MSc Applied Mathematics

Track: Data Science & Optimization

Master thesis

Routing Predictions in the Home Health Care

by

Elham Wasei

February 15, 2023

Supervisor: dr. René Bekker

Assistant supervisor: Yoram Clapper, PhD candidate

Second examiner: dr. Eduard Belitser

Department of Mathematics
Faculty of Science

VU UNIVERSITY AMSTERDAM

Abstract

Motivated by routing and scheduling problems in the home health care, we have followed three research directions. The first and third can be related to planned, deterministic demand for health care, while the second can be related to stochastic demand. The first direction is an asymptotic analysis of TSPs and VRPs as the number of customers grows large, where we were interested in the convergence of the optimal tour length. Second, we were interested in the expected waiting time in the dynamic traveling repairman problem (DTRP) where customers are served in a FCFS manner. The DTRP with FCFS is related to the M/G/1 queue but with a spatial arrival process, so that travel times between successive customers are involved, which cause total service times to have a dependency structure. We found the ordinary M/G/1 queue to be an inaccurate model for this, based on a simulation study. Third, we considered the Home Health Care Routing and Scheduling Problem. For this problem we built a machine learning model to predict cost measures of optimal routing and scheduling solutions. The model has been trained using about a hundred features of various kinds, such as distance related features, spatio-temporal features and more. Its performance is reasonable for practical purposes.

Title: Routing Predictions in the Home Health Care
Author: Elham Wasei
Supervisor: dr. René Bekker
Assistant supervisor: Yoram Clapper, PhD candidate
Second examiner: dr. Eduard Belitser
Date: February 15, 2023

Department of Mathematics
VU University Amsterdam
de Boelelaan 1081, 1081 HV Amsterdam
`http://www.math.vu.nl/`

Popular summary

For people with disabilities, such as the elderly, health care must be provided more at their homes, due to recent health care reforms. And due to an aging population and workforce shortages, there is a need for efficient delivery of such home care. Home care can be scheduled or unscheduled. With scheduled care, health care workers need a plan, at the start of their workday, for what kind of route to follow and in what order the clients should be visited and at what time. The difficulty in solving this routing and scheduling problem (RSP) lies in the fact that clients have a time preference and employees work in shifts, not the whole day. It was already known how to create solution routes and schedules for such RSPs, but what was of interest to us is to predict the costs (such as travel time) associated with such solutions, without knowing the solution. We used machine learning (in particular, a Random Forest Regressor, RFR) on a dataset of simulated cities, in which the location of clients, their time preference, their service duration, and the workers' shifts were known, including the (approximately) optimal solutions and the corresponding costs. With the RFR, we can reasonably predict the costs for a new city that was not in the dataset.

For a second research direction, we were interested in unscheduled care in a more abstract setting. Here, the demand for care appears randomly throughout the day at random locations (but within a fixed region). We considered a simplified version of the problem where there is one worker with an infinitely long shift, working an infinitely long day. The times between appearances/arrivals of clients with demand are random, but the type of randomness is known. By performing a computer simulation, we found that the average time that a customer has to wait, counting from the arrival moment, differed from what was stated in previous literature.

There was a third research direction, in which we again considered a simple routing problem, related again to scheduled care. If you need to visit a number of clients who are located inside some fixed region, and you need to return to your starting location, what happens to the length of the optimal (shortest) tour if you increase the number of clients? The same question was considered when multiple travelers are available, which allows for a division of labor. We found that the length of the optimal tour converged to $c\sqrt{An}$ for both problems, provided that in the second problem the number of possible starting locations and the number of travelers is relatively small compared to n, where n is the number of clients, A is the area of the region, and c is a constant, whose value is the same for both problems.

Contents

List of Figures

1 Introduction

Home health care is a type of health care that is provided at the patient's home, rather than in a hospital or other medical facility. It is an important aspect of health care in the Netherlands. This type of care is provided to patients who cannot adequately care for themselves or who need temporary or permanent assistance with their daily activities. The goal of home care is to enable patients to live as independently as possible in their own environment while supporting their health and well-being.

In the Netherlands, and the EU in general, the health care for the elderly and other people with disabilities is under increasing pressure from an affordability and accessibility perspective, and this pressure will only increase in the future. This is a multifaceted problem. First, the EU has to face the problem of an aging population, which has been a long trend and it is only going to continue in the coming decades. This, of course, will only increase the demand for (home) health care. To give an idea of the size of this increase, the proportion of 80+ year olds in the EU is expected to increase two and a half fold from 6.0% to 14.6%, between 2021 and 2100 [1]. Second, not only does this increase the demand, but it also reduces the supply of health care professionals in a relative sense, as the number of working-age people relative to the number of elderly people declines. Third, and this applies to the Netherlands recently, the home care sector has seen a continual departure of professionals toward other sectors, and employing new personnel has been difficult. Due to this pressure, the rate of sick leave has increased lately. In the media, there have even been reports in the media of elderly patients who have been declined some important home care services because health care providers cannot fully handle the workload.

To address these challenges and to ensure that health care for this target group remains affordable, a major health care reform law was enacted in 2015 ("Wet langdurige zorg"). The goals of this law were fourfold: to have less costs, to enable the elderly to care for themselves longer, to have less residential care and to reform non-residential care (i.e., home care) [2]. Still, there is a need for health care organizations to deliver this service to the patient population in a more efficient way, and data-driven optimization can play an important role here. Proper coordination and timing of home care services can not only reduce costs, but timely care can greatly improve the quality of life of those in need of care.

This coordination and timing of the provision of care can also be called capacity planning. Currently, most health care organizations do capacity planning by hand, and data-driven decision-making is not involved. To perform capacity planning algorithmically, we can consider this problem to have the following phases (see Moeke & Bekker, 2020 [3] for more information on this in the context of nursing home care).

1. Predicting the workload
2. Finding the best shift pattern
3. Planning the routes and tasks

In the first phase, we want to know, on a particular day, how much work needs to be done in order to know how many health care workers (and of which qualification level) are needed at each point in time during the day. It is important to note that there can be deterministic demand and stochastic demand, and in reality the demand can be a mix of both, although we do not consider such problems where there is a mix. For the deterministic demand, we know how many patients need care, the time each patient prefers to receive care (given by a time window), the duration of care for each patient (service time), what type of qualification is needed for the home care worker, and of course the locations of patients (and the travel time between locations). There are no advanced methods to take all of this information into account to compute the workload and subsequently determine the required number of staff, beyond simple heuristic methods. Some demand can be stochastic if patients have the ability to call the organization and request a worker to visit them as soon as possible.

In the second phase, we want to determine the appropriate start times and end times of different (types of) shifts, as well as the number of workers needed to take each of those shifts, subject to capacity constraints. This can be done based on the workload prediction in phase 1.

In the third phase, we need to assign a subset of the patients to each worker, and for each worker we determine in which order the patients are going to be visited. By doing this, we also determine when all health care activities start and end. The optimal solution minimizes a weighted sum of three terms: the total (cumulative) travel time of all workers combined, the total shift overtime and the total waiting time of all patients, which accumulates when patients have to wait beyond their preferred time window.

Phases 2 and 3 do not need to be handled sequentially. This is because although the choice of shift pattern influences the routing and timing of health care activities, the choice of the appropriate timing of care activities will also influence the chosen shift pattern. Thus, it can be argued that these two phases should be executed simultaneously. When doing this, these two phases

combined can also be called the Home Health Care Routing and Scheduling Problem (HHCRSP), but this name is used more often for phase 3. This problem is NP-hard and cannot be solved to optimality for large instances. Nevertheless, recent advances have made large instances tractable, and solutions that are close to optimal can now be computed. One example is the novel paGOMEA algorithm (Clapper et al. (2023) [4]), which is an evolutionary algorithm. More about this in Chapter 7.

Our contribution is not to advance the methods that address the challenges of phases 2 and 3. Instead, the research in this master project is done with phase 1 in mind, although we indirectly tackle that problem. Phase 1 is vaguely defined, and in fact, the workload depends on the solutions of phases 2 and 3. However, we want to address phase 1 without having to solve phases 2 and 3. In particular, we are interested in predicting the total travel time, waiting time, and shift overtime (i.e. costs) that correspond to the approximate optimal solution (according to paGOMEA) of a particular instance in the HHCRSP, without having to solve the instance using paGOMEA. This is the primary research problem, along with two other problems, described in the next two paragraphs. The instances will have purely deterministic demand. We will use a machine learning model to make these predictions, and essentially, this model will have as input the information gathered in phases 1 and 2. Namely, given an instance, a rough estimate of the workload is used to fix a shift pattern, and with these inputs we compute some features that we then use to make predictions. With this capability, we can compare different shift patterns by running them through the ML model, and in this way we can determine the required number of workers that are needed to have an acceptable amount of costs. That is, we are then determining the workload, so this ML model can be used to solve phase 1. But the ML model can also be used to make decisions on a strategic/tactical level in general, by querying the model about the impact of small alterations in an instance.

In Chapter 5, we investigate the so-called dynamic traveling repairman problem (DTRP), which is a somewhat separate research direction, but it relates to home health care as follows. This problem is related to the setting of purely stochastic demand for home care, where requests for service appear randomly in time as patients call the health care organization for some type of help (e.g. emergency help) that was not planned. This is useful not only to better understand routing problems in home care settings where there is stochastic demand, but it is also useful for our ML model, in which we consider purely deterministic demand. We will study some related queueing theory in Chapter 4 (Preliminaries) and together with this queueing theory (in particular, some theory about the M/M/s and G/G/s queues), we will use the DTRP as a simple model for the HHCRSP, in order to get an estimate of the total waiting time and use this as a feature in our ML model.

In Chapter 6, we mathematically analyze a problem that is also not directly

related to the home health care setting, but the analysis will be useful in the ML model. Here, we analyze asymptotic behavior in the traveling salesman problem (TSP) and the vehicle routing problem (VRP) in the sense that the number of locations goes to infinity. For the TSP, we study the convergence of the length of optimal tour as the number of cities goes to infinity. For the VRP, we study the convergence of the total length of the optimal set of tours that start and end at a depot. Although this is a separate research direction, this information can provide a useful feature in our ML model. This is because the TSP can be used as a very simple model for the HHCRSP, for predicting the total travel time especially, by neglecting all the details (such as time windows, service times, etc.) of instances in the HHCRSP except the travel times. The same holds for the VRP, which is a slightly more complicated model, though closer to the HHCRSP. In fact, when the time windows are easy to satisfy, and the shift start times and end times are somewhat unconstrained, solutions to such instances will more closely resemble VRP solutions, as minimizing travel time will be the deciding factor.

In Chapter 7 we introduce the ML model and explain how we generate data to train the ML model and what features we extract from the HHCRSP instances. Furthermore, we will show the performance of the ML model.

2 Literature review

In this chapter, we provide some related literature to the DTRP (Chapter 5), and ML models for approximating VRP solutions (Chapter 7). The literature related to asymptotic properties of TSP and VRP (Chapter 5) will be considered in the corresponding chapter.

DTRP

The dynamic traveling repairman problem (DTRP) was introduced by Bertsimas and Van Ryzin, 1991 [5]. We will explain this problem in Chapter 3 and we discuss that paper in Chapter 5. To be brief, it is similar to an M/G/1 queue, except that the Poisson arrival process occurs inside a region, so there are also travel times involved. This introduces dependence between successive service times, where we interpret the travel time as a component of the service time. Of interest to us is the expected waiting time. To analyze this, it is helpful to review the literature on such dependency structures in queueing models. There are more papers that address dependency structures than the ones presented below, but they are less relevant. For example, they focus more on cross-correlation between service time and interarrival time, or autocorrelation of interarrival times. .

Livny et al. (1993) [6] studied the M/M/1 queue with dependence between successive service times (i.e., lag-1 autocorrelation). They conducted a simulation study to investigate the impact of the autocorrelation on performance measures such as waiting time. Two different simulation methods were used to induce the autocorrelation. They concluded that autocorrelation can have a significant impact on the waiting time, and should not be neglected.

Adan et al. (2003) [7] analyze a single-server queue where the interarrival times and the service times both depend on an underlying discrete-time Markov chain. They used this to model autocorrelation of service times, autocorrelation of interarrival times, as well as cross-correlation between interarrival times and service times. They analytically derive the Laplace-Stieltjes transform of the waiting time distribution. Such a transform can be numerically inverted to obtain the actual distribution. The authors found a slightly lower expected waiting time when interarrival times and service times are positively correlated, compared to no autocorrelation. Note that the DTRP does not have

such an underlying Markov chain and does not have autocorrelated interarrival times.

Miller (2003) [8] considered $n + 1$ points uniformly distributed in a square region, of which one point acts as a base location and the other n points as link points (points linked to the base location). The author derives the joint distribution $F(d_1, \ldots, d_n)$ of the distances d_i between link points and the base location. In the DTRP this can be applied to the setting $n = 2$, to obtain the joint distribution of the lengths of successive travel times.

Machine learning in VRPs

We have found two very recent papers in which authors use models that can predict the total travel distance in the optimal solution of a (capacitated) vehicle routing problem (VRP). Mei et al. (2022) [9] tested several models from the previous literature on their own generated data, which had a diversity of instances with regard to clustering and dispersion of customers. Furthermore, they contributed by using a measure of the clustering/dispersion of points (namely, the Average Nearest Neighbor Index) in order to improve one particular model by fine-tuning one regression coefficient in the model. They did not give the MAPE on the test set but instead showed the results graphically, but from the graph we can deduce that the MAPE was perhaps around 10%.

Akkerman et al. (2022) [10] compared various machine learning models. To train the models, they used many features that they collected from previous studies, as well as features that they contributed themselves. They tested the full model with all the features and multiple reduced models with less features, by using multiple feature selection techniques. With the Random Forest Regressor with feature selection, they obtained a MAPE of 3.07% on the test set, while without feature selection and using linear regression, they obtained 8.97%.

Bengio et. al (2021) [11] have studied something different. They surveyed recent attempts at solving combinatorial optimization problems using a machine learning model trained on a dataset of combinatorial optimization problems and their solutions. Note that the HHCRSP is also a combinatorial optimization problem. However, we are not interested in predicting the complete optimal solution, but only some characteristics (i.e., the costs). But in this paper, the authors were also interested, like us, in avoiding using an expensive combinatorial optimization algorithm.

3 Model description

Since we study three problems that are related but dissimilar, we will give a separate model description for each of them.

TSP and VRP

First we consider the traveling salesman problem (TSP). Given are a set of cities (or in our case customers), located in some square with area A, that have to be visited by a traveler. An instance of a TSP can be generated in an unbounded region, but we consider the square only. There is a direct path from each location to each other location, and the travel distance is known. The traveling salesman problem is to find the shortest possible route that starts and ends at the same customer and visits all customers exactly once. This problem can also be formulated using graph theory. An instance of the TSP is a complete, weighted, undirected graph, where each vertex is a customer, the edges are the paths between customers, and the edges' weights are the travel distances. The traveling salesman problem is then to find a Hamiltonian cycle with the shortest length.

The vehicle routing problem (VRP) is a generalization of the TSP. We again have customers inside a square, but now we also have one or multiple depots and we have multiple vehicles/travelers. The number of vehicles can be a decision variable, but if it is not, then the vehicle routing problem is to find a set of TSP routes, one for each vehicle, that starts and ends at a depot such that all customers are visited by exactly one vehicle. There are more variants of this problem, such as the capacitated VRP, where products with certain weights/volumes need to be delivered to customers, and the vehicles have a limited carrying capacity. Another variant is the VRP with time windows (VRPTW), where each customer has a preference for what time they want to be visited, given by a time window.

The problem that we are considering is to analyze what happens to the total travel distance for the optimal solution of a TSP, when the number of customers grows large. Specifically, if we sequentially generate n customers inside the square, drawing from a uniform distribution on the square, and if L_n is the length of the shortest TSP tour through those n customers, then of interest is the convergence of L_n, as $n \to \infty$. The same problem will be studied for the VRP, where also the number of depots can grow large.

Dynamic Traveling Repairman Problem

In the dynamic traveling repairman problem (DTRP), we have a single server system, where the server is active in some service region, which can be some inhabited area. Requests for service by customers arrive at some uniformly random location inside the region, randomly over time. The server, when idle, can return to some base location or can just wait at the last serviced customer, but we consider the second option only. See Figure 3.1 for an illustration. At the start of operations, the server can also be at the base location or at a uniformly random point in the region. The request for service can be handled again in an FCFS fashion or some other routing policy. Bertsimas and Van Ryzin [5] considered the problem of finding a good policy for routing the service vehicle that minimizes the expected waiting time of an arbitrary customer in stationarity.

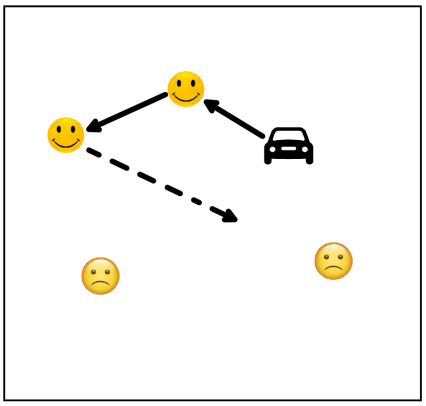

Figure 3.1: Illustration of the DTRP

This problem formulation can be applied to, for example, a patrolling police car, a health care provider who provides service to people with disabilities who make a phone call when they need help, and a repairman who visits customers as they request a repair. Although the second example is relevant to our research, we will call this problem the dynamic traveling repairman problem (DTRP). Here we are not interested in finding the optimal policy, but we want to analyze the FCFS policy and find the expected waiting time for customers in stationarity.

To describe the framework more formally, let customers arrive inside the region according to a Poisson process with rate λ. Let the service region be a square of area A. The travel time between two locations is the Euclidean distance. Let D be the travel time between two arbitrary locations. From theory we know that $\mathbb{E}D = c_1\sqrt{A}$, with $c_1 \approx 0.52$ [5]. The duration of on-site service S is independently and identically distributed for all customers. We can take the exponential distribution with some rate μ for simplification.

HHCRSP

Now we will consider the Home Health Care Routing and Scheduling Problem (HHCRSP). Recall that this problem is phase 2 and 3 of capacity planning (see the introduction) combined, hence in this problem we do not optimize the number of health care workers but we assume this number is given. An instance of the problem is a routing problem for one day.

Given is a set of customers of size n, located inside a square region of area A who each require care once during the day. They all have a preference for the moment they would like to receive care, given by a time window. For customer i this is $[\text{Start}_i, \text{End}_i]$. Customers cannot be served before their time window start time, but can be helped after the end time. In this case, a customer experiences waiting time, which is defined as the difference between the time that a worker arrives and the time window end time. The duration of care for customer i, i.e. the service time S_i, is known. Also given are a set of employees, each of whom works on given shifts. For each employee, the start time and end time of the shifts are given, and hence also the shift duration. The total number of shifts (i.e. employees) is n_s. Workers cannot start working before their shift start time, but they can end later than their end time, which will cause shift overtime. The overtime is then defined as the difference between the return time to the base location and the shift end time. Since the type of care can be specialized, we can also have a $n \times n_s$ qualification matrix Q, with $Q_{iv} = 1$ if worker v is qualified to provide service to customer i and zero otherwise. We neglect qualifications, i.e. Q consists of ones. We did this to simplify the problem, since the problem of building a machine learning model with good performance was already challenging. There is also a base location where workers depart from to serve their first customer, and return to after serving their last customer, like in a VRP. The locations are ordered $\{0, 1, \ldots, n\}$ where location 0 is the base location and the remaining ones are the customers. The travel time between location i and j, d_{ij}, is also given.

The HHCRSP is to find a set of TSP routes for each shift, starting and ending at the depot, such that each customer is served by exactly one shift, and such that the routes minimize a weighted sum of three cost measures, namely total combined travel time of all workers, total waiting time of all customers and total shift overtime of all workers combined. The three weights are specified in the parameters of paGOMEA. Such a set of routes also contains the arrival times at each location, and together they are called a schedule. To reiterate, when a worker is able to arrive at a customer too early, the arrival time will be set to the start of the customer's time window, so there will be some idle time for the worker. The problem we consider is not to build a good solver. The algorithm paGOMEA fulfills this task. We are interested in accurately predicting the three cost measures for a given instance, corresponding to the optimal solution, but without knowing the solution, using a machine learning

model. This will be explained in more detail in Chapter 7. As we neglect qualifications, we will also refer to the HHCRSP as the VRP with time windows (VRPTW). An overview of the notation used here and in Chapter 7 is given below.

Table 3.1: Notation for the VRPTW

A	area of the region
n	the number of customers
n_s	the number of shifts/servers
d_{ij}	the travel time between customers i and j
d_{ij_k}	the travel time between customer i and its kth nearest neighbor
$\bar{d}$	the length of the diagonal of the region, i.e. the maximum travel time
S_i	the on-site service time for customer i
Start_i, End_i	the start and end of the time window of customer i

4 Preliminaries

In this chapter, we discuss some theory that serves as a basis for subsequent chapters. In Section 4.1 we introduce some basic queueing theory and analyze the M/G/1 queue. Next, we consider the implications of this theory on the dynamic traveling repairman problem in Section 4.2, and in Section 4.3, we study a related mathematical problem. In Section 4.4 we show some known theoretic results on the M/M/s and the G/G/s queues without going into detail, which we will use in Chapter 7 when we discuss feature engineering. Lastly, we will explain the Random Forrest Regressor algorithm in Section 4.5, which we use as our main ML algorithm in Chapter 7.

4.1 Analysis of an ordinary M/G/1 queue

In queueing theory, we are interested in queueing systems, which are systems in which customers arrive at a location where they request service that can be provided by one or more servers. In some systems, customers will be waiting in a queue when all servers are busy with other customers, whereas in other systems, queueing is not allowed, and customers depart from the system. In queueing systems, depending on the type of system, various quantities are of interest such as the probability distribution of the waiting time in the queue, of the number of customers in the queue at a random point in time, and of the number of customers in the system. When systems are too complicated, the full probability distributions are intractible but a few moments of the distributions can remain tractible to derive.

In this section, we analyze the M/G/1 queue, which is a queueing system that is a simplification of another system in a later section that we are more interested in. In the M/G/1 queue, there is one server available (the 1 in M/G/1) and one queue, and customers arrive at the service location where they queue up if necessary. There is no limit to the length of the queue. Many queueing disciplines (ways in which certain customers in the queue are prioritized) are possible, but we analyze the commonly used first come, first serve (FCFS) discipline. The service time S, that is, the time it takes to serve a customer when a customer enters service, has some unspecified (i.e. general, the G in M/G/1) probability distribution that is identical for all customers. The way in which customers arrive at the system is called a *Poisson process* with some rate λ. This means that all the interarrival times (times between successive

arrivals) are independent and exponentially distributed with rate λ (the M in M/G/1 stands for Markovian that refers to the exponential distribution). We can also interpret λ as the number of arrivals per unit of time. It is important to note that in an ordinary M/G/1 queue, there is no dependency either between service times of different customers, interarrival times, or between an interarrival time and a service time. This will be different in the next section.

The objective here is to derive the Pollaczek-Khinchin (PK) formula for the expected waiting time, which is given by

$$\mathbb{E}W = \frac{\lambda \mathbb{E}S^2}{2(1-\rho)} \tag{4.1}$$

where $\rho = \lambda \mathbb{E}S$ is called the load on the system. We need $\rho < 1$ for the system to be stable and the queue length not to run to infinity (since otherwise the expected service time is larger than the expected interarrival time). To be precise, W is the waiting time in the queue of an arbitrary customer in a stationary situation, or equivalently, $W = \lim_{n\to\infty} W_n$ where W_n is the waiting time of the nth customer. We present a proof based on [12].

Let us first introduce some more notation. Denote S_i as the service time of the ith customer and N_i as the number of customers in the queue that the ith customer sees upon arrival. Let R_i be the residual service time seen by customer i, i.e. the remaining service time of the customer being served when customer i arrives. We then have the following relation for the waiting time:

$$W_i = R_i + \sum_{j=i-N_i}^{i-1} S_j \tag{4.2}$$

where the second term is the sum of service times of the customers in the queue before customer i. Next, we take expectations on both sides of the equality, and by Wald's lemma we have

$$\mathbb{E}W_i = \mathbb{E}R_i + \mathbb{E}N_i \mathbb{E}S \tag{4.3}$$

where we use independence between N_i and all S_j as well as between different S_j. Now taking the limit $i \to \infty$, we get

$$\mathbb{E}W = \mathbb{E}R + \mathbb{E}N\mathbb{E}S \tag{4.4}$$

where R and N are the limiting versions of the variables R_i and N_i, respectively. By PASTA, R and N are also the residual service time and the number of customers in the queue seen by a customer in stationarity, respectively. We state without proof that these limits exist as long as $\rho < 1$. Next, by Little's

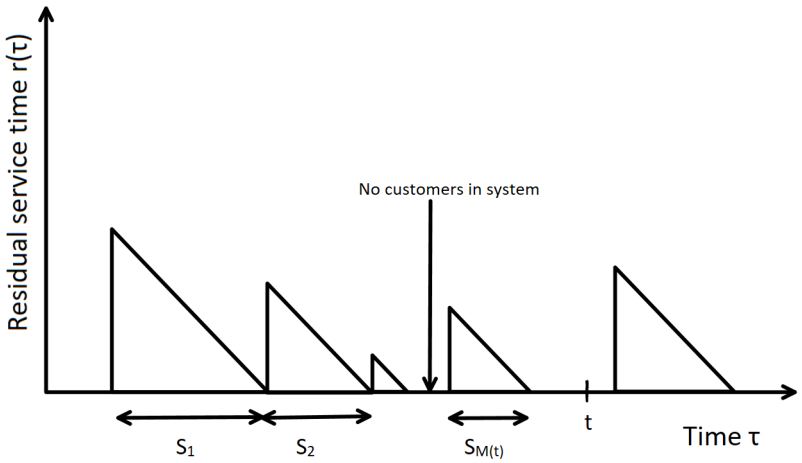

Figure 4.1: Realization of residual service time. $M(t)$ is the number of service completions before time t.

law (which holds in generality, regardless of the type of arrival process and service time distribution) (but did we need PASTA?) we have that $\mathbb{E}N = \lambda \mathbb{E}W$. Plugging this in eq. (4.4), we get $\mathbb{E}W = \mathbb{E}R + \rho \mathbb{E}W$, which leads to

$$\mathbb{E}W = \frac{\mathbb{E}R}{1-\rho}. \tag{4.5}$$

We need to analyze $\mathbb{E}R$ and we will do this graphically. Consider Fig. 4.1 where we see a realization of the residual service time $r(\tau)$ as a function of time τ. When the service starts for customer i, the function jumps to S_i and decays linearly at rate 1.

We will assume that the long-term time average residual service time coincides with the expected limiting residual service time, i.e.

$$\mathbb{E}R = \lim_{t\to\infty} \frac{1}{t} \int_0^t r(\tau) d\tau \tag{4.6}$$

Consider a time t when the system is empty. Inspecting the figure, we see that

$$\frac{1}{t} \int_0^t r(\tau) d\tau = \frac{1}{t} \sum_{i=1}^{M(t)} \frac{1}{2} S_i^2 \tag{4.7}$$

$$= \frac{1}{2} \frac{M(t)}{t} \frac{\sum_{i=1}^{M(t)} S_i^2}{M(t)} \tag{4.8}$$

By taking limits on both sides and assuming that the following limits exist, we have

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t r(\tau) d\tau = \frac{1}{2} \lim_{t\to\infty} \frac{M(t)}{t} \cdot \lim_{t\to\infty} \frac{\sum_{i=1}^{M(t)} S_i^2}{M(t)} \tag{4.9}$$

where the first limit is the long-term average departure rate, which is also the arrival rate λ, and the second limit is $\mathbb{E}S^2$ by the law of large numbers. Hence

we get $\mathbb{E}R = \frac{1}{2}\lambda\mathbb{E}S^2$, and plugging this into eq. (4.5), we get the PK formula (4.1).

4.2 Application of queueing theory to the DTRP

The M/G/1 model can be used to analyze the DTRP with the FCFS policy as follows. When the server needs to start traveling to the next customer, we can consider this to be the moment when a new service starts. So the service time can be considered as the sum of travel time and the on-site service time. Since the server waits at the last serviced customer when idle, all travel time random variables are, in stationarity, identically distributed, since they are simply the distance between two uniformly random locations. Although on-site service times are always independent, a slight dependency does exist between two successive travel times. For example, when the previous customer is located near the corner of the square, a relatively longer travel time can be expected to go to the next customer. We neglect this dependency in this section. Bertsimas and Van Ryzin also neglected this. In this case, the (total) service times of successive customers are independent.

In stationarity, the expectation of the total service time is $\mathbb{E}S + \mathbb{E}D = \mathbb{E}S + c_1\sqrt{A}$ (see Chapter 3) while the second moment is

$$\mathbb{E}(S+D)^2 = \mathbb{E}S^2 + \mathbb{E}D^2 + 2\mathbb{E}S\mathbb{E}D = \mathbb{E}S^2 + c_2A + 2c_1\sqrt{A}\mathbb{E}S \tag{4.10}$$

with $c_2 = 1/3$, where we use that $\mathbb{E}D^2 = c_2A$ (see [5]). When the stability condition $\lambda(\mathbb{E}S + c_1\sqrt{A}) < 1$ holds, the PK formula gives us

$$\mathbb{E}W = \frac{\lambda(\mathbb{E}S^2 + c_2A + 2c_1\sqrt{A}\mathbb{E}S)}{2(1 - \lambda\mathbb{E}S - \lambda c_1\sqrt{A})} \tag{4.11}$$

where W is the expected waiting time for a customer in the 'queue' in stationarity, i.e., the time from arrival until the server starts driving to that customer. The expected 'real' waiting time, i.e. the time from arrival until the server reaches the customer, is then $\mathbb{E}W + \mathbb{E}D = \mathbb{E}W + c_1\sqrt{A}$. In Chapter 5 we investigate whether neglecting this successive service time dependency was justified.

4.3 Covariance of adjacent travel segments

In the previous section, we neglected the dependence between successive service times. This dependence can in part be described by the covariance between successive travel times, i.e. the length of adjacent travel segments. To

analytically derive the expected waiting time by taking the dependency structure into account, one may need some theory on this covariance. We do not further analyze the DTRP analytically in this thesis, but we will refer to this section in the final discussion, Chapter 8.

The problem can be reduced to the following. Suppose that we have a square of area A, and we have three points inside, P_1, P_2, and P_3, all sampled independently from $\mathcal{U}([0,\sqrt{A}]^2)$. Let P_i have coordinates (x_i, y_i). Suppose that we travel from P_1 to P_2 (having distance D_1) and then from P_2 to P_3 (having distance D_2. We are interested in $\mathrm{Cov}(D_1, D_2)$ and also the correlation, $\mathrm{Corr}(D_1, D_2)$, but for this we need an expression for $\mathbb{E}(D_1 D_2)$. Since the density of all x_i as well as y_i is $f(x) = 1/\sqrt{A}$, we can condition on the coordinates of all three points to obtain a six-fold integral

$$\mathbb{E}(D_1 D_2) = \int_{[0,\sqrt{A}]^6} \mathbb{E}(D_1 D_2 | P_1, P_2, P_3)(f(x))^6 dx_1 dx_2 dx_3 dy_1 dy_2 dy_3 \tag{4.12}$$

$$\begin{aligned} &= \int_{[0,\sqrt{A}]^6} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \times \\ &\quad \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \left(\frac{1}{\sqrt{A}}\right)^6 dx_1 \dots dy_3 \end{aligned} \tag{4.13}$$

Using the coordinate transformation $\bar{x}_i = x_i/\sqrt{A}$ and $\bar{y}_i = y_i/\sqrt{A}$ (and letting $\bar{P}_i = (\bar{x}_i, \bar{y}_i)$, we can rewrite the integral as

$$\begin{aligned} &\int_{[0,1]^6} \sqrt{(\sqrt{A}\bar{x}_1 - \sqrt{A}\bar{x}_2)^2 + (\sqrt{A}\bar{y}_1 - \sqrt{A}\bar{y}_2)^2} \times \\ &\sqrt{(\sqrt{A}\bar{x}_2 - \sqrt{A}\bar{x}_3)^2 + (\sqrt{A}\bar{y}_2 - \sqrt{A}\bar{y}_3)^2} \left(\frac{1}{\sqrt{A}}\right)^6 (\sqrt{A})^6 d\bar{x}_1 \dots d\bar{y}_3 \end{aligned} \tag{4.14}$$

$$= A \int_{[0,1]^6} \sqrt{(\bar{x}_1 - \bar{x}_2)^2 + (\bar{y}_1 - \bar{y}_2)^2} \sqrt{(\bar{x}_2 - \bar{x}_3)^2 + (\bar{y}_2 - \bar{y}_3)^2} d\bar{x}_1 \dots d\bar{y}_3 \tag{4.15}$$

$$= A\mathbb{E}(\bar{D}_1 \bar{D}_2) \tag{4.16}$$

where $\bar{D}_1 = ||\bar{P}_1 - \bar{P}_2||$ and $\bar{D}_2 = ||\bar{P}_2 - \bar{P}_3||$. Eq. (4.16) holds because $\bar{P}_i$ can be seen as a random point from a $\mathcal{U}([0,1]^2)$ distribution, having density $\bar{f}(x) = 1$. Using numerical integration we found that $\mathbb{E}(\bar{D}_2 \bar{D}_2) = 0.27890$. Now, for the covariance, we have

$$\mathrm{Cov}(D_1, D_2) = \mathbb{E}(D_1 D_2) - \mathbb{E}D_1 \mathbb{E}D_2 = A(\mathbb{E}(\bar{D}_1 \bar{D}_2) - c_1^2), \tag{4.17}$$

where $c_1 = 0.52$ (see [5]) and it is interesting to see that the covariance scales linearly with the area. For the correlation, we have

$$\mathrm{Corr}(D_1, D_2) = \frac{\mathrm{Cov}(D_1, D_2)}{\sqrt{\mathrm{Var}D_1 \mathrm{Var}D_2}} = \frac{A(\mathbb{E}(\bar{D}_1 \bar{D}_2) - c_1^2)}{c_2 A} = \frac{1}{c_2}\mathbb{E}(\bar{D}_1 \bar{D}_2) - c_1^2) \tag{4.18}$$

where $c_2 = 1/3$ (see [5]). Here we see that the correlation is constant with respect to area, which makes sense because the correlation is a scale-invariant quantity.

4.4 Notes on the M/M/s and G/G/s queues

The M/M/s queue, also called the Erlang C model, or the Erlang delay model, is another type of queueing model. While the interarrival times are exponentially distributed with rate λ, as in the M/G/1 queue, now the service times also have an exponential distribution, say with rate μ. There is also not one, but a total of s servers ready to serve customers. Each server can handle only one customer. Servers cannot collaborate on one customer to serve that customer faster. Servers do not have separate queues. Instead, there is a central queue, and customers lined up in the queue will receive service whenever one of the servers is done with their customer. Again, we consider the FCFS policy. We are interested in the full probability distribution of the waiting time W of an arbitrary customer in stationarity. It is clear that a customer does not need to wait when it finds fewer than s customers in the system upon arrival. Therefore, the distribution of waiting time is dependent on, and can in fact be derived from, the distribution of the number of customers in the system.

The number of customers in the system evolves as a birth-death process, which is a type of Markov Chain. From the theory of birth-death processes, one can find $\pi(x)$, the probability that there are x customers in the system, in stationarity. Then one also needs PASTA (Poisson arrivals see time averages) to find that

$$\mathbb{P}(W > t) = \mathbb{P}(W > 0)e^{-(s\mu-\lambda)t} \tag{4.19}$$

where the delay probability is

$$\mathbb{P}(W > 0) = \frac{1}{1-\rho}\pi(s) = \frac{1}{1-\rho}\frac{a^s}{s!}\pi(0) \tag{4.20}$$

with $a = \lambda/\mu$ being the load and $\rho = \lambda/(\mu s)$ being the load per server, and $\pi(0)$ given by

$$\pi(0) = \left[\sum_{j=0}^{s-1}\frac{a^j}{j!} + \frac{a^s}{s!}\frac{1}{1-\rho}\right]^{-1} \tag{4.21}$$

Now for the G/G/s queue, we have a general, unspecified probability distribution for the service times as well as for the interarrival times. The two distributions can be different. The exact waiting time distribution in closed form is unknown, so we are interested in a relatively simple approximation. Abate et al. (1995) [13] proposed the simple exponential form

$$\mathbb{P}(W > t) \approx \alpha e^{-\eta t} \tag{4.22}$$

that has a similar form as the expression for the M/M/s queue, eq. (4.19). Through numerical examples, they showed that this approximation is remarkably good. This approximation holds when t is sufficiently large. In fact, this approximation is asymptotically correct, as $t \to \infty$, for various (phase-type) multi-server queueing systems (Whitt, 1993 [14]). We need appropriate values for the constants α and η which do not depend on t. Note that for large t, the approximation does not strongly depend on the value of α. To see this, consider the pth quantile w_p (with, e.g., $p = 0.95$), that is, let $\mathbb{P}(W > w_p) = 1 - p$. From the exponential form it follows then that

$$w_p = \log\left(\frac{\alpha}{1-p}\right)\frac{1}{\eta} = (\log(\alpha) - \log(1-p))\frac{1}{\eta} \tag{4.23}$$

If p is close to 1, and assuming a very small value for α is not appropriate, we see that w_p does not greatly depend on α.

Therefore, for simplicity, we follow the M/M/s theory and for the value of α we will use the probability that the waiting time is larger than zero in the M/M/s queue. But now we use $a = \lambda \mathbb{E}S$ (with in this case $\lambda = 1/\mathbb{E}A$, with A the interarrival time random variable) and $\rho = a/s$. In fact, the value of $\mathbb{P}(W > 0)$ is similar for the M/M/s and G/G/s queues, under certain conditions (see Whitt, 1992 [15]). A suitable value for η, according to [13], is given by

$$\eta = \frac{2s(1-\rho)}{\mathbb{E}S(c_a^2 + c_s^2)} \tag{4.24}$$

where c_a^2 and c_s^2 are the squared coefficients of variation for the interarrival times and the service times, respectively. The squared coefficient of variation for a random variable X is defined by $\mathrm{Var}(X)/(\mathbb{E}X)^2$. Comparing this value for η with the expressions (4.19) and (4.22), we can see that the value of η in the M/M/s queue, namely $s\mu - \lambda$, is multiplied by $2/(c_a^2 + c_s^2)$ to correct for the fact that we have a G/G/s queue now.

Next, we are interested in an expression for $\mathbb{E}(W - T)^+$ for some fixed T. Consider that T is some acceptable maximum amount of waiting and that we want the expected amount of violation of this waiting time limit. This will be of interest in Chapter 7. Using the notation of eq. (4.22), the probability density of W is given by

$$f(t) = \frac{d}{dt}\mathbb{P}(W < t) = \alpha\eta e^{-\eta t} \tag{4.25}$$

Hence, we have

$$\mathbb{E}(W - T)^+ = \int_0^\infty \max(0, t - T) f(t) dt \tag{4.26}$$

$$= \int_T^\infty (t - T)\alpha\eta e^{-\eta t} dt \tag{4.27}$$

Integration by parts gives the result

$$\mathbb{E}(W-T)^+ = \frac{\alpha}{\eta} e^{-\eta T}. \tag{4.28}$$

Since the exponential form (4.22) is a generalization of the tail probability in the M/M/s queue, we can use expression (4.28) for both types of queues. So for both the M/M/s and the G/G/s queue, we use for α, the expression (4.20) (with $\mu = 1/\mathbb{E}S$ in the G/G/s model). For η, we use $s\mu - \lambda$ in the M/M/s queue and expression (4.24) in the G/G/s queue.

4.5 Random Forest Regressor

Since we will use a Random Forest Regressor (RFR) as the machine learning model in Chapter 7, we will explain briefly how this algorithm works. RFRs use ensemble learning, which is the method of training multiple ML models and using the mean predicted value ("aggregation") across the individual models as the prediction for a new data point (in the case of regression). Ensemble learning can be done with different types of ML models, but RFRs use decision trees only.

A decision tree is a binary tree that splits on each node based on the value of a particular feature being higher or lower than some threshold. Which feature is chosen at each node is decided when the decision tree is trained on some dataset, where each instance contains multiple features and one target value. Training is done in a greedy way as follows. First, compute the variance of the target values of all the (training) data. Then compare all possible splits and choose the split that reduces the variance the most. The post-split variance is defined as follows. A candidate split splits the dataset into two subsets. The post-split variance is the sum of the two weighted variances of the target values of the two corresponding subsets of the data. The weights are given by the size of the subsets (normalized, so the sum is 1). This variance reduction criterion is used for all successive nodes until the tree has reached the desired depth (based on some criterion). The prediction on a new data point is simply the average of the target values of the training instances that correspond to the leaf node to which the new data point belongs.

A decision tree suffers from low bias and high variance, which makes it not so generalizable. Using an ensemble of such trees, i.e. an RFR, decreases the variance and has a much better bias-variance trade-off. But this only works when the individual trees are not similar and there is diversity. This is done in two ways. First, for each tree and each node of the tree, only a random subset of features are allowed to be chosen from. Second, each tree is trained on a dataset that was bootstrapped from the original dataset (i.e. instances are picked with replacement and the sizes of new datasets are the same as the size

of the original). This second aspect was introduced by Breiman (2001). He also showed that there is no problem of overfitting when the number of trees is increased. RFRs use aggregation of the trees, and together with bootstrapping this is also called bagging.

This algorithm can also provide information on the relative importance of each feature. There are two ways but we use the "Gini importance", also called "mean decrease in impurity" (MDI) [16]. Each node of each tree in the RFR has a decrease in impurity and this is the variance reduction as described earlier. Based on a single tree, we can assign an importance score to a feature by summing all variance reductions across all nodes that split on that feature. The MDI of a feature according to the RFR is simply the average of such scores across all individual trees.

5 DTRP simulation

In Section 4.2 we have introduced the dynamic traveling repairman problem, which is the problem of finding a good policy for routing a service vehicle to service all customers in need of repair, which arrive according to a Poisson process in some region. The locations of the customers are independent and uniformly distributed in the region and the on-site service times are i.i.d. The problem is to find the best routing policy that minimizes the average time that a customer spends in the system.

Bertsimas and Van Ryzin (1991) [5] contended that for the FCFS policy, the M/G/1 model can be used and the Pollaczek-Khinchine (PK) formula for the expected waiting time still holds even though this dependency exists, where they refer to Bertsekas and Gallager [12] for the proof. However, we did not find a justification for their statement in their reference. We will use a computer simulation (in particular a discrete event simulation) to show a discrepancy between what the PK formula predicts and the real expected waiting time. And hence, we will have shown that the claim of Bertsimas and Van Ryzin is incorrect.

5.1 Methodology

As the region we will use a square of a by a, where we set $a = 7$. The distance metric between two points inside the square is the Euclidean distance. Let λ be the arrival rate of the Poisson process, D be the distance between two arbitrary points, and S be the on-site service time of an arbitrary customer. We know that $\mathbb{E}D \approx 7 * 0.5214$ and $\mathbb{E}D^2 = 49 * 1/3$. In the simulation we use, for the on-site service time, the Exp(1) distribution, so that $\mathbb{E}S = 1$ and Var $S = 1$. Let $S_{tot} = D + S$ be the total service time. In this case, the travel time is a bigger component, on average, of the total service time than the on-site service time and hence the impact of the dependency between successive service times should be more clearly noticeable. The PK formula predicts a value for the expected waiting time, $\mathbb{E}W_{PK}$, that can be computed with eq. (4.11).

Define the load ρ as $\rho = \lambda \mathbb{E}S_{tot}$. We simulate the system for various values of ρ between 0 and 1, in particular, for $\{0.1, 0.2, \ldots, 0.9, 0.95\}$. By fixing ρ, we fix $\lambda = \rho / \mathbb{E}S_{tot}$. The system will process a total of 1e6 arriving customers per

experiment. The waiting times of the first 1e5 will not be taken into account, as we consider this the warm-up period, after which we can consider the system to have reached the steady state. By steady state we mean that after the warm-up period, the waiting times of customers will be (not independently) sampled from the steady-state distribution, and we are interested in its expected value.

To validate the correctness of the discrete event simulation, as a first check, we will do a simulation with $\rho = 0.3$. Now we set $a = 0$, so that there is no travel time, which turns the system in an ordinary M/M/1 queue. The running mean waiting time can be seen in Figure (5.1). From standard queueing theory, it is expected to approach $\mathbb{E}W_{PK}$, which it does, validating the simulation.

For another validation, we do 700 experiments of 1e6 arrivals each, with $\rho = 0.95$ and again with $a = 0$. We then register the queue length that each arriving customer sees before joining the queue. According to the principle of PASTA (Poisson Arrivals See Time Averages), these arrival moments can be considered as arbitrary moments, as if they are moments chosen by an outside observer to observe the system. We then compute the mean queue length observed by arriving customers per experiment. By Little's law, for the expected queue length $\mathbb{E}L$ we have $\mathbb{E}L = \lambda \mathbb{E}W_{PK}$, which for this load equals 18.05. A histogram of the results is shown in Figure 5.2. The mean of all data is 18.048, validating the results. Notice that with this load, there is a large variance in the results, which is also visible if we make plots like in Figure 5.1 for different experiments, where we have noticed that convergence is much slower. For smaller loads, we did not observe this large variance. For a high load, it takes many arrivals for the mean to converge to the theoretical value.

This method of conducting a large number of experiments will be used to compute confidence intervals. For each experiment, we create a 95% confidence interval for the expected waiting time as follows, using a kind of batch-means method [17]. We run $n = 1000$ experiments for each value of the load, with 1e6 arrivals per experiment and a warm-up of 1e5 arrivals. Let w_i be the sample mean of the waiting times of all customers except for those in the warm-up period, from experiment i, $i = 1, \ldots, n$. Let $\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$. By the central limit theorem, we can say that $w_1, ..., w_n$ are approximately normally distributed with mean $\mathbb{E}W$ and unknown variance. From basic statistical theory it follows that $\sqrt{n}(\bar{w} - \mathbb{E}W)/\sigma$ has approximately a t-distribution with $n-1$ degrees of freedom, where σ is the sample standard deviation of $\{w_1, \ldots, w_n\}$. From this it follows that a 95% confidence interval for $\mathbb{E}W$ is given by $\bar{w} \pm \frac{\sigma}{\sqrt{n}} t_{n-1;\, 0.975}$.

5.2 Results

We will find that the PK formula underpredicts the mean waiting time in this setting, so we are actually interested in confidence intervals for the quantities $\mathbb{E}W - \mathbb{E}W_{PK}$ which we call the absolute excess waiting time compared to the

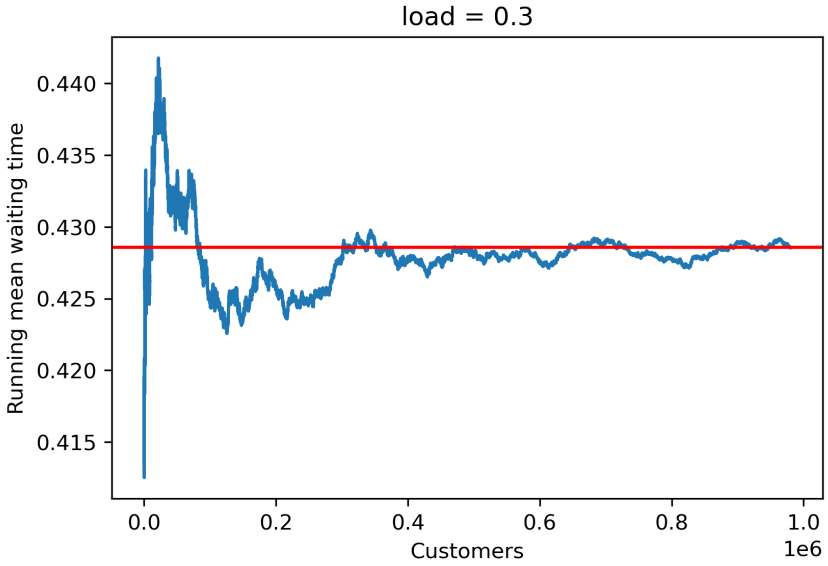

Figure 5.1: The running mean waiting time of one experiment. That is, for each arrival we compute the sample mean waiting time of all the previous arriving customers (excluding the ones in the warm-up). The red line is $\mathbb{E}W_{PK}$

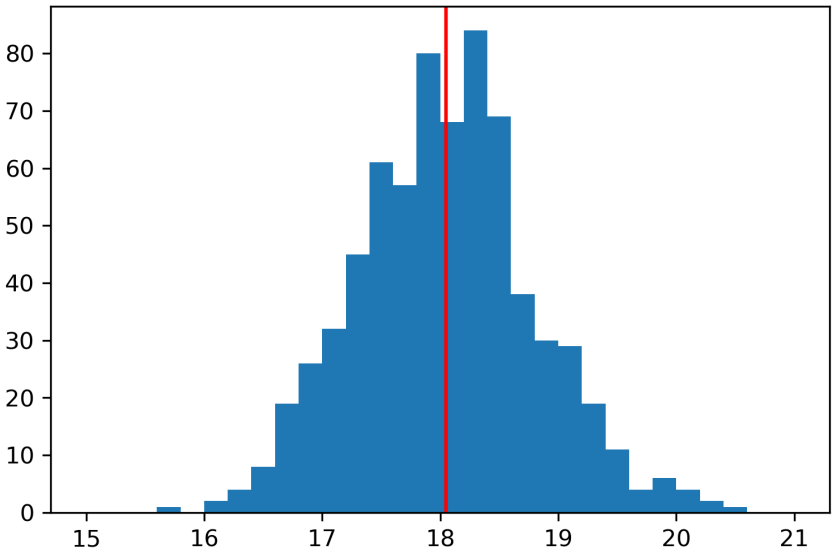

Figure 5.2: Histogram of $\{q_1, \ldots, q_{700}\}$, where q_i is the mean queue length in experiment i excluding the warm-up period. The red line is the PK value of 18.05. The mean of $\{q_1, \ldots, q_{700}\}$ is approx. 18.048.

PK prediction, and $(\mathbb{E}W - \mathbb{E}W_{PK})/\mathbb{E}W_{PK}$, the relative excess (the percentage of the excess compared to the PK value). For the absolute excess, the location of the confidence interval changes to $\bar{w} - \mathbb{E}W_{PK}$ but not its width, whereas for the relative excess we have the confidence interval

$$\frac{\bar{w} - \mathbb{E}W_{PK}}{\mathbb{E}W_{PK}} \pm \frac{1}{\mathbb{E}W_{PK}} \frac{\sigma}{\sqrt{n}} t_{n-1;\, 0.975}. \tag{5.1}$$

For each tested $\rho \in \{0.1, \ldots, 0.9, 0.95\}$ these intervals are plotted in Figure 5.3.

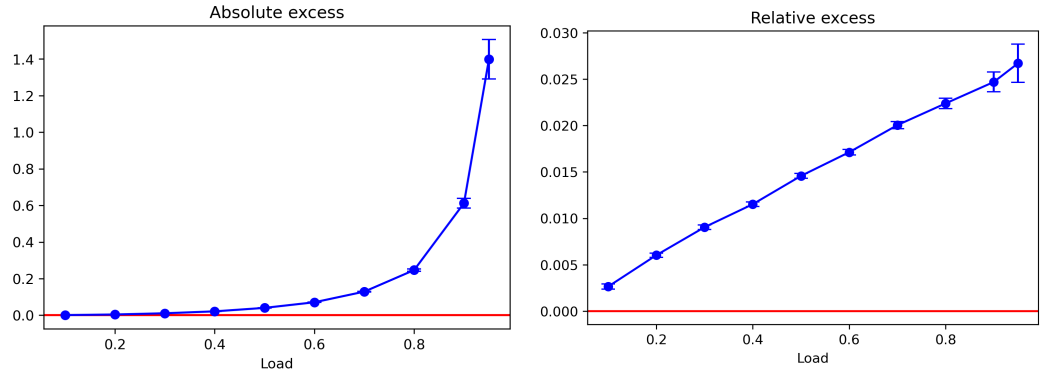

Figure 5.3: Plot of $\bar{w} - \mathbb{E}W_{PK}$ (left) and $\frac{\bar{w} - \mathbb{E}W_{PK}}{\mathbb{E}W_{PK}}$ (right) including confidence intervals.

The absolute excess seems to increase asymptotically when $\rho \to 1$, as the stochasticity increases with higher loads. Notice that the relative excess seems to be a straight line, not necessarily through the origin. Note that when $\rho \to 0$, fewer customers have waiting time as the queue size decreases to zero, but $\mathbb{E}W_{PK}$ also goes to zero. Thus, it is not clear which one of the two dominates and whether the straight line goes through the origin. So, it seems that

$$\frac{\mathbb{E}W - \mathbb{E}W_{PK}}{\mathbb{E}W_{PK}} = c_1 + c_2 \lambda \mathbb{E}S_{tot}, \tag{5.2}$$

that is,

$$\mathbb{E}W = \mathbb{E}W_{PK}(\bar{c}_1 + c_2 \lambda \mathbb{E}S_{tot}) \tag{5.3}$$

with c_1 and c_2 being constants and $\bar{c}_1 = c_1 + 1$. These constants probably depend on the service area.

Using simple linear regression on the midpoints of the confidence intervals, we can find the coefficients c_1 and c_2 in eq. (5.2). Now we will test if these linear regression coefficients indeed depend on the service area. For a having values of 70, 700 and 7000 we perform the same kind of data generation, although with fewer arrivals, to limit the excessive computation time. The number of arrivals increases linearly from 1e5 to 1e6 when ρ goes from 0.1 to 0.95. The warm-up

period also increases from 1e3 to 1e4. As the order of magnitude of a increases, the travel time becomes a more dominant component of the total service time, making the on-site service time perhaps negligible. The dependence between successive service times is expected to increase as a increases, until the on-site service time becomes negligible. We can see in Figure 5.4 that also with larger values of a, we have an approximately linear relationship between load and relative excess.

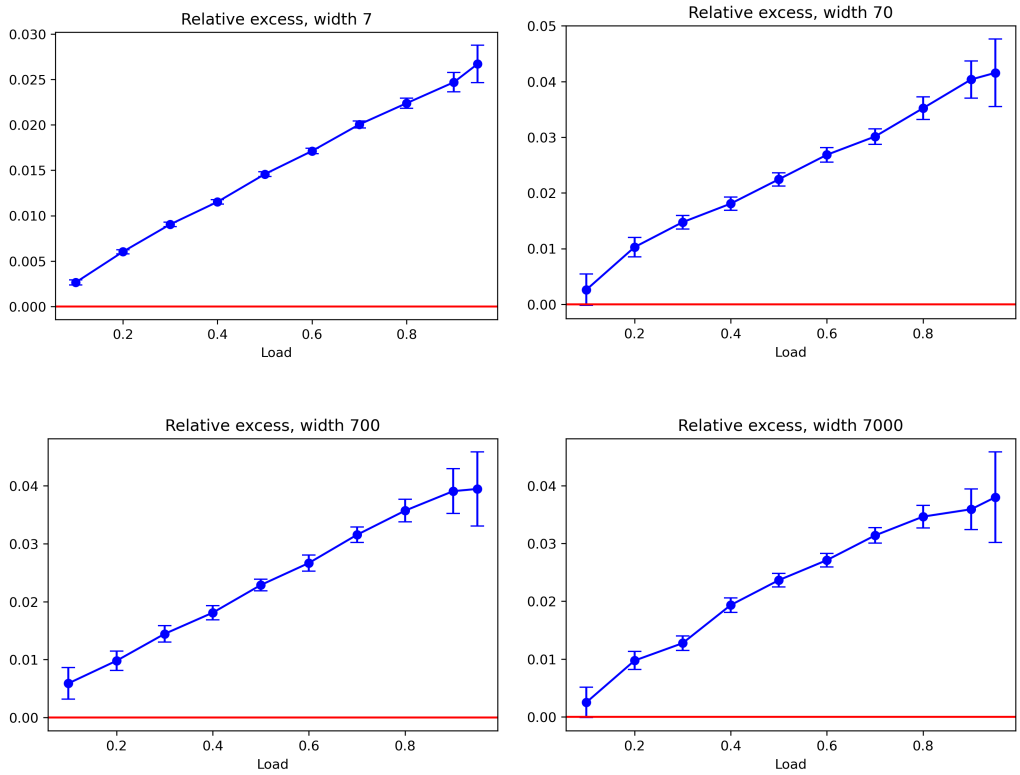

Figure 5.4: Relative excess confidence intervals plotted for different values of the width a of the square service region. When $a > 7$, the confidence intervals are larger because fewer experiments (i.e. batch means) are available.

The values of the linear regression coefficients can be found in Table 5.1. We see that the slope converges to a value of 0.041 as a increases. Again, convergence was expected due to the on-site service time becoming negligible. No convergence was detected for the intercept, maybe because of not having enough data.

To illustrate, in a different way, that the expected real waiting time differs significantly from $\mathbb{E}W_{PK}$, two histograms are shown in Figure (5.5) for two particular loads of 0.6 and 0.95. We see that for both loads (especially for $\rho = 0.6$), there is a significant difference between the histogram peak and the PK value. For $\rho = 0.6$ it seems that the probability that the PK value was the

correct value for the waiting time is almost zero. For $\rho = 0.95$, this probability also seems low. Also, note that the normality assumption of the batch means is justified, indicating that our batch means method gives accurate results.

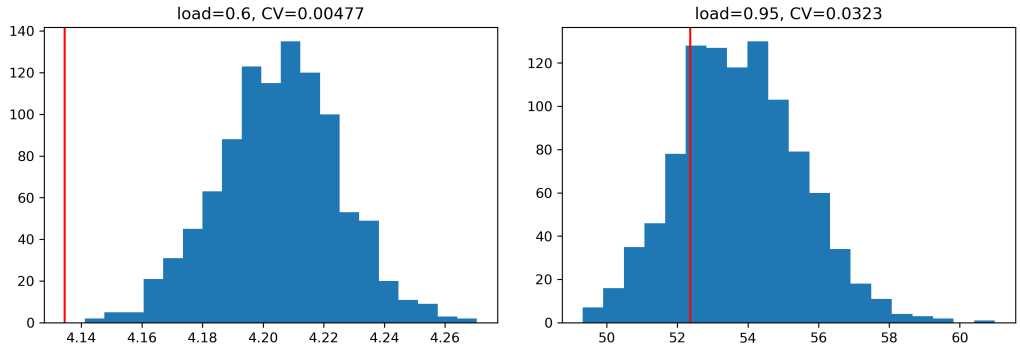

Figure 5.5: Two histograms of the batch means. CV is the coefficient of variation of the batch means. The red line is $\mathbb{E}W_{PK}$.

a	c_1	c_2
7	0.00047	0.028
70	0.00030	0.044
700	0.0020	0.041
7000	0.0013	0.041

Table 5.1: Linear regression coefficients from eq. (5.2), comparing different widths a of the square service region.

6 Asymptotic analysis

In this chapter we give an asymptotic analysis of the TSP and a particular variant of the VRP, in the sense that we are interested in the tour length as the number of locations that need to be visited grows large. In both problems, we find that the convergence is the same, up the constant of convergence. The variant of the VRP studied has similarities to the type of VRP that we use in Chapter 7. In Section 6.1 we study the TSP, and then we study the VRP in Section 6.2. We present existing theory from the literature in this chapter.

6.1 TSP

Beardwood, Halton, and Hammersley (BHH) described and proved in 1959 the asymptotic behavior of the length of the shortest path of a TSP problem with Euclidean metric, as the number of points in d-dimensional space goes to infinity. The theorem assumes an arbitrary density function of the distribution of points in $\mathbb{R}^d$. We will consider a simplified version of the theorem, where the points are uniformly distributed in two dimensions on the unit square. For full details of the proof, see Steele, 1990 [18].

Theorem 1 (BHH, 1959, simple case)**.** *Let X_i, $i \in \mathbb{N}$ be independent and uniformly distributed points on $[0,1]^2$ and let L_n be the length of the shortest path through $X_1, \ldots, X_n$. Then*

$$\frac{L_n}{\sqrt{n}} \to \alpha(TSP) \quad \textit{almost surely, as } n \to \infty \tag{6.1}$$

for some constant $\alpha(TSP) > 0$.

Proof outline. Let $N(t), t > 0$ be a Poisson process with rate 1, where an arrival corresponds to a point appearing uniformly random on the unit square. It suffices to prove that $L_{N(t)}/\sqrt{t} \to \alpha(TSP)$ a.s. where $L_{N(t)}$ is the length of the shortest path through the $N(t)$ points that have arrived so far.

Divide the unit square into m^2 smaller subsquares $Q_1, \ldots, Q_{m^2}$ (i.e. with sides $1/m$). Define the random variable $\lambda_t^m(Q_i)$ as the length of the shortest route through the points in square Q_i that arrived before t. Note that these lengths in different squares are i.i.d. Also, note that $m\lambda_{m^2t}^m(Q_i)$ has the same distribution as $L_{N(t)}$, because in expectation, at time m^2t there are m^2t points

in the unit square, and hence t points in one subsquare, the same number of points in the unit square at time t. Since the length of such a path in a subsquare is proportional to the length of the side, a factor m is needed. We will use this to show that

$$\sum_{i=1}^{m^2} \frac{\lambda^m_{m^2t}(Q_i)}{m} \to \mathbb{E}L_{N(t)} \text{ almost surely, as } m \to \infty. \tag{6.2}$$

Using this observation, if we define the event A_m as

$$|\sum_{i=1}^{m^2} \frac{\lambda^m_{m^2t}(Q_i)}{m} - \mathbb{E}L_{N(t)}| > \varepsilon, \tag{6.3}$$

it follows from Chebyshev's inequality that

$$\mathbb{P}(A_m) \leq \frac{\mathrm{Var}(L_{N(t)})}{m^2\varepsilon^2} \tag{6.4}$$

since $\frac{1}{m^2}\sum_{i=1}^{m^2} m\lambda^m_{m^2t}(Q_i) = \sum_{i=1}^{m^2} \frac{\lambda^m_{m^2t}(Q_i)}{m}$ has expectation $\mathbb{E}L_{N(t)}$ and variance $\mathrm{Var}(L_{N(t)})/m^2$. We have used here that $\mathrm{Var}(L_{N(t)}) < \infty$ as can be seen by the simple fact that $L_{N(t)} \leq N(t)\sqrt{2}$ ($\sqrt{2}$ is the maximal distance between 2 points). If we sum the rhs of eq. (6.4) over all $m > 0$, we get a finite value, so we also have $\sum_{m=1}^{\infty} \mathbb{P}(A_m) < \infty$. It now follows from the Borel-Cantelli lemma that $P(\limsup_{m\to\infty} A_m) = 0$ where $\limsup_{m\to\infty} A_m$ is defined as the set of outcomes (i.e. the locations of X_i) for which infinitely many events A_m occur. Thus it follows that with probability 1, there is an $M > 0$ such that A_m does not occur for all $m > M$, which is to say the inequality in (6.3) is flipped. This means that

$$\mathbb{P}\left(\lim_{m\to\infty} \sum_{i=1}^{m^2} \frac{\lambda^m_{m^2t}(Q_i)}{m} = \mathbb{E}L_{N(t)}\right) = 1 \tag{6.5}$$

which is the same statement as eq. (6.2).

Next, it can be shown that, for all m and t,

$$-6m + \sum_{i=1}^{m^2} \lambda^m_t(Q_i) \leq L_{N(t)} \leq m\sqrt{5} + \sum_{i=1}^{m^2} \lambda^m_t(Q_i) \tag{6.6}$$

The right inequality follows by first starting with the optimal tours within each subsquare and connecting them using a snake raster order (i.e. using adjacent subsquares) to make a tour through all the points in the unit square (we get an overhead of $m\sqrt{5}$ since the maximum distance between two points in two adjacent subsquares is $\sqrt{5}/m$). The left inequality can be shown by first

starting with the optimal path through all points in the unit square and then cutting that tour to pieces to create tours through the points in the subsquares.

Furthermore, using (6.6), it can be shown that there is a $C > 0$ such that

$$\lim_{t\to\infty} \frac{\mathbb{E}L_{N(t)}}{\sqrt{t}} = C \tag{6.7}$$

Dividing everything in (6.6) by m, considering time m^2t and taking the limit as $m \to \infty$, we get almost surely

$$-6 + \mathbb{E}L_{N(t)} \leq \liminf_{m\to\infty} \frac{L_{N(m^2t)}}{m} \leq \limsup_{m\to\infty} \frac{L_{N(m^2t)}}{m} \leq \sqrt{5} + \mathbb{E}L_{N(t)} \tag{6.8}$$

Dividing everything by $\sqrt{t}$ and taking the limit as $t \to \infty$, we get

$$C \leq \liminf_{s\to\infty} \frac{L_{N(s)}}{\sqrt{s}} \leq \limsup_{s\to\infty} \frac{L_{N(s)}}{\sqrt{s}} \leq C \tag{6.9}$$

and the theorem follows, with $C = \alpha(TSP)$. ■

More generally, when we have a square with area A, then we also have the convergence $L_n/\sqrt{An} \to \alpha(TSP)$ almost surely. With the Euclidean distance metric and a uniform distribution for the customers, this constant has been estimated by simulation to be $\alpha(TSP) \approx 0.76$ (Stein, 1978 [19])

6.2 VRP

The simplest generalization of the traveling salesman problem is the vehicle routing problem (VRP). In this problem, there is again a set of locations/customers that have to be visited, but now there are multiple travelers available who start traveling from usually one depot. Each traveler gets assigned a set of customers (all sets are disjoint and the union covers all customers), and one possible objective is to find a set of routes, each starting and ending at a depot, that minimizes the total travel distance.

There are good algorithms to find a high quality solution to this problem, such as the Clarke and Wright savings algorithm [20]. However, to the best of our knowledge, no mathematical theory was known about the asymptotic behavior of this problem until relatively recently. In [21], Baltz et al. have done this analysis, although for a slightly different problem, namely the Multiple Depot Vehicle Routing Problem (MDVRP), where there are k depots and n customers, and both the customers and the depots are uniformly distributed in $[0, 1]^2$. The problem is to find a set of TSP tours such that all customers are served and each tour contains exactly one depot, whereas not all depots

have to be used. The routes should be disjoint, except that a depot can be shared. The objective is to minimize the total travel distance.

We will present the results of [21] and some details of their analysis that are within the scope of this thesis. The main result is that when $k = o(n)$, i.e., when there are a "small" number of depots compared to the number of customers, the asymptotic behavior is exactly the same as the TSP, while if k grows linearly in n (i.e. $k = \beta n$, $\beta > 0$), then the asymptotic behavior is still similar, but the constant of convergence α_λ differs and is dependent on k. One would expect this, since if β is very large, each customer can have its own depot that is also close to the customer, and each TSP tour would have a short distance. Still, the total travel distance grows with $\sqrt{n}$. Thus, we will show an upper bound on α_λ, but there is also a lower bound.

In the home health care setting, the case $k = o(n)$ applies when there are one or a few "depots", i.e., work places of the health care organization where the employees have to go before starting their shifts. The case $k = \beta n$ applies to the situation where the health care professionals leave their home to go directly to their first client. This setting is actually the one we work with in Chapter 7, however, we will go more in depth into the first case due to the tractability.

First, we need some definitions before presenting the results:

Definition 6.2.1 (Complete convergence of random variables)**.** *We say that the sequence of random variables $(Y_n)_{n\geq 1}$ converges completely to the random variable Y (write $Y_n \to Y$ c.c.) when, for all $\varepsilon > 0$,*

$$\sum_{n=1}^{\infty} \mathbb{P}(|Y_n - Y| > \varepsilon) < \infty \tag{6.10}$$

Definition 6.2.2 ($O(g(n))$ and $o(g(n))$)**.** *For the functions $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ we say that*

(i) $f(n) = O(g(n))$ when there is an $M > 0$ and $N > 0$ such that $f(n) \leq Mg(n)$ for all $n > N$

(ii) $f(n) = o(g(n))$ when for all $\varepsilon > 0$ there is an $N > 0$ such that $f(n) \leq \varepsilon g(n)$ for all $n > N$.

We state without proof that complete convergence is a stronger form of convergence than almost sure convergence. The main results are presented in Theorem 2.

Theorem 2. *Let $D = \{D_1, \ldots, D_k\}$ be the depots and $P = \{P_1, \ldots, P_n\}$ be the customers, both independent and uniformly distributed inside $[0, 1]^2$ and let $L(D, P)$ be the length of the optimal set of TSP tours solving the MDVRP. Then we have*

(i) $\lim_{n\to\infty} \frac{L(D,P)}{\sqrt{n}} = \alpha_\lambda \quad$ *c.c., if* $k = \beta n$, $\beta > 0$

(ii) $\lim_{n\to\infty} \frac{L(D,P)}{\sqrt{n}} = \alpha(TSP) \quad$ *c.c., if* $k = o(n)$.

where α_λ *is some constant and* $\alpha(TSP) \approx 0.76$. *The following bounds are known for* α_λ.

(iii) $\min\left\{\alpha(TSP), \frac{2\alpha(TSP)}{\sqrt{1+\beta}}\right\} \geq \alpha_\lambda \geq \frac{1}{2\sqrt{1+\beta}}\left(1 + \frac{1}{4(1+\beta)}\right)$

From Theorem 2(iii), we see that α_λ decays with speed $1/\sqrt{\beta}$. To prove Theorem 2(ii), we first need a lemma, where we will show that when $k = o(n)$, the difference between the length of the TSP route (without depots) and the MDVRP route is small, namely $o(\sqrt{n})$.

Lemma 3. *Let* D, P *and* $L(D,P)$ *be defined as before, and let* L_n *be the length of the shortest TSP tour through the points in* P. *Then we have that* $|L_n - L(D,P)| = o(\sqrt{n})$.

Proof. We will show that $L_n - L(D,P) = O(\sqrt{k})$ and that this implies that it is also $o(\sqrt{n})$. Consider the optimal MDVRP route that has length $L(D,P)$ as a graph and add a feasible TSP tour γ through the points in D to that graph. The added tour has length $O(\sqrt{k})$, which we know from Theorem 1. For the resulting graph, note that every vertex has even degree (where the degree of a vertex is the number of edges incident to the vertex), since every depot vertex had even degree before adding the TSP tour and the degree increases with 2 after, while the degree of the customer vertices has not changed. Since the graph is also connected, it follows from Euler's theorem that there is an Euler cycle (a closed path that visits every edge exactly once). In fact, we can consider the following Euler cycle. We start at a depot, follow all TSP tours (visiting customers) associated with the optimal MDVRP route, and move onto the next depot in tour γ. We can create a feasible TSP route by starting at an arbitrary $p \in P$, following the Euler cycle but bypassing the depots by taking shortcuts to other points in P that are connected to other depots (see Fig. 6.1 for an illustration). This TSP route, and hence also the optimal one, has length at most $L(D,P) + O(\sqrt{k})$.

Thus we know that there exist $M > 0$, $N_1 > 0$ such that $L_n - L(D,P) \leq M\sqrt{k}$ for $n > N_1$. Letting $\varepsilon > 0$, we know that there is an N_2 such that $k \leq \frac{\varepsilon^2}{M^2} n$ for $n > N_2$ (since $k = o(n)$). But then for $n > \max(N_1, N_2)$, $L_n - L(D,P) \leq M\sqrt{k} \leq \varepsilon\sqrt{n}$, and hence $L_n - L(D,P) = o(\sqrt{n})$.

To prove the statement with the absolute value sign, we simply show that $L(D,P) \leq L_n + 2\sqrt{2}$. Start with the optimal TSP tour through the points in P and delete one edge. We can create a feasible MDVRP route by connecting both customers from that edge to a depot. The two new edges have length at most $\sqrt{2}$ (the diagonal of the square). ■

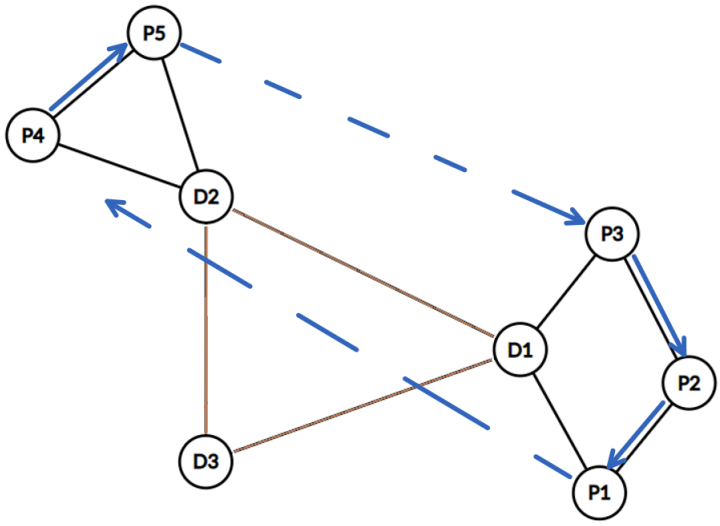

Figure 6.1: Starting from an arbitrary point, say P_4, we can go to P_5, then take the shortcut to P_3. Then we continue the route like this, bypassing all depots, following the blue route. The brown tour is γ, and the black tours is the optimal solution to the MDVRP.

Proof of Theorem 2(ii). Let $\varepsilon > 0$. We have

$$\sum_{n=1}^{\infty} \mathbb{P}\left[\left|\frac{L(D,P)}{\sqrt{n}} - \alpha(TSP)\right| > \varepsilon\right] \tag{6.11}$$

$$= \sum_{n=1}^{\infty} \mathbb{P}\left[\left|\frac{L(D,P) - L_n + L_n}{\sqrt{n}} - \alpha(TSP)\right| > \varepsilon\right] \tag{6.12}$$

$$\leq \sum_{n=1}^{\infty} \mathbb{P}\left[\left|\frac{L(D,P) - L_n}{\sqrt{n}}\right| + \left|\frac{L_n}{\sqrt{n}} - \alpha(TSP)\right| > \varepsilon\right] \tag{6.13}$$

$$\leq \sum_{n=1}^{\infty} \mathbb{P}\left[\left|\frac{L(D,P) - L_n}{\sqrt{n}}\right| > \frac{\varepsilon}{2}\right] + \mathbb{P}\left[\left|\frac{L_n}{\sqrt{n}} - \alpha(TSP)\right| > \frac{\varepsilon}{2}\right] \tag{6.14}$$

where eq. (6.13) holds because of the triangle inequality and eq. (6.14) holds because either the first term or the second term inside $\mathbb{P}(.)$, in eq. (6.13), has to exceed $\frac{\varepsilon}{2}$ in order for that event inside $\mathbb{P}(.)$ to hold, and the probability of either event is smaller than the sum of the probabilities. The series of the second term is finite, since there is complete convergence in the asymptotics of the TSP (Yukich, 1998 [22]), although we only showed almost sure convergence in Theorem 1. The series of the first term is also finite, since there are finitely many non-zero terms because of Lemma 3 (noting that $|L(D,P) - L_n| \leq \frac{\varepsilon}{2}\sqrt{n}$ for n larger than some N).

■

7 Machine learning of costs in the VRPTW

In this chapter we use a ML model to approximate costs for the HHCRSP, which we specified in Chapter 3, and which can better be named as the VRPTW as we neglect qualification levels of employees.

7.1 Approach and outline

We will generate a large dataset of 1200 instances and run each of them through paGOMEA to obtain the optimal solution and the corresponding cost measures (travel time, waiting time, and shift overtime). Then we extract features from those instances, some of which are straightforward characteristics, and some of which require some computation, although the computation should be easy. In particular, the computation of all the features from an instance should be significantly faster than running paGOMEA itself on that instance. We split the dataset into a training set of 90% and a test set of 10%. Then we train a Random Forest Regressor (RFR) on the training set, using one of the three cost measures as the target variable/actuals. To obtain predictions on a different cost measure, we have to retrain the model on that target and we will use the same features, although some features do not seem relevant for all three cost measures. We use `scikit-learn` with default RFR parameters. Subsequently, we make predictions using the trained RFR model on the test set and compare the predictions to the real values (according to paGOMEA). We make a visual comparison, but we also use the following performance metrics. If y_i is the prediction on instance i, and a_i is the actual, and if the test set has size n, then

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - a_i|}{a_i} \tag{7.1}$$

$$\text{WAPE} = \sum_{i=1}^{n} w_i \frac{|y_i - a_i|}{a_i}, \quad w_i = \frac{a_i}{\sum_{i=1}^{n} a_i} \tag{7.2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - a_i| \tag{7.3}$$

For travel time predictions, the MAPE is used in the literature, so we will also use this, but the MAPE is sensitive to (close to) zero actuals. Since total waiting times and shift overtimes (actuals) can be near zero for many instances, we will use WAPE for those costs instead. For all three costs we will also use the MAE, as this metric is well-interpretable (e.g., if MAE = 3 for waiting time, it implies that the predictions are on average 3 minutes off). To get insight into which features have the greatest impact on the ML model, we use feature importance scores that are built into the RFR algorithm.

For a brief outline of the chapter, in Section 7.2 we give a short introduction to evolutionary algorithms. Subsequently, in Section 7.3, we describe how we generate instances for our dataset. In Section 7.4, we describe the features we implemented and finally, in Section 7.5 we will show the results.

7.2 Evolutionary algorithms and paGOMEA

Evolutionary algorithms (EAs) are a broad class of population-based optimization algorithms that are inspired by the evolution of species in nature. Given a search space and a blackbox "fitness" function, such an algorithm seeks to find the point in the search space (i.e. the optimal solution) that has maximal (or minimal) fitness function value. In our application, the fitness function is a weighted sum of the three cost measures. Such algorithms are population-based, meaning that we start with a population of candidate solutions and we iteratively let them evolve through generations, in order to let some of the offspring to converge to the optimal point. To follow this procedure for a particular optimization problem, an EA needs the following components (Eiben, 2003 [23]):

- a representation of candidate solutions,
- a mechanism to select parents out of the population to create offspring solutions,
- variation operators for reproduction, namely recombination and mutation, so that offspring is different from the parents,
- and a survivor selection mechanism, to determine which solutions among parent and child populations survive into the next generation.

In traditional EAs, a fixed fitness landscape is assumed, and parent selection, survivor selection, and variation operators are fixed rules that do not change during the search process. To address problems with a changing fitness landscape, model-based EAs (MBEAs) were introduced, which are EAs for which the evolutionary mechanisms are replaced by machine learning models so that they adapt during the search process (Cheng et al. (2018) [24]). The novel paGOMEA is such an MBEA, but we will not go into the details of this algo-

rithm. For details, see [4]. We only note that we use all default settings, except that the population size is 600 instead of 200 (as we found this modification to give significantly better solutions) and that we use the following weights for the fitness function:

$$\text{fitness} = 7 \times \text{shift overtime} + 2 \times \text{travel time} + 1 \times \text{waiting time}, \qquad (7.4)$$

rather than equal weights.

7.3 Instance generation

How we generate instances is described next. There is one depot, but the distance from the depot to each customer is set to zero, so we effectively do not have a depot. This represents a situation where care workers travel directly to the first client and traveling from home to the first client is not taken into account. The travel time between customers is the Euclidean distance. All quantities have a fixed value in some instance. However, for each instance, some values are generated according to some probability distribution, as indicated.

Generation of customers

For the generation of customers we use the following data.

- The length of the diagonal of the square region has a uniform distribution between 5 and 15. The bottom left corner is the origin of the coordinate system.
- The square is divided into 3 by 3 smaller subsquares (like a chess board). To generate the location of one customer, we first choose a subsquare according to some probability distribution. This probability distribution is generated by randomly choosing 9 numbers $w_1, \ldots, w_9$ from $\{0, 1, \ldots, 100\}$, which give probabilities $p_i = w_i / \sum_{j=1}^{9} w_j$. The distribution is fixed for one instance. Next, a uniformly random point is chosen inside the chosen subsquare. This procedure is repeated for the remaining customers.
- The number of customers is uniformly random between 29 and 60. Having more than 60 customers introduces excessive computation time.
- To generate time windows for customers, we divide the day into morning (7AM-12 noon), afternoon (12 noon - 5PM), and evening (5PM-10PM). It suffices to generate the midpoint and the length of the time window. The midpoint of a time window is in the morning with probability (w.p.) 0.5, in the afternoon w.p. 0.05 and in the evening w.p. 0.45. The midpoint is at a uniformly random moment inside the chosen part of the day. Next, the length of the time window is 30 min. w.p. 0.05, 60 min.

w.p. 0.85 and 120 min. w.p. 0.1. The time window cannot start before 7AM or end after 10PM and is cut short at these times when appropriate. The start and end times are rounded to the nearest integer. We use the same length of time windows for all customers in one instance.

- The (on-site) service time of each customer has a lognormal distribution with mean 20.91 and standard deviation 12.34. This distribution was fit on a dataset of service durations from a health care organization.

Generation of shifts

To generate shifts, the following parameters are used.

- The duration of all shifts is 5 hours.
- The number of shifts is determined for the morning, afternoon, and evening separately, as follows. First, the customers are each assigned to the three periods according to where the midpoint of their time window lies (when a midpoint is at a boundary, it is assigned to the earlier period). Then the number of shifts for a particular period is $\lceil w/300 \rceil$ where w is the workload in minutes present in the period, which is in turn given by $0.76\sqrt{|J|A} + \sum_{j \in J} S_j$ (predicted travel time plus service demand). Here J is the set of customers assigned to the corresponding period. We are assuming the problem is close enough to a VRP so that the asymptotic approximation of travel time is accurate.
- The morning, afternoon and evening shifts start at 7AM, 12 noon, and 5PM, respectively.

7.4 Features used by the ML model

We use various types of features in our ML model, and in total they are about 100 features. These can be roughly categorized as follows: VRP/distance based, time (window) based, based on a combination of time and distance (spatio-temporal), based on simple heuristic solutions (routes), queuing theory based, and workload based. In the following subsections we present all features used corresponding to each category.

7.4.1 VRP based

From the literature on traveling distance estimation in VRPs, we will use Mei (2022) and Akkerman (2022) as inspiration to build some features for the ML model. Since these authors created models to predict travel time, we do not expect to improve waiting time and shift overtime predictions by drawing

inspiration from these papers. Instead, we can hope for improvement in travel time prediction, although the VRP considerably differs from the HHCRSP.

Mei (2022)

Mei used a regression model from previous literature,

$$\text{total VRP distance} = k_l\sqrt{An} + k_m\bar{r}m \tag{7.5}$$

where k_l and k_m are regression coefficients, $\bar{r}$ is the average distance to the depot and m is the number of tours required, taking into account the vehicle capacity. The second term does not apply to us, since we effectively do not have a depot in our instances. The first term, that estimates the distance traveled between customers, is of importance to us, and we will use $\sqrt{An}$ as a feature. But we wanted to use this feature regardless, since in Chapter 6 we have seen that the length of the TSP solution as well as the MDVRP solution is asymptotically proportional to this quantity.

Mei's contribution was to use a measure of dispersion and clustering, the Average Nearest Neighbor Index (ANNI), to fine-tune the regression coefficient k_l. It is defined by

$$\text{ANNI} = \frac{\text{ANND}_{obs}}{\text{ANND}_{exp}} = \frac{\frac{1}{n}\sum_{i=1}^{n} d_{ij_1}}{1/(2\sqrt{An})} \tag{7.6}$$

Here ANND_{obs} is the observed average nearest neighbor distance, i.e. the average distance between a point and its nearest neighbor, while ANND_{exp} is what you would expect this quantity to be, based on some probability theory, if points are uniformly distributed in the region. If $\text{ANNI} > 1$, then the points are more dispersed than you would expect from uniformly distributed points, and if $\text{ANNI} < 1$, they are more clustered. We use ANNI as a feature.

Akkerman (2022)

Like in the paper of Mei, Akkerman also worked with a capacitated VRP, and the author built many features that involve taking vehicle capacities into account. Since the concept of capacities does not apply to our situation, we will not use the related features. Moreover, we will also not use features that involve the depot since we effectively do not have one, but we will use all other features that were presented in that paper. We include all features in the model as we do not employ feature selection methods. The features are the following.

(1) The number of customers, n.

(2) Of the smallest enclosing rectangle around the customers, we take the area, perimeter, height, and width (so 4 features).

(3) Of the convex hull around all locations, take the area and perimeter.

(4) Average distance between customers, that is, $\frac{2}{n^2-n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}$.

(5) Average distance between customers and the rectangle centroid (the center of the enclosing rectangle).

(6) Average distance between customers and the customer centroid, which has x coordinate equal to the average of customer x coordinates, and similar for the y coordinate (i.e., center of mass).

(7) Perform a coordinate transformation by simply shifting the origin to the rectangle centroid. Let α_i be the angle between the positive x-axis and the vector from the origin to customer i (i.e. if the new coordinates of customer i is $(\tilde{x}_i, \tilde{y}_i)$, then $\alpha_i = \text{atan2}(\tilde{y}_i, \tilde{x}_i)$, see Wikipedia for atan2). The feature is the variance of $\{\alpha_1, \ldots, \alpha_n\}$.

(8) The same as (7) but using the customer centroid instead of the rectangle centroid.

(9) Variance of $\{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ where (x_i, y_i) are the original coordinates of customer i.

(10) Variance of $\{x_1 \cdot y_1, \ldots, x_n \cdot y_n\}$.

(11) Variance of $\{\text{distance from customer } i \text{ to the customer centroid} : i = 1, \ldots, n\}$.

(12) The same as (11) but now for the distance to the rectangle centroid.

(13) Variance of $\{d_{ij} : i = 1, \ldots, n-1 \text{ and } j = i+1, \ldots, n\}$ (i.e. the set of all distances between customers).

(14) The number of customers within a radius of $0.5M$ from the customer centroid, where M is the distance between the circle midpoint (customer centroid in this case) and the customer farthest away. For three other features, use a radius of $0.75M$ and use the rectangle centroid as the circle midpoint with radius $0.5M$ and $0.75M$.

(15) Use a 10×10 rectangle partitioning (see Fig. 7.1 for an example with 2×2 partitioning). Take the average distance between all centers of activated rectangles, i.e., rectangles that contain customers.

(16) Using rectangle partitioning, the average of the set of all distances between customers that are in the same rectangle.

(17) For two more features, use (15) and (16) but with 15×15 rectangle partitioning

(18) The number of shifts, n_s (Akkerman uses the minimum required number of vehicles in relation to capacity constraints).

(19) We create n_s geographical clusters of customers by first choosing n_s customers ('seeds'). From the convex hull in 3., choose as many vertices as possible as seeds (though not more than n_s), such that there are (at least) two non-chosen vertices in between two chosen vertices when walking along the boundary from one to the other chosen vertex. If at this point we have less than n_s seeds, we iteratively choose new seeds by taking the points that have the largest distance to its closest seed. Then we cluster all remaining points to seeds by geographic proximity. As a feature, compute the average distance between centroids (i.e., centers of mass) of customer clusters.

(20) Using the clusters, compute the average of the set of all distances between customers that are in the same cluster.

(21) Using the clusters, compute the average of the set of all distances between customers and their cluster centroids.

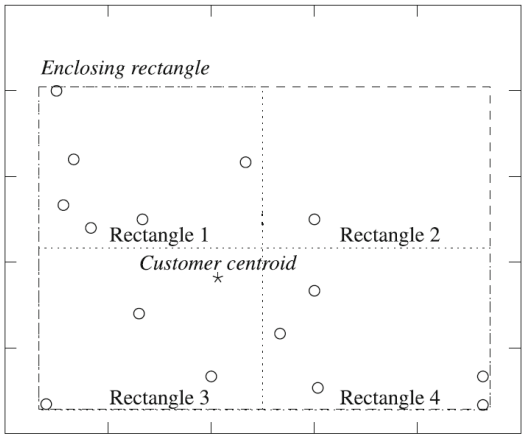

Figure 7.1: Example of 2×2 rectangle partitioning, from Akkerman's paper.

Features 7 and 8 aim to express a measure of dispersion of the customers. Features 8-13 express "geographical variance", that is, it is also a measure of dispersion. The rectangle partitioning mechanism in 15-17 and the clustering mechanism in 19-21 are two different ways to assign customers to different vehicles/shifts using simple heuristics, and with them we can compute distance related metrics corresponding to those simple assignments. The algorithm for choosing seeds was not properly specified in the paper of Akkerman, so we have used our own simple to compute heuristic.

7.4.2 Only time window related

The following features only take time windows of the customers into account.

- Time window dispersion: let $m_1, \ldots, m_n$ be the midpoints of the n time windows, sorted from early to late, and let $\ell_1, \ldots, \ell_{n+2}$ be an equidistant

grid starting at 420 (i.e., 7AM) and ending at 1320 (10PM). Then the feature is $\sum_{i=1}^{n} |m_i - \ell_{i+1}|$. A low value indicates high dispersion, as the time windows are more equidistantly spread over the day. Note that we create margins since we do not expect the midpoints to be close to the endpoints of the day.

- Standard deviation of the time window midpoints, which are determined per period (so three features).

Both features are a measure of temporal dispersion. With more clustered time windows, there may be more waiting time involved, as serving one customer can cause another customer with an overlapping time window to have to wait. But solving VRPTWs involves a complex interplay between spatial and temporal characteristics of the data, hence using features that take only space or only time into account is insufficient. Features are needed that capture this interplay, i.e., spatio-temporal features.

7.4.3 Spatio-temporal based

We construct a matrix that is anticipated to contain information about probabilities of 'successful' transitions from one customer to another. In particular, define Γ_{ij} as the probability that, given that the server has arrived at customer i at a uniformly random moment within its time window (i.e., on time), it will arrive within the time window of customer j, if the server chooses to go there. Thus,

$$\Gamma_{ij} = \mathbb{P}(U + S_i + d_{ij} \in [\text{Start}_j, \text{End}_j]) \tag{7.7}$$

where $U \sim \mathcal{U}([\text{Start}_i, \text{End}_i])$. This probability is, in fact, the length of the overlap between $[\text{Start}_i + S_i + d_{ij}, \text{End}_i + S_i + d_{ij}]$ and $[\text{Start}_j, \text{End}_j]$ divided by the length of the time window of customer i.

We derive several features from this matrix. Define $\tilde{d}_{ij} = (\bar{d} - d_{ij})/\bar{d}$, and define $D_i = \sum_{j=1}^{n} \tilde{d}_{ij}\Gamma_{ij}$. Let $\hat{D}_i = \sum_{k=1}^{3} \Gamma_{ij_k}$. Both D_i and $\hat{D}_i$ (with high values) indicate how easy it is to go from i to another close customer that can be served on time. To aggregate over all customers, we have the following features.

- The number of non-zeros in $D_1, \ldots, D_n$
- The number of non-zeros in $\hat{D}_1, \ldots, \hat{D}_n$.
- The sum $\sum_{i=1}^{n} D_i$.
- The sum $\sum_{i=1}^{n} \hat{D}_i$.
- Define $E_i = d_{ij_k} - d_{ij_1}$ with $k > 1$ being the smallest number such that $\Gamma_{ij_k} > 0.5$. This indicates the extra distance that one has to travel from

i to the closest customer that can feasibly be reached on time, compared to the nearest neighbor. The feature is $\sum_{i=1}^{n} E_i$.

- Using the spatial clusters of Akkerman [10], we try to make chains of customers within each cluster. Pick a cluster and a random member i within, and pick the closest customer j within that cluster with $\Gamma_{ij} > 0.7$ and add this customer to the chain. Complete the chain in this way and count how many customers are excluded at the end. Do this for all other clusters as well and count the total number of excluded customers.

Next we will construct a feature that captures the following idea: when two customers are close in space and "close" in time (i.e. the time windows follow closely when traveling from one to the other), this is very convenient. When they are both far in time and space, this is somewhat convenient (the optimal route will likely not connect them anyway). When distances are close and time windows are far apart, this is undesirable, and when it is the other way round, it is even more undesirable and indicates a hard instance.

Let $m_1, \ldots, m_n$ be the midpoints of the time windows. Define

$$\Delta_{ij} = \frac{|(m_i + S_i) - m_j|}{\max_{i,j} |(m_i + S_i) - m_j|} \tag{7.8}$$

This measures how distant the time windows are of customer i and j, taking into account the service time of i. Also normalize all distances d_{ij} and call this $\hat{d}_{ij}$ (so that $\hat{d}_{ij} \in [0, 1]$). Let

$$M_{ij} = \min(|d_{ij} - \Delta_{ij}|, |d_{ji} - \Delta_{ji}|) \tag{7.9}$$

$$\tilde{D}_{ij} = \min(\sqrt{2}, \hat{d}_{ij} M_{ij}). \tag{7.10}$$

Here M_{ij} indicates (with a low value) that customer i closely follows j in time (or the other way round) and they are also close in space, or that they are far in time as well as far in space. Then $\tilde{D}_{ij}$ captures the idea that we just described. The feature becomes $\sum_{i,j} \tilde{D}_{ij}$ with low values indicating easier instances.

7.4.4 Based on simple heuristic solutions

We build simple heuristic solutions that are easy to calculate, in order to get more information about how hard the instance is. However, note that a good objective value indicates a simple instance, but a bad objective value does not necessarily indicate a hard instance, because a good value could be possible with an intelligent solution that paGOMEA can find, that is outside the grasp of the heuristic. From the following routes, we collect the total waiting time, the total overtime, and the travel distance as features.

- Local FCFS: We use the clusters created from seeds as in Akkerman [10]. The clustering was done on the basis of geographic proximity, and there

are as many clusters as shifts. In each cluster, the server will follow an FCFS policy (the order is based on the time window due time).

- Global FCFS: All customers are ordered on the time window due time. One server will serve the first $\approx n/n_s$ (rounded appropriately) customers with an FCFS policy, another will serve the next n/n_s etc.
- Load-based FCFS: Let w be the estimated workload of the instance (see Section 7.4.6). Ordering the customers again on the time window due time, we let one server handle the first k customers, such that the workload of those customers does not exceed w/n_s. Then the next group of customers will be assigned to the next server in the same way. Note that the number of routes could now exceed n_s, but we can still compute the features.
- Period-wise FCFS: The previous heurstics were based on multiple single-server queues, while this feature involves a multi-server queue. Again, sort the customers on time window due time. But we make one adjustment. If two consecutive customers have time windows that largely overlap, and the later customer has a much shorter service time than the earlier one, we want to swap these two customers. This is because the customer with the long service time can cause a delay for all subsequent customers. More precisely, we make this swap when the overlap is greater than 90% of a time window width and when the later customer has a service time smaller than 30% of the service time of the earlier customer. We loop through all customers and make the swaps appropriately.

  We will construct routes for each period of the day (morning, afternoon, and evening) separately. Let there be n_s^m, n_s^a and n_s^e shifts in the morning, afternoon, and evening, respectively (so that $n_s = n_s^m + n_s^a + n_s^e$). Assign the first n_s^m customers to each of the morning shifts one to one. Next, the morning shifts follow an FCFS policy until they refuse the next customer. They will refuse a new customer if they would otherwise be finished serving that customer after 12 noon, unless that customer has a due time before 10:20AM (otherwise, that customer has to wait long if it is served by an afternoon shift). Then an afternoon shift will take on this customer instead, and the morning shift will be done. FCFS will continue in the morning until all morning shifts are completed. Then the afternoon shifts will continue the FCFS policy until the same switch occurs in the transition from afternoon to evening, when the evening shifts will finish the policy. A similar refusal condition will be followed in this transition (also here with an exception for a customer having a due time before 15:20PM). In this way, we try to avoid as much overtime as possible in the creation of these routes.

7.4.5 Queueing theory based

We will use the theory in Section 4.4. The M/M/s queue will be used as a simple mathematical model for each of the three parts of a day with s equal to the number of shifts available in the corresponding part of the day. We will build three features with this model (one for each period), and we will build three other features using the same principle but with the G/G/s queue.

For the M/M/s model, we assume that the subset of customers and their time windows in one part of the day are a realization of an underlying Poisson arrival process and that their service times are sampled from some underlying exponential distribution with rate μ. Based on the instance data, we make estimates of μ and of the load on the system a. For the mapping from the waiting time in the VRPTW to the M/M/s queue, an arriving customer in stationarity is assumed to be waiting if the waiting time is larger than T, where we take T as the average time window length of all customers in the subset. So we consider the arrival moment of a customer in this queue as the start of his time window, and we assume that the time window length is always T. Suppose that the subset of customers has size m. Then, for an approximation of the total (combined) waiting time for that subset of customers, we take $(m-1)\mathbb{E}(W-T)^+$ where this expression can be calculated with eq. (4.28) (note that the first customer has no waiting time).

To use that expression, we define the load a as the total service demand divided by the combined capacity of the shifts. The total on-site service demand is known, and the travel times are estimated using the asymptotic approximation. The capacity of shifts is 300 minutes times the number of shifts in the relevant part of the day, say n_s^m. Thus, we take

$$a = \frac{\sum_{i=1}^{m} S_i + 0.76 * \sqrt{mA}}{300 n_s^m} \tag{7.11}$$

where the S_i's are the service times of the subset of customers. Then we take $\lambda = as$ and $\rho = a/s$. Furthermore, for the expected service time in this queueing model, we use the average service time per customer,

$$\bar{S} = \frac{\sum_{i=1}^{m} S_i + 0.76 * \sqrt{mA}}{m} \tag{7.12}$$

and the service time of an arriving customer will have the exponential distribution with rate $\mu = 1/\bar{S}$. Now we can compute expression (4.28) for the M/M/s queue.

For the G/G/s queue, we take the same value for a and ρ. To compute η in expression (4.28) we need an expression for $\mathbb{E}S$, c_a^s and c_s^2. For $\mathbb{E}S$, we take $\bar{S}$. The value of c_s^2 will simply be the squared coefficient of variation of the relevant service times in the instance. For c_a^2, take the relevant customers and

sort them by time window start time. Then interarrival time i will be the start time of customer i minus the start time of customer $i-1$, $i = 1, \ldots, m$ and we take the squared coefficient of variation of these samples.

Note that the queueing models are vast oversimplifications of the VRPTW, the M/M/s more so than the G/G/s, since exponential distributions were not used in generating data. But these features still might give rough approximations of the actual waiting times. Also note that we in fact modeled the problem as a multi-server DTRP (but neglecting the dependency between successive service times).

7.4.6 Workload based

So far we have not taken the service time of customers into consideration. The service times are an important component of the workload, and the following features are related to this.

- service time mean per part of the day
- service time standard deviation per part of the day
- time window relative size mean: calculate the mean of $\{\frac{\text{End}_i - \text{Start}_i}{S_i} : i = 1, \ldots, n\}$, i.e., the relative size of the time window compared to the service time
- time window relative size standard deviation
- estimated workload: as before, take $0.76\sqrt{An} + \sum_{i=1}^{n} S_i$

7.5 Results

We will first explore the performance of the RFR on predicting the travel time of VRPTW instances, then the waiting time, and then the shift overtime. For each of the three costs, we also show some feature importance analysis, using Gini importance scores (see Section 4.5). When showing bar charts of feature importance scores, we only display those features that have importance score above a certain threshold. Otherwise, the chart will be cluttered with too many bars. We also want to briefly share the results obtained using a linear regression model with the same set of features, as some prefer this model due to its interpretability, but we focus on the RFR.

7.5.1 Travel time prediction

First, we are interested in the effectiveness of the VRP features from the Akkerman and Mei papers. For this purpose, we have generated a separate

dataset of 1200 instances, in the same way as described in Section 7.3, except for setting the time window length of customers to infinity, as well as setting the shift durations to infinity. The number of shifts is chosen to be a random integer from $\{5, 6, 7\}$. This effectively turns the problem into a VRP, as waiting times and shift overtimes will always be zero, and the travel time is the deciding factor in finding the optimal solution. Using that dataset, the RFR produces a MAPE of 5.1% and MAE of 1.9 on the test set. The performance on the test set is further illustrated in Figure 7.2.

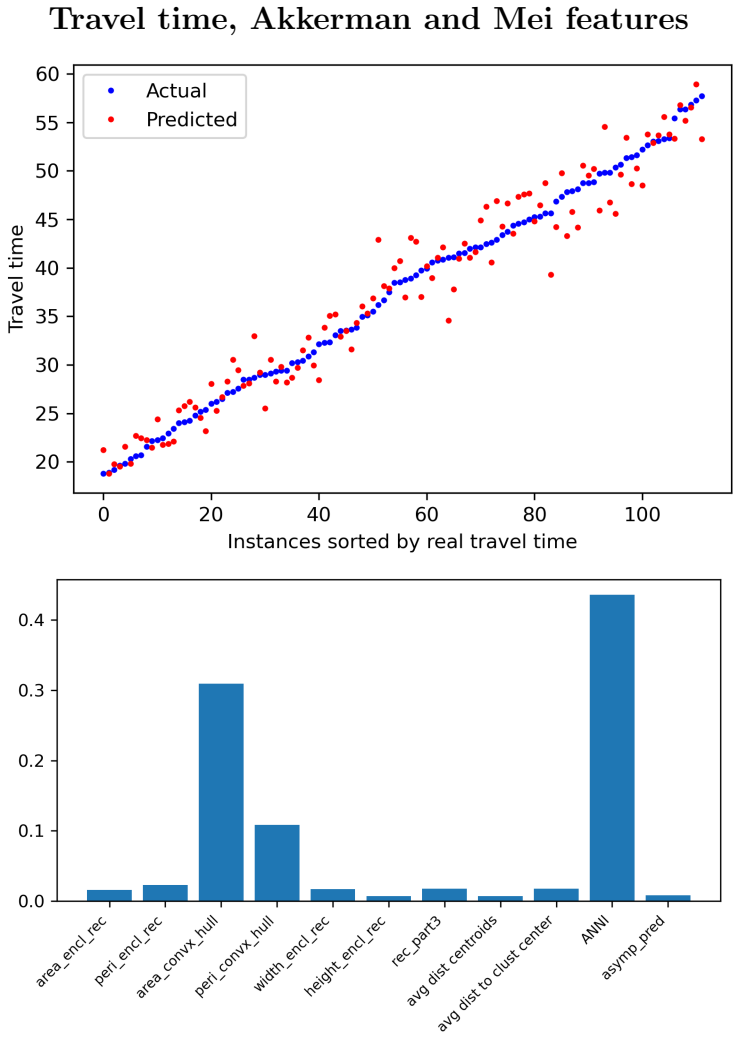

Figure 7.2: On a separate, VRP based dataset, this is the performance of the ML model with only the features from Akkerman and Mei, where we predict travel time on the test set. The corresponding feature importance scores are shown in the lower chart. The features from Akkerman are all but the two rightmost features.

To compare this to a benchmark, we train an RFR on the VRP-based dataset mentioned previously, using only one feature, namely the asymptotic predic-

tion, $\sqrt{An}$. See Figure 7.3 for the performance on the test set. The MAPE and MAE are 7.4% and 2.7, respectively. We see that the asymptotic prediction is already accurate, and that the additional features from the papers of Mei and Akkerman provide a modest improvement in accuracy. Interestingly, in Figure 7.2 we see that the asymptotic prediction is an unimportant feature, according to the Gini importance scores. The ANNI feature from the paper of Mei and two features related to the convex hull from the paper of Akkerman seem important features in this case.

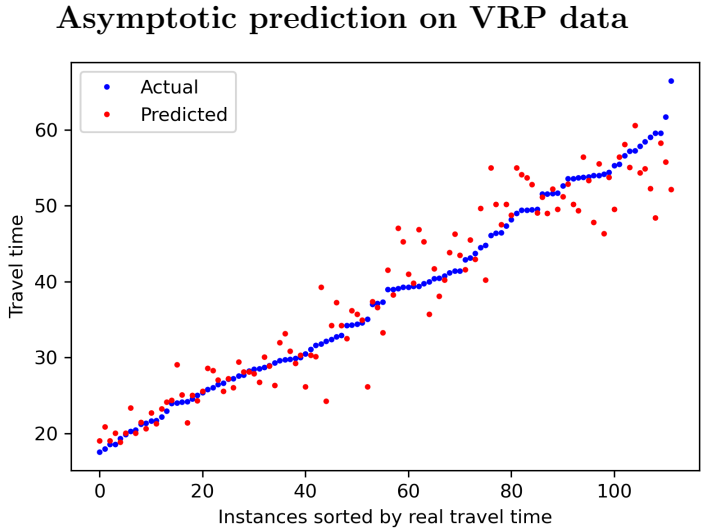

Figure 7.3: Performance on the test set, using the VRP based dataset, of the ML model with $\sqrt{An}$ as the only feature.

Now we will consider the original dataset, with finite time windows and shift durations. Using all the features to train the ML model (full model), the performance on the test set is illustrated in Figure 7.4. We see that the model performs decently across the entire range of real travel times, although perhaps a bit better when predicting instances with small travel times. The MAPE on the test set is 8.2%, whereas the MAE is 6.2 minutes. In the paper by Akkerman, the authors obtained a MAPE of 3.07% when they used feature selection (with an RFR as model) and 8.97% without feature selection (with linear regression). It is hard to compare our result with theirs, since the type of VRP problem studied differs, but our result is better than theirs if we look at their model without feature selection. Again, this can be due to the fact that the problem differs or due to our model having additional features. With linear regression we obtained a MAPE and MAE of 8.0% and 5.8, respectively.

In Figure 7.4 we can see that all features based on simple heuristic solutions, which estimate travel time, are among the most important features according to the Gini importance scores. But this can be misleading; see Figure 7.5. There we see that there is a clear multicollinearity among the features with

highest importance scores. Perhaps only one of those simple heuristic routes is needed in the model. It is interesting to note that none of the features from Akkerman's paper and Mei's paper seem to be important. Because customers have short time windows in our dataset, the VRPTW is closer to a scheduling problem than to a VRP. We have found this to be true by studying the optimal solutions. This may be the reason why VRP-based features are not important.

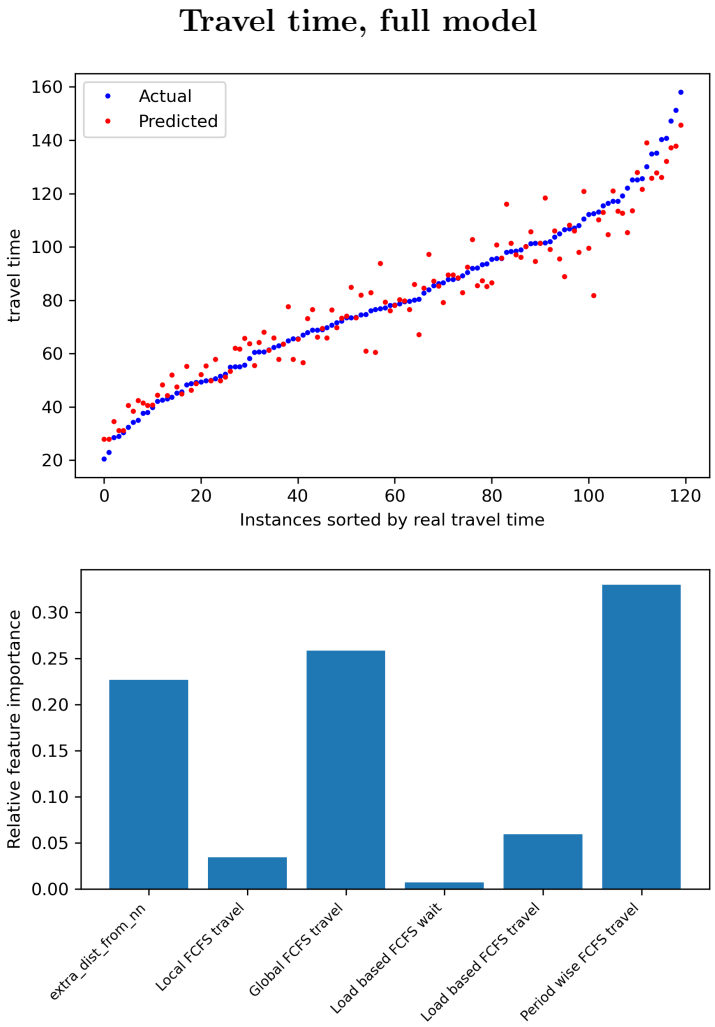

Figure 7.4: The performance of the full model on travel time predictions (upper) on the test set and the corresponding feature importance scores (lower). For the leftmost one, see Section 7.4.3

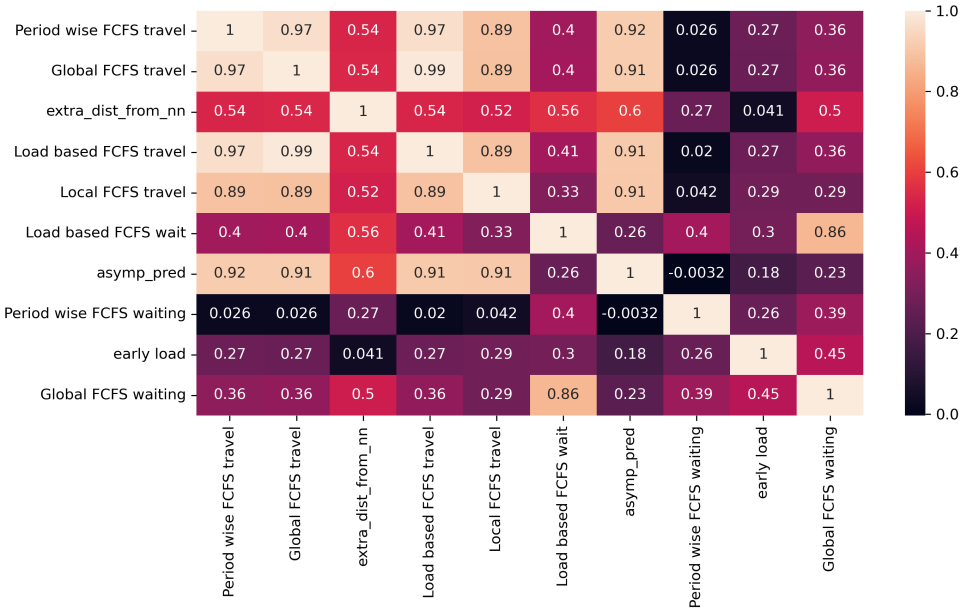

Figure 7.5: Correlation matrix of the ten features with highest importance scores when predicting travel time using the full model.

7.5.2 Waiting time prediction

For the predictions of the total waiting time, using the full model, see Figure 7.6. We see that there are many instances with (near) zero waiting time, and the model seems to predict them very well. For instances with significant positive waiting time, the model has more difficulty predicting the real waiting time accurately, although we do see a general trend of higher predictions for instances with higher real waiting times. The WAPE is 43% and the MAE is 13.6 minutes. We can compare these values to the following (naive) benchmark. Using the mean waiting time of the training data as a prediction for all instances in the test set, we get a WAPE and MAE of 130% and 41.4 minutes, respectively. Therefore, compared to this benchmark, the full model has a three-fold improvement in terms of MAE and a significantly improved WAPE score. With linear regression, we obtained a WAPE of 57% and an MAE of 18.2, which is significantly worse than the RFR.

Looking at the feature importance scores in Figure 7.6, we see that one feature seems to play a very prominent role in the model, the "Period-wise FCFS waiting time" (Section 7.4.4). The feature is indeed very important, but it is not wise to rely on this feature alone, and we can see this as follows. When building a very reduced model with that feature as the only feature in the model, we get a WAPE of 52% and a MAE of 16.7, a deterioration compared to the full model. Thus, the other features do contribute significantly to the model. And they do have predictive power, as we can see in Figure 7.7. In

this figure, we see the performance of a reduced model, where we delete the "Period-wise FCFS waiting time" feature and keep all other features. We can still see that the predictions follow the upward trend of the real waiting times. Here we have a WAPE of 60% and a MAE of 19.6, also much better than the benchmark. In Figure 7.7 (and also in Figure 7.6) we also see that the M/M/s and G/G/s queueing models provide valuable features to the ML model.

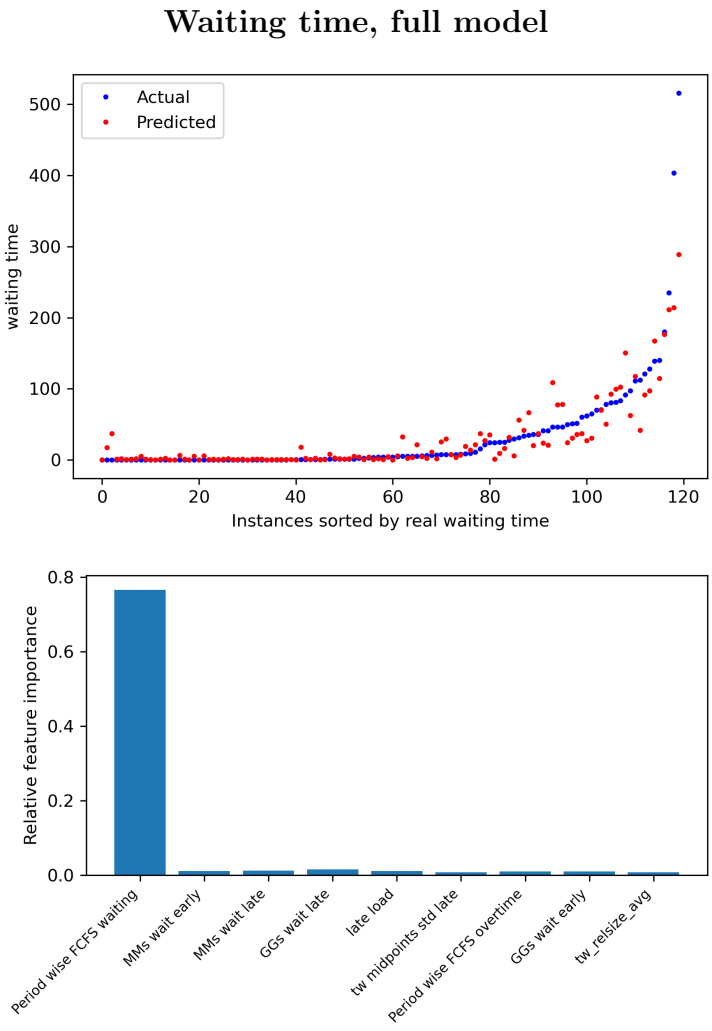

Figure 7.6: The performance of the full model on total waiting time predictions (upper) on the test set and the corresponding feature importance scores (lower).

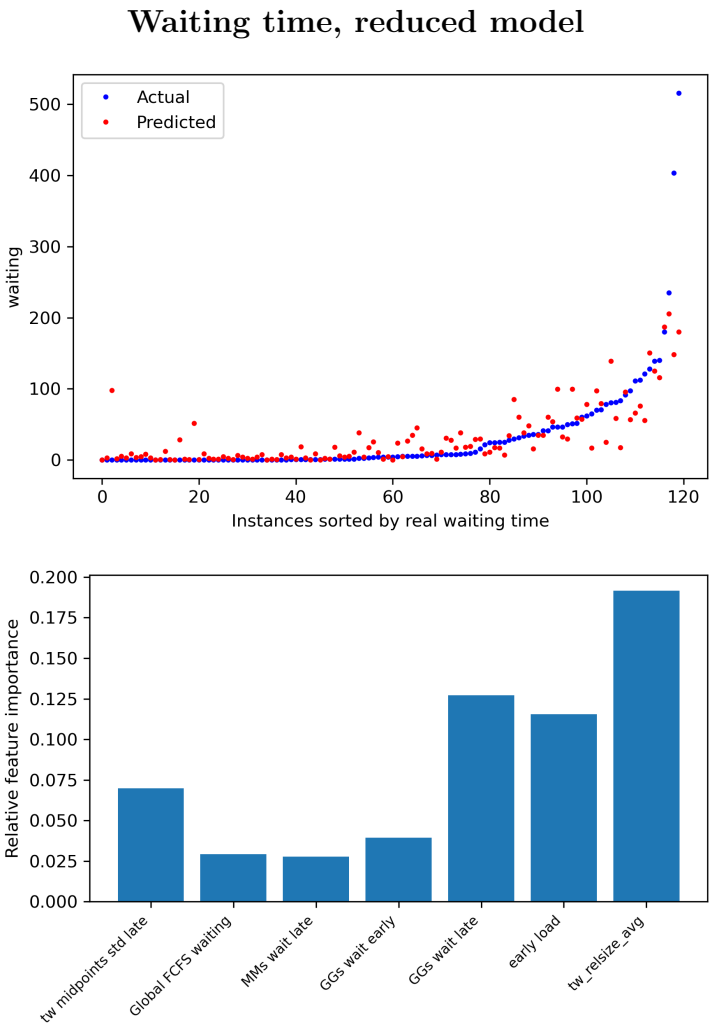

Figure 7.7: Performance on test set when predicting waiting time, using all features except "Period-wise FCFS waiting time" (upper) and the corresponding feature importance scores (lower).

7.5.3 Shift overtime prediction

With the full model, the predictions for the total shift overtime have a WAPE of 34% and a MAE of 4.1 minutes. Again, for an illustration of the performance, see Figure 7.8. We again see a large number of near-zero values of the real shift overtime, which the model predicts quite accurately, although once in a while the error is quite large. For the instances with larger than zero overtimes, we see that the predictions closely follow the real values most of the time, which can explain why the WAPE is smaller compared to the prediction of the waiting time (43%), although the model seems to systematically underpredict for those instances. The linear regression model gave a WAPE of 64% and an MAE of 7.7, again significantly worse than the RFR.

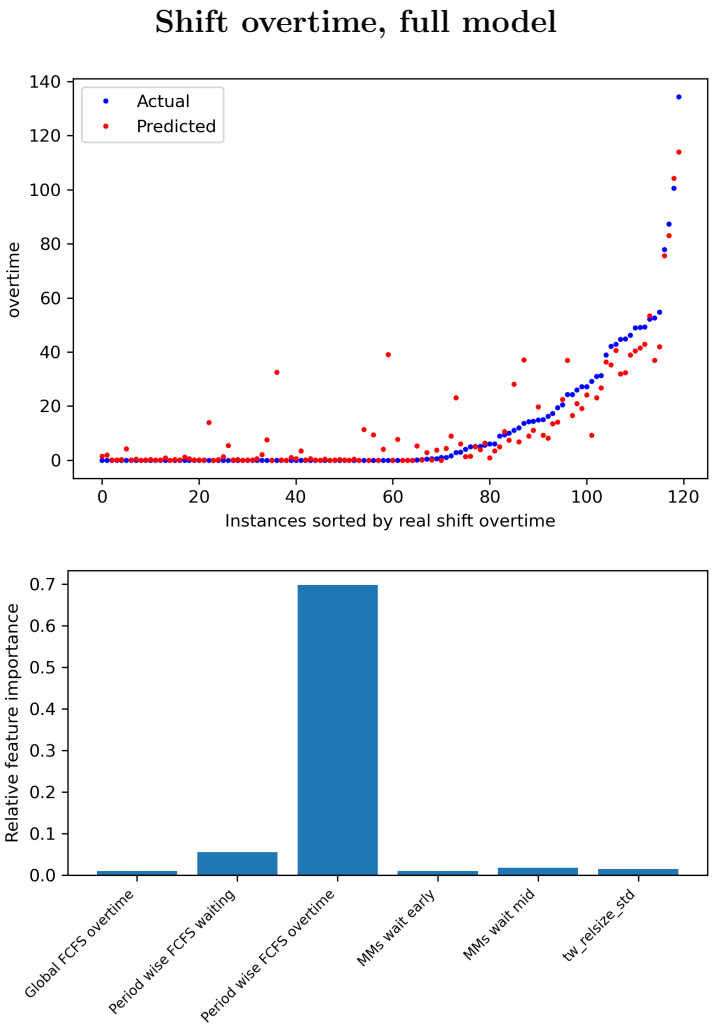

Figure 7.8: Performance on test set when predicting total shift overtime time, using all features (upper) and the corresponding feature importance scores (lower).

In Figure 7.8 we again see that the same simple heuristic route provides a very important feature, but this time the shift overtime feature of that heuristic route, instead of the waiting time. It seems that this heuristic route closely resembles the optimal route according to paGOMEA. We also see that again the M/M/s model seems to be valuable as well, although this is surprising because the M/M/s model aims to predict the waiting time.

8 Conclusion and Discussion

Three lines of research related to routing problems in the home health care were conducted. First, an asymptotic analysis of the optimal tour length of TSPs and VRPs with multiple depots (MDVRP) was performed, as the number of locations n to be visited, or customers, within a square of area A grows large. Second, we investigated the dynamic traveling repairman problem (DTRP) with the FCFS solution, in which we were interested in the expected waiting time of customers in stationarity. The DTRP with FCFS forms a model that is similar to an M/G/1 queue, but the arrival process occurs in a region, so travel times are involved. These travel times have an obvious dependency structure. Third, we built a machine learning model that can predict three cost measures of optimal solutions to the VRP with time windows (VRPTW). The optimal solutions were obtained with paGOMEA. We give suggestions for further research on the second and third topic.

In the asymptotic analysis, we have found that for the TSP as well as the MDVRP where the number of depots is small, namely $o(n)$, the convergence of the optimal tour length L_n is the same. In both cases we have $L_n/\sqrt{An} \to C$, with C a constant.

For the DTRP, Bertsimas and Van Ryzin stated that the system can be analyzed as an M/G/1 queue and the waiting time would be exactly given by the Pollaczek-Khinchine (PK) formula. We tested whether this was correct using a simulation study with different service region sizes and different loads on the system. With loads closer to one, we found a clear discrepancy, and we can conclude that the PK formula underpredicts. The relative error can be up to 4% of the value predicted by PK, for high loads and large service regions, leading to long travel times (compared to on-site service times).

The DTRP seems intractable to analytically derive the expected waiting time in stationarity. An attempt can be made in future research by viewing the waiting time as a discrete-time stochastic process, in particular a Lindley process, in order to solve the Lindley equation and obtain a type of transform (Laplace-Stieltjes) of the waiting time distribution. From this transform, the moments can be derived. To do this, one probably needs the joint distribution of successive travel distances, as this fully specifies the dependency structure of the service times. Then the DTRP can simply be seen as an M/G/1 queue with this specified dependency structure. This type of analysis was carried out

in [7] but with a different dependency structure. The joint distribution can be found in [8]. In particular, the covariance between successive travel distances can be of interest, which we studied in Section 4.3.

For VRPTW, we built the ML model using many features, some of which we got from previous studies, and some that we contributed, a total of ~ 100. For predictions of the total travel time, we got a MAPE of 8.2%, which is worse compared to the result of Akkerman [10], who had 3.07%. But they obtained that result using feature selection, so perhaps we should compare our model with their model without feature selection, for which the MAPE was 8.97%. Nevertheless, the problem studied was not the same, so the results are hard to compare. With waiting time and shift overtime predictions, there is no previous study to compare it to, but we had a WAPE of 43% and 34%, respectively. The MAE was 13.6 and 4.1, respectively. Although the WAPE scores in this case seem high, from the perspective of MAE the models seem useful. An average error of 4.1 minutes on the combined shift overtime across all shifts seems very reasonable, and the same can be said for the total waiting time. Overall, the model seems to be useful for practical applications.

We built some features by analyzing the VRPTW almost as a DTRP. These features seemed important according to the feature importance scores, so the DTRP seems like a useful model for the VRPTW. Furthermore, there was one very important feature for waiting time predictions, and one very important one for shift overtime predictions, both features engineered using a simple heurstic route ("period-wise FCFS"). The way these two features were engineered was based on how the optimal solutions looked like, and this in turn depended on how instances were generated. To be specific, in our dataset, the lengths of time windows are short, which causes the travel time to play a negligible role for paGOMEA in finding the optimal solutions, as there is not much leeway to optimize travel time since this would introduce excessive waiting time. Thus, the optimal solutions looked like an FCFS policy in a multi-server queue. If time windows are longer, the problem becomes closer to a VRP and looks less like a scheduling problem. If instances are generated differently, these two features may not work properly. Thus, the ML may not be very generalizable, for example to instances provided by a health care organization.

For further research on the third topic, one can employ sophisticated feature selection techniques to improve the ML model, as this made a big difference in Akkerman's ML model. We do not recommend performing feature selection based on the features' Gini importance scores, as we found this unreliable. For example, deleting the most important feature for travel time predictions actually improved the model. Although this feature was highly correlated with other features, Gini importance is not influenced by multicollinearity, according to [16]. Furthermore, some improvement may be gained using hy-

perparameter tuning, as we only tested the default hyperparameters of the Random Forest Regressor. We implemented some spatio-temporal features, but they did not perform well. We invented those features, but perhaps inspiration for more sophisticated spatio-temporal statistics can be drawn from [25].

Bibliography

[1] "Population structure and ageing." https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Population_structure_and_ageing, 2022. Eurostat. Last date accessed: 28-01-2023.

[2] J. H. Maarse and P. P. Jeurissen, "The policy and politics of the 2015 long-term care reform in the netherlands," *Health Policy*, vol. 120, no. 3, pp. 241–245, 2016.

[3] D. Moeke and R. Bekker, "Capacity planning in healthcare: finding solutions for healthy planning in nursing home care," *Integrating the Organization of Health Services, Worker Wellbeing and Quality of Care: Towards Healthy Healthcare*, pp. 171–195, 2020.

[4] Y. Clapper, J. Berkhout, R. Bekker, and D. Moeke, "A model-based evolutionary algorithm for home health care scheduling," *Computers & Operations Research*, vol. 150, p. 106081, 2023.

[5] D. J. Bertsimas and G. Van Ryzin, "A stochastic and dynamic vehicle routing problem in the euclidean plane," *Operations Research*, vol. 39, no. 4, pp. 601–615, 1991.

[6] M. Livny, B. Melamed, and A. K. Tsiolis, "The impact of autocorrelation on queuing systems," *Management science*, vol. 39, no. 3, pp. 322–339, 1993.

[7] I. J.-B. F. Adan and V. G. Kulkarni, "Single-server queue with markov-dependent inter-arrival and service times," *Queueing Systems*, vol. 45, pp. 113–134, 2003.

[8] L. E. Miller, "Joint distribution of link distances," in *2003 Conference on Information Sciences and Systems*, pp. 1–2, 2003.

[9] X. Mei, K. M. Curtin, D. Turner, N. M. Waters, and M. Rice, "Approximating the length of vehicle routing problem solutions using complementary spatial information," *Geographical Analysis*, vol. 55, no. 1, pp. 125–154, 2022.

[10] F. Akkerman and M. Mes, "Distance approximation to support customer selection in vehicle routing problems," *Annals of operations research*, pp. 1–29, 2022.

[11] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021.

[12] D. Bertsekas and R. Gallager, *Data networks*. Prentice-Hall, 1987.

[13] J. Abate, G. L. Choudhury, and W. Whitt, "Exponential approximations for tail probabilities in queues, i: waiting times," *Operations Research*, vol. 43, no. 5, pp. 885–901, 1995.

[14] W. Whitt, "Approximations for the gi/g/m queue," *Production and Operations Management*, vol. 2, no. 2, pp. 114–161, 1993.

[15] W. Whitt, "Understanding the efficiency of multi-server service systems," *Management Science*, vol. 38, no. 5, pp. 708–723, 1992.

[16] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[17] M. Rossetti, "Simulation modeling and arena." `https://rossetti.github.io/RossettiArenaBook/`, 2021. 3rd and Open Text Edition. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

[18] J. M. Steele, "Seedlings in the theory of shortest paths," *Disorder in physical systems*, p. 277, 1990.

[19] D. M. Stein, "An asymptotic, probabilistic analysis of a routing problem," *Mathematics of Operations Research*, vol. 3, no. 2, pp. 89–101, 1978.

[20] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research*, vol. 12, no. 4, pp. 568–581, 1964.

[21] A. Baltz, D. Dubhashi, L. Tansini, A. Srivastav, and S. Werth, "Probabilistic analysis for a multiple depot vehicle routing problem," in *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science: 25th International Conference, Hyderabad, India, December 15-18, 2005. Proceedings 25*, pp. 360–371, Springer, 2005.

[22] J. E. Yukich, *Probability theory of classical Euclidean optimization problems*. Springer, 2006.

[23] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2015.

[24] R. Cheng, C. He, Y. Jin, and X. Yao, "Model-based evolutionary algorithms: a short survey," *Complex & Intelligent Systems*, vol. 4, no. 4, pp. 283–292, 2018.

[25] N. Cressie and C. K. Wikle, *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.