. Fogliatto and Michel J. Anzanello. "Learning Curves: The State of the Art and Research Directions". In: *Learning Curves: Theory, Models, and Applications*. Ed. by Mohamad Y. Jaber. Boca Raton, FL: CRC Press, 2011, pp. 3–22.

[14] Kwaku Forkuo. "Shortage of Truck Drivers: The Genesis and Way Forward". In: *Journal of Traffic and Transportation Management* 4.2 (2023), pp. 49–54.

[15] *Global Truck Driver Shortage Report 2023*. Tech. rep. International Road Transport Union, Feb. 2024.

[16] Chris Groër, Bruce Golden, and Edward Wasil. "The Consistent Vehicle Routing Problem". In: *Manufacturing & service operations management* 4 (Oct. 2009), pp. 630–643.

[17] JL Heskett. "Putting the service-profit chain to work". In: *Harvard Business Review* (1994).

[18] Mohamad Y Jaber and Hemant V Kher. "Variant versus invariant time to total forgetting: the learn–forget curve model revisited". In: *Computers & Industrial Engineering* 46.4 (2004), pp. 697–705.

[19] Mohamad Y. Jaber. *Learning Curves: Theory, Models, and Applications*. CRC Press, 2011.

[20] Jochen Janssens et al. "Multi-objective microzone-based vehicle routing for courier companies: From tactical to operational planning". In: *European Journal of Operational Research* 242.1 (2015), pp. 222–231.

[21] Ahmad I. Jarrah and Jonathan F. Bard. "Large-scale pickup and delivery work area design". In: *Computers & Operations Research* 39.12 (2012), pp. 3102–3118.

[22] Brian Kallehauge. "Formulations and exact algorithms for the vehicle routing problem with time windows". In: *Computers & Operations Research* 35.7 (2008). Part Special Issue: Includes selected papers presented at the ECCO'04 European Conference on combinatorial Optimization, pp. 2307–2330.

[23] Brian Kallehauge et al. *Vehicle routing problem with time windows*. Springer, 2005.

[24] R. M. Karp. “Reducibility among combinatorial problems”. In: *Complexity of Computer Computations*. Ed. by R. E. Miller and J. W. Thatcher. New York: Plenum Press, 1972, pp. 85–103.

[25] C Yalçın Kaya and Helmut Maurer. “Optimization over the Pareto front of nonconvex multi-objective optimal control problems”. In: *Computational Optimization and Applications* 86.3 (2023), pp. 1247–1274.

[26] R. F. Kirby and J. J. McDonald. “The Savings Method for Vehicle Scheduling”. In: *Operational Research Quarterly (1970-1977)* 24.2 (June 1973), pp. 305–307.

[27] Wouter Kool et al. *Hybrid Genetic Search for the Vehicle Routing Problem with Time Windows: a High-Performance Implementation.* Tech. rep. 2022.

[28] Attila A Kovacs, Sophie N Parragh, and Richard F Hartl. “A template-based adaptive large neighborhood search for the consistent vehicle routing problem”. In: *Networks* 63.1 (2014), pp. 60–81.

[29] Attila A. Kovacs et al. “The Generalized Consistent Vehicle Routing Problem”. In: *Transportation Science* 49.4 (Nov. 2015), pp. 796–816.

[30] Attila A. Kovacs et al. “Vehicle routing problems in which consistency considerations are important: A survey”. In: *Networks* 64.3 (Oct. 2014), pp. 192–213.

[31] G. Kozyreff, K. Meerbergen, and F. Zobiri. *Driver happiness in e-commerce.* Tech. rep. Leuven: Mathematics for Industry, 2022.

[32] Marcel Kunkel and Michael Schwind. “Vehicle Routing with Driver Learning for Real World CEP Problems”. In: *2012 45th Hawaii International Conference on System Sciences.* Vol. 37. IEEE, Jan. 2012, pp. 1315–1322.

[33] Stephen A. LeMay, Zachary Williams, and Michael Carver. “A triadic view of truck driver satisfaction”. In: *Journal of Transportation Management* 21.2 (Oct. 2009), pp. 1–15.

[34] Jan Karel Lenstra and A.H.G. Rinnooy Kan. “Complexity of vehicle routing and scheduling problems”. In: *Networks* 11.2 (June 1981), pp. 221–227.

[35] Hernán Lespay and Karol Suchan. “A case study of consistent vehicle routing problem with time windows”. In: *International Transactions in Operational Research* 28.3 (Oct. 2020), pp. 1135–1163.

[36] Yifu Li et al. “Experience-based territory planning and driver assignment with predicted demand and driver present condition”. In: *Transportation Research Part E: Logistics and Transportation Review* 171 (Mar. 2023), p. 103036.

[37] Rodrigo Linfati, Fernando Yáñez-Concha, and John Willmer Escobar. “Mathematical Models for the Vehicle Routing Problem by Considering Balancing Load and Customer Compactness”. In: *Sustainability* 14.19 (Oct. 2022), p. 12937.

[38] Fei Liu et al. *Heuristics for Vehicle Routing Problem: A Survey and Recent Advances.* 2023. arXiv: `2303.04147 [cs.AI]`.

[39] Simona Mancini, Margaretha Gansterer, and Richard F. Hartl. “The collaborative consistent vehicle routing problem with workload balance”. In: *European Journal of Operational Research* 293.3 (2021), pp. 955–965.

[40] P. Matl, R.F. Hartl, and T. Vidal. “Workload equity in vehicle routing: The impact of alternative workload resources”. In: *Computers & Operations Research* 110 (2019), pp. 116–129.

[41] R. H. Mole. “A Survey of Local Delivery Vehicle Routing Methodology”. In: *Journal of the Operational Research Society* 30.3 (Mar. 1979), pp. 245–252.

[42] Yuichi Nagata and Shigenobu Kobayashi. “A Memetic Algorithm for the Pickup and Delivery Problem with Time Windows Using Selective Route Exchange Crossover”. In: *Parallel Problem Solving from Nature, PPSN XI.* Ed. by Robert Schaefer et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 536–545.

[43] Christopher J.D. Patten et al. “Driver experience and cognitive workload in different traffic environments”. In: *Accident Analysis & Prevention* 38.5 (2006), pp. 887–894.

[44] *PyVRP 0.10.0a0 documentation.* URL: `https://pyvrp.org/` (visited on 08/26/2024).

[45] Olivier Quirion-Blais and Lu Chen. “A case-based reasoning approach to solve the vehicle routing problem with time windows and drivers’ experience”. In: *Omega* 102 (2021), p. 102340.

[46] Bart van Rossum, Rui Chen, and Andrea Lodi. “Optimizing Fairness over Time with Homogeneous Workers”. In: *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).* Ed. by Daniele Frigioni and Philine Schiewe. Vol. 115. Open Access Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 17:1–17:6.

[47] Elm’aghraby Salah. “Economic manufacturing quantities under conditions of learning and forgetting (EMQ/LaF)”. In: *Production Planning & Control* 1.4 (1990), pp. 196–208.

[48] Michael Schneider et al. "Territory-Based Vehicle Routing in the Presence of Time-Window Constraints". In: *Transportation Science* 49.4 (Nov. 2015), pp. 732–751.

[49] Karen Smilowitz, Maciek Nowak, and Tingting Jiang. "Workforce Management in Periodic Delivery Operations". In: *Transportation Science* 47.2 (May 2013), pp. 214–230.

[50] Ilgaz Sungur et al. "A Model and Algorithm for the Courier Delivery Problem with Uncertainty". In: *Transportation Science* 44.2 (May 2010), pp. 193–205.

[51] Paolo Toth and Daniele Vigo. *Vehicle Routing*. Ed. by Daniele Vigo and Paolo Toth. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014.

[52] Man Yiu Tsang and Karmel S. Shehadeh. *A Unified Framework for Analyzing and Optimizing a Class of Convex Fairness Measures*. 2024. arXiv: `2211.13427 [math.OC]`.

[53] Thibaut Vidal. "Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood". In: *Computers & Operations Research* 140 (2022), p. 105643.

[54] Thibaut Vidal et al. "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows". In: *Computers & Operations Research* 40.1 (2013), pp. 475–489.

[55] C. D. J. Waters. "Interactive Vehicle Routeing". In: *Journal of the Operational Research Society* 35.9 (Sept. 1984), pp. 821–826.

[56] Min Wen. "Rich Vehicle Routing Problems and Applications". PhD thesis. DTU Management, 2010.

[57] Donnie F. Williams, Stephanie P. Thomas, and Sara Liao-Troth. "The Truck Driver Experience: Identifying Psychological Stressors from the Voice of the Driver". In: *Transportation Journal* 56.1 (Jan. 2017), pp. 54–76.

[58] Niels A. Wouda, Leon Lan, and Wouter Kool. "PyVRP: a high-performance VRP solver package". In: *INFORMS Journal on Computing* (2024).

[59] Theodore. P. Wright. "Factors Affecting the Cost of Airplanes". In: *Journal of the Aeronautical Sciences* 3.4 (Feb. 1936), pp. 122–128.

[60] Meng Yang, Yaodong Ni, and Qinyu Song. "Optimizing driver consistency in the vehicle routing problem under uncertain environment". In: *Transportation Research Part E: Logistics and Transportation Review* 164 (2022), p. 102785.

[61] Yu Yao et al. "The consistent vehicle routing problem considering path consistency in a road network". In: *Transportation Research Part B: Methodological* 153 (2021), pp. 21–44.

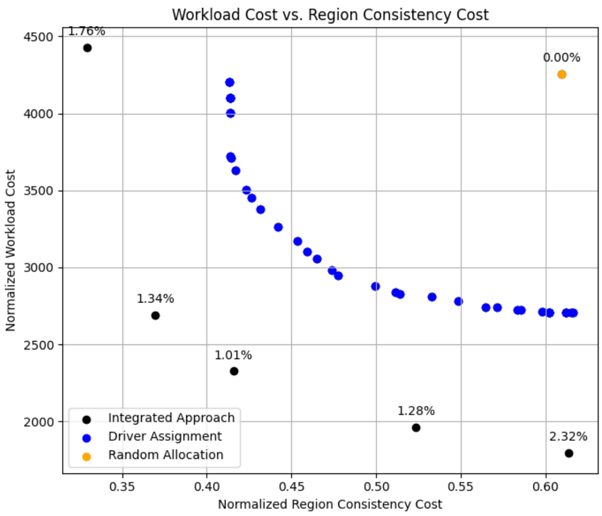
[62] Haifei Zhang et al. "Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms". In: *Archives of Computational Methods in Engineering* 29.1 (Jan. 2022), pp. 195–221.

[63] Hongsheng Zhong, Randolph W. Hall, and Maged Dessouky. "Territory Planning and Vehicle Dispatching with Driver Learning". In: *Transportation Science* 41.1 (Feb. 2007), pp. 74–89.

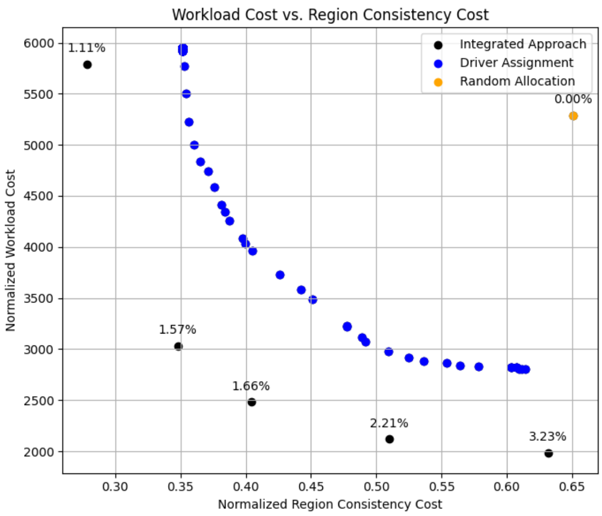
Appendices

Appendix A

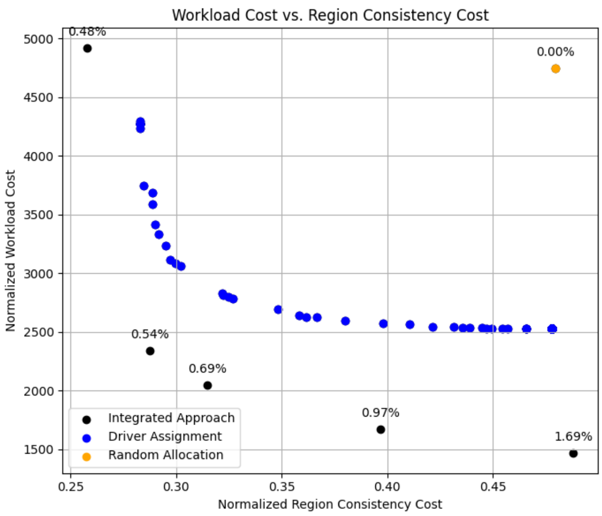
Pareto plots



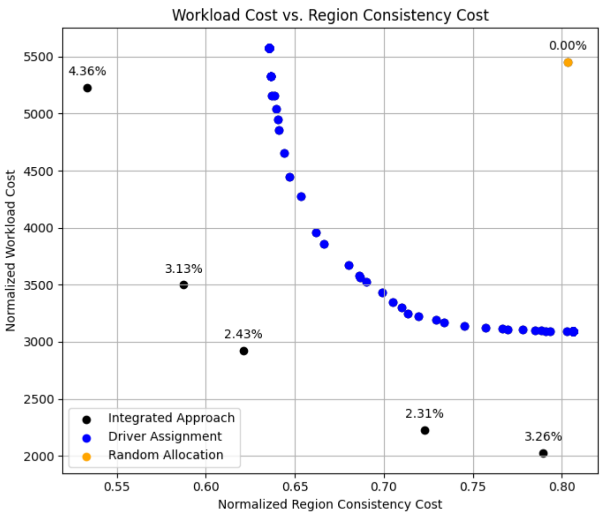
(a) SMALL depot 2



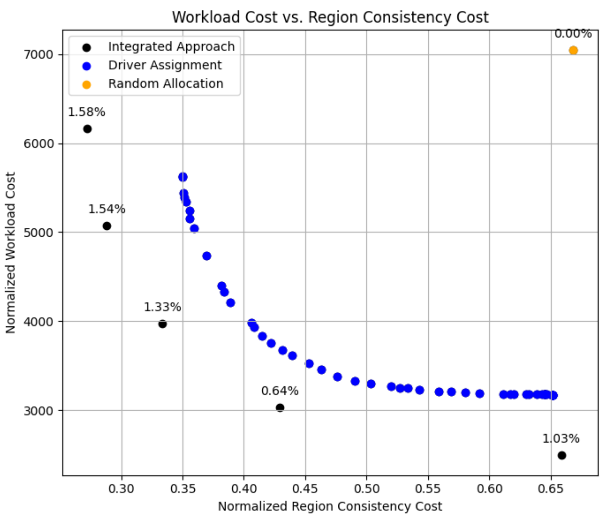
(b) SMALL depot 3



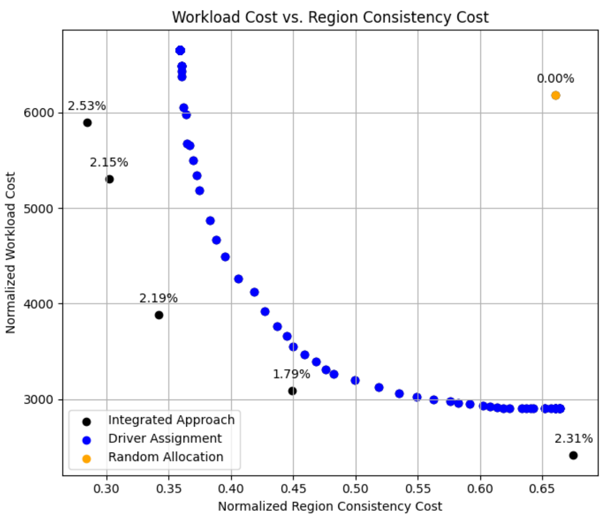
(c) SMALL depot 4



(d) SMALL depot 5

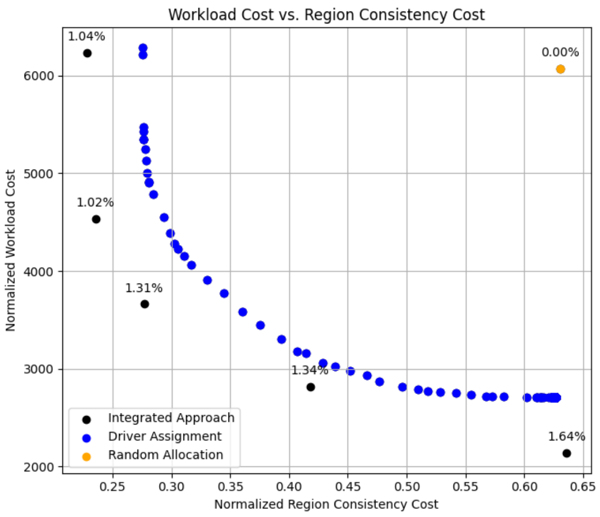


(e) MEDIUM depot 2

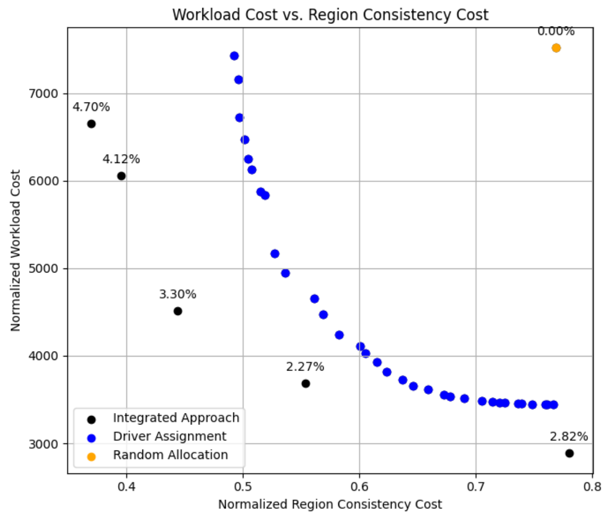


(f) MEDIUM depot 3

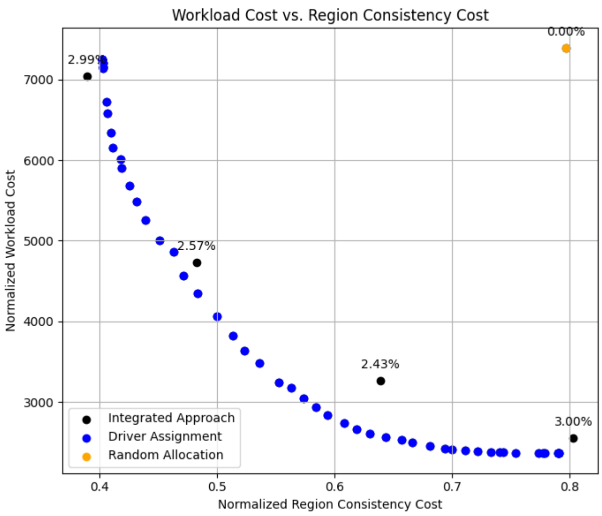
Figure A.1: Pareto plots of all other depots considered.



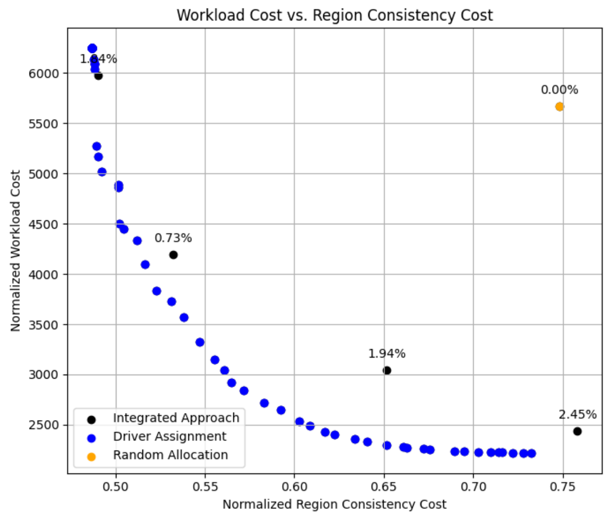
(g) MEDIUM depot 4



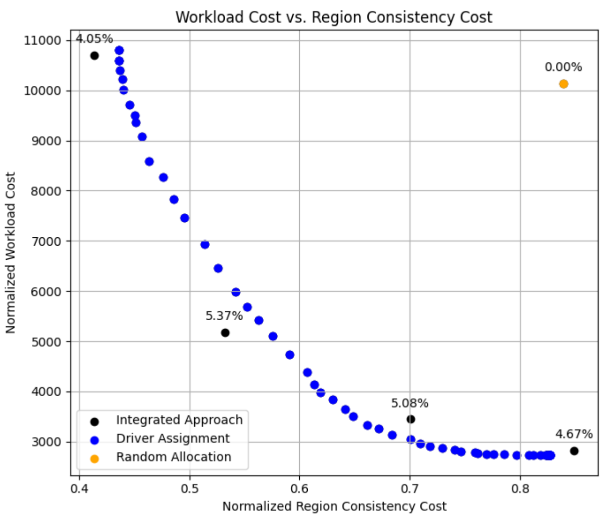
(h) MEDIUM depot 5



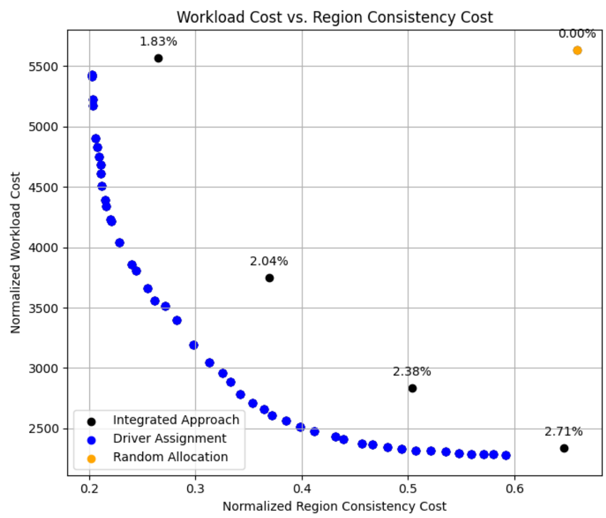
(i) LARGE depot 2



(j) LARGE depot 5



(k) LARGE depot 4

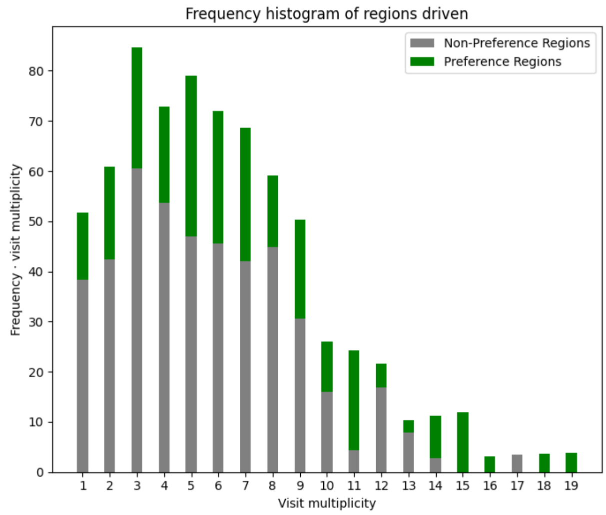


(l) LARGE depot 5

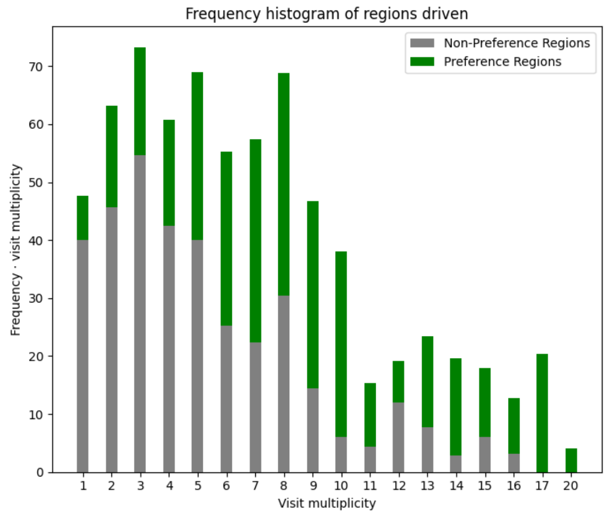
Figure A.1: Pareto plots of all other depots considered. (cont.)

Appendix B

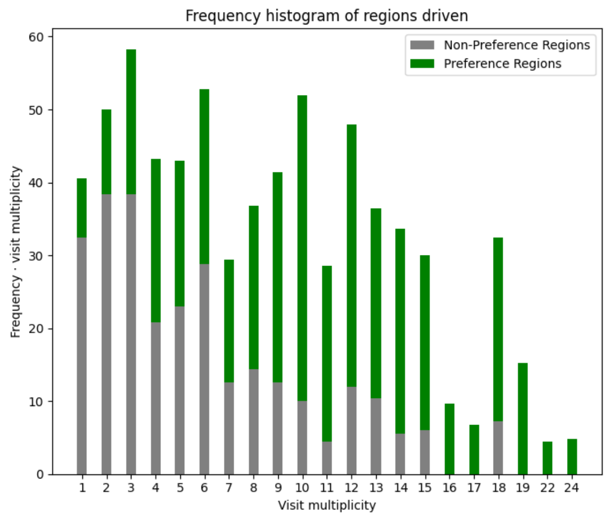
Frequency histograms



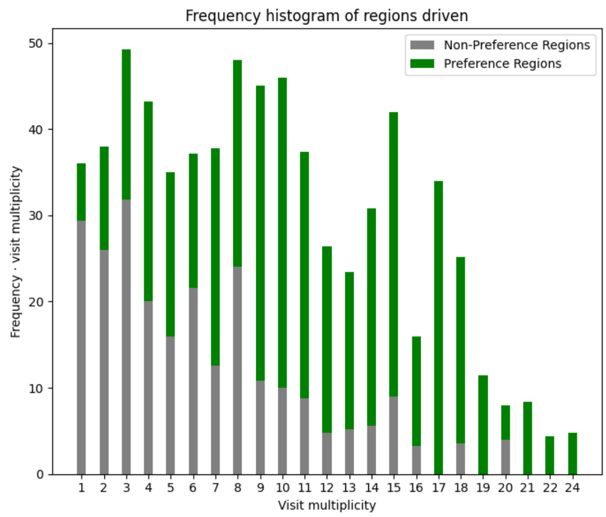
(a) SMALL, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-14}, 0)$



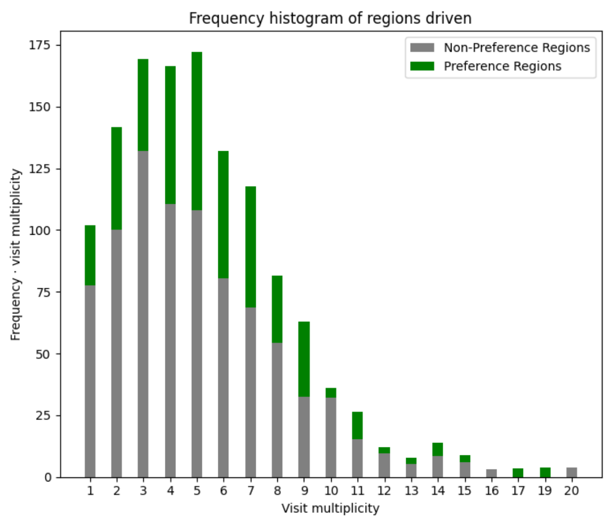
(b) SMALL, Integrated Approach $(\alpha, \beta) = (2 \cdot 10^{-14}, 2)$



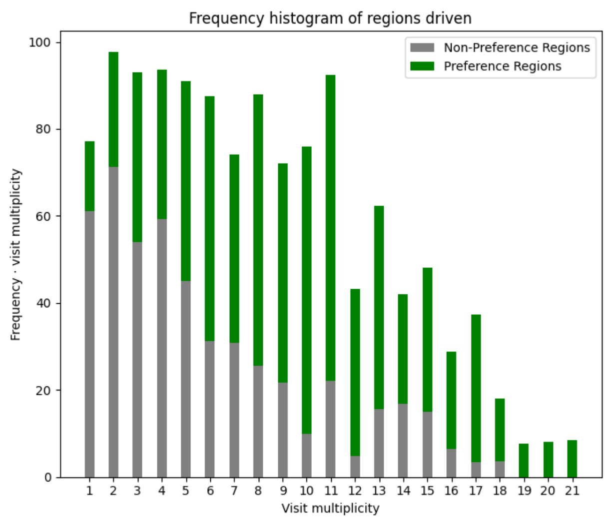
(c) SMALL, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-15}, 10)$



(d) SMALL, Integrated Approach $(\alpha, \beta) = (2 \cdot 10^{-15}, 20)$

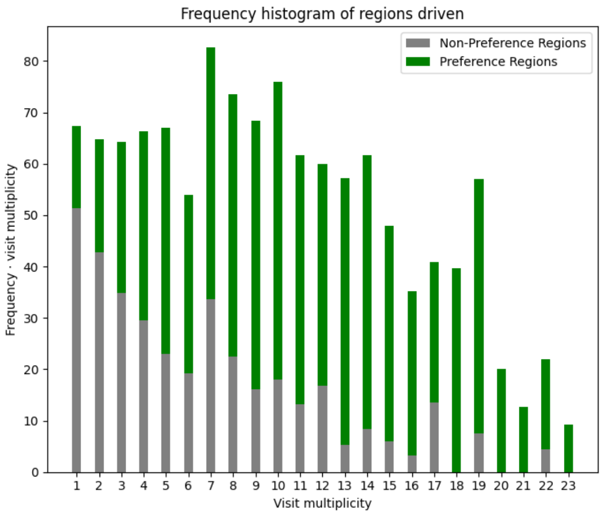


(e) MEDIUM, Integrated Approach $(\alpha, \beta) = (2 \cdot 10^{-15}, 0)$

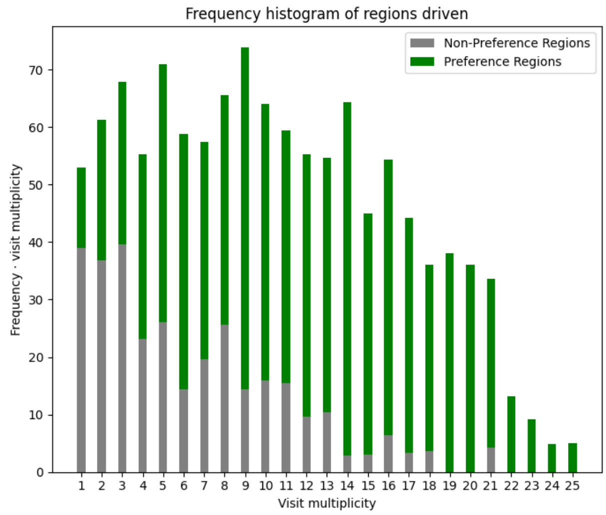


(f) MEDIUM, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-16}, 2)$

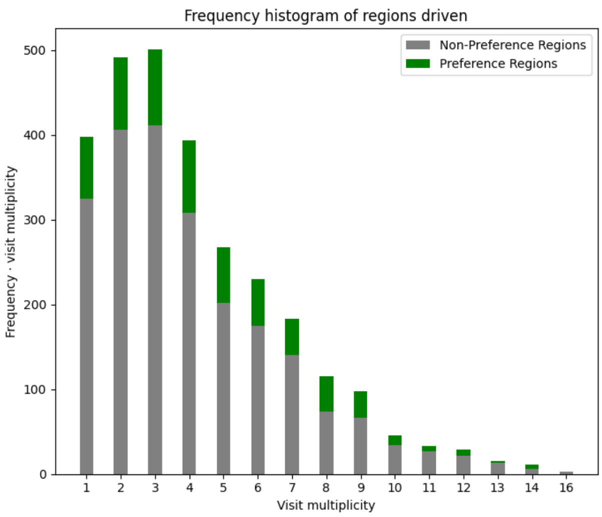
Figure B.1: Frequency histograms of regions driven for all other considered weightings. Graphs are the mean of results of all 5 depots considered.



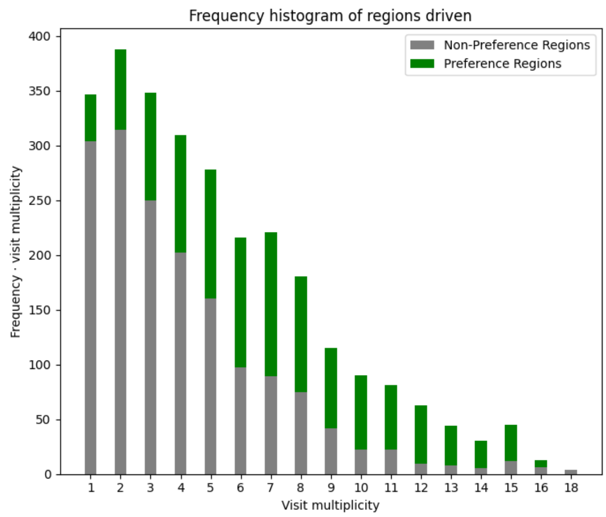
(g) MEDIUM, Integrated Approach $(\alpha, \beta) = (2 \cdot 10^{-16}, 10)$



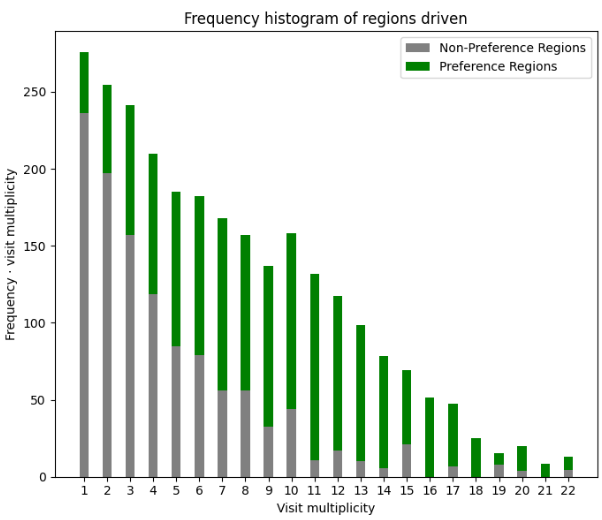
(h) MEDIUM, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-17}, 20)$



(i) LARGE, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-16}, 0)$



(j) LARGE, Integrated Approach $(\alpha, \beta) = (2 \cdot 10^{-16}, 0.5)$



(k) LARGE, Integrated Approach $(\alpha, \beta) = (5 \cdot 10^{-17}, 2)$

Figure B.1: Frequency histograms of regions driven for all other considered weightings. Graphs are the mean of results of all 5 depots considered. (cont.)