Optimizing On-Site Inspection Planning at Eurofins

Author
Vinh Schoorlemmer

First supervisor: Dr. A. Asadi
Second supervisor: Dr.ir. E.A. Lalla
Company supervisor: Bart van de Vrie

Bachelor Industrial Engineering and Management
Department of IEBIS

University Of Twente
Drienerlolaan 5
7522NB Enschede

July 2024

Management summary

This thesis aims to reduce travel times for inspectors at Eurofins Food, Feed and Water Benelux. Bureau de Wit, a company of Eurofins that specializes in hygiene and food safety provides inspections and other services to ensure facilities are compliant with the HACCP regulations. Bureau de Wit has numerous inspectors spread throughout the Netherlands and Belgium to provide inspections at their clients' facilities to ensure hygiene standards and food safety at the facilities. Bureau de Wit is facing pressure to meet growing demand and ensure all the facilities get inspected in time. Hence it is essential, that the inspectors work more efficiently to meet these challenges. We aim to improve the routes of the inspectors by optimizing travel each day and thus allowing more time for inspections over the planning horizon.

An analysis of the current logistics situation was first performed to acquire an understanding of the current logistics structure, constraints, limitations, KPIs and practices. The inspectors visit customers and inspect their facilities, where they also take samples. These samples are transported to the laboratory for further analysis. Since it is infeasible for inspectors to drive to the laboratory each day the inspectors drop the samples off at a predetermined location. In the night the samples get collected by an external party and get transported to the laboratory. The planning of customers and dropping the samples off at a predetermined location are the logistics this thesis focuses on. From the analysis of the current situation, the challenge at Bureau de Wit was identified to be a routing problem.

Bureau de Wit currently has no planning system or tool in place and is entirely reliant on the planning the inspectors make themselves. This offers flexibility to the inspectors but also causes the routes to be suboptimal. Eurofins is unsure whether route optimization would lead to significant cost and time savings. The aim of this research was thus to determine how Eurofins could improve their inspector planning without compromising service quality.

The literature suggests that given the characteristics of the problem, we can formulate it as a vehicle routing problem (VRP). Including the constraints and characteristics of the routing problem, we can define it as a periodic vehicle routing problem with time windows and dropoffs (PVRPTWD). The model that comes closest in the literature to this VRP variation is the PVRPTW. The problem is formulated as a Mixed Integer Linear Program (MILP) and is solved using two solution methods, one exact and one approximate solution method. The solution methods are an exact MILP solver and tabu search (TS) metaheuristic respectively.

The MILP is solved using a general-purpose solver. Before applying the solution approaches to the real-world instances the parameters of the models were tuned. We create artificial data instances based on the Eurofins data for parameter tuning. The TS parameters were optimized to induce desired search behaviour to effectively produce a high-quality solution. The runtime parameters for both the TS and MILP were set to 1800 seconds based on the results of the testing. The TS algorithm consists of an initialization phase, where an initial solution is constructed using a customized greedy insertion heuristic and an optimization phase. In the optimization phase, the tabu search swaps customer schedules to descend to local optima and escape them using a diversification mechanism.

After setting the parameters, the TS and MILP approaches were applied to three real-world instances based on the monthly workload of an inspector. The 3 real-world data instances were related to three different inspectors in different working areas each having unique characteristics. Each instance has a unique node density, average speed, and support facility availability testing the efficacy of the solution approaches in different environments. The results of the TS and MILP are as follows.

The Tabu Search outperforms the MILP in complex data instances, finding solutions that are, on average, 1,71% better than those produced by the MILP. The MILP outperforms the TS in small-scale instances, achieving an optimal solution within ten minutes. The tabu search can improve from an initial solution by around 30% when allocated a computing time of 1800 seconds.

The MILP was unable to achieve the optimal solution for any real-world data instance within the allocated runtime with an average optimality gap of 6,54%. Using TS as the main solution approach the current travel time can be reduced by up to 25%. The MILP achieves lower values when applied to real-world instances.

To conclude, replacing manual planning with an optimized planning approach and incorporating this approach in the operations improves the routing performance and leads to reduced costs and travel times. We thus recommend Eurofins to implement our suggested planning model for their inspectors or a similar approach. A reduction of 25% of travel times is possible when applying the solution approach. Therefore, the action problem is solved using this solution approach. Additionally, we recommend that Eurofins investigates the working areas of the inspectors, so the customers assigned to each inspector are optimal from a logistics perspective. We also recommend that a logistics check is done before signing a contract with a customer since this can lead to operational difficulties. The logistics check is to ensure that adding a customer to the operations is feasible and not at considerably high costs. The scientific contributions of this work lie in the development of a customised construction heuristic and add to the metaheuristic body of knowledge by providing a case study of TS demonstrating its effectiveness in a complex real-world problem.

List of Acronyms

- **BdW**: Bureau de Wit
- **CRISP-DM**: Cross Industry Standard for Data Mining
- **CVRP**: Capacitated Vehicle Routing Problem
- **HACCP**: Hazard Analysis and Critical Control Points
- **KPI**: Key Performance Indicator
- **LZV**: Lab Zeeuws Vlaanderen
- **MDVRP**: Multi Depot Vehicle Routing Problem
- **MILP**: Mixed Integer Linear Programming
- **MPSM**: Managerial Problem Solving Method
- **MTZ**: Miller-Tucker-Zemlin
- **NP-hard**: Nondeterministic Polynomial-time hard
- **PTSP**: Periodic Travelling Salesman Problem
- **PVRP**: Periodic Vehicle Routing Problem
- **PVRPTW**: Periodic Vehicle Routing Problem with Time Windows
- **PVRPTWD**: Periodic Vehicle Routing Problem with Time Windows and Dropoffs
- **TS**: Tabu Search
- **VNS**: Variable Neighborhood Search
- **VRP**: Vehicle Routing Problem
- **VRPTW**: Vehicle Routing Problem with Time Windows

Contents

Chapter 1

Introduction

This chapter provides an overview of the Eurofins situation by discussing the problem this thesis addresses and outlines the problem-solving approach for this problem. First, Eurofins' background and company profile are described followed by Section 1.1, which outlines the problem as perceived by Eurofins and the motivation for the setup of this thesis. The relevant core problem is identified and the action problem to be solved in this project is discussed in Section 1.2. The research questions are covered in Section 1.3 discussing relevant research questions and the problem-solving approach. The scope of this research is discussed in Section 1.4.

1.1 Background

In this section, we discuss the organizational structure of Eurofins, its core business, and current operational challenges. Eurofins Food, Feed & Water Testing Benelux, hereafter referenced as Eurofins is a business line of Eurofins Benelux that provides food and water safety solutions offering services ranging from laboratory testing and inspections to consultancy services. Eurofins mission is: *"To contribute to global health and (food) safety by providing our customers with high-quality laboratory and advisory services"* (Eurofins Food Feed Water Benelux, 2024). A business line is a division of the Eurofins Company, which operates with a high degree of autonomy. A business line consists of business units, which are independent subsidiary companies of the business line. Eurofins has 8 business units that each offer unique services related to ensuring food and water safety. To determine food or water safety laboratory testing of samples and on-site analysis are crucial, which are the core business of Eurofins. Eurofins faces operational challenges in offering these services, which leads to the problem described in the following paragraph.

Eurofins faces challenges regarding the transportation of samples from customers to its laboratories for testing and analysis in its logistics operations. Currently, business units each coordinate their sample transportation process, some business units operate their own in-house logistics units and others outsource this process or opt for a combination. The networks are generally limited in resilience with high operating costs varying in quality of operations between logistics divisions but allow the business units to maintain control over their operations and allow for a customized logistics design to satisfy their needs. Management and operations personnel are convinced that many optimizations and efficiencies are to be exploited by integrating capabilities, processes, and networks. However, Eurofins can be characterized as a divisional organization (Mintzberg, 1980). Eurofins maintains autonomous business units with their own goals and subsequent logistical needs, which makes integration complex. Figure 1 shows an organizational overview of Eurofins' divisions relevant to this thesis. This thesis focuses on one of these business units, Bureau de Wit.

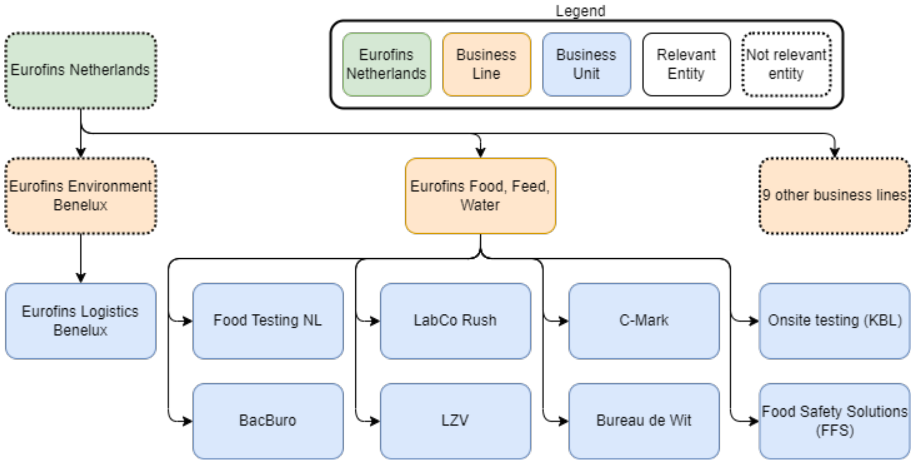

Figure 1.1: *Organizational chart of Eurofins and the relevant entities.*

Bureau de Wit is an expert in food safety and hygiene through performing hygiene inspections and consolatory services. To ensure the hygiene and food safety of their clients samples are taken from the clients tested at the laboratory. This causes challenges for Bureau de Wit to satisfy customer demand and ensure food safety throughout the Netherlands and Belgium. Research is needed to identify opportunities and propose new solutions to increase the logistics efficiency at Bureau de Wit.

1.2 Problem identification

This section identifies the problems faced by BdW regarding its logistics operations. Through the use of a problem cluster one core problem is identified: manual inspection planning.

1.2.1 Action Problem

Bureau de Wit is operating in an industry where finding personnel is becoming increasingly difficult and costs are rising. The need to spend the limited amount of hours an employee has efficiently is thus becoming more urgent.

This eventuality becomes even more clear when the amount of customer orders increases significantly. Serving all customers, while maintaining high-quality service is becoming increasingly challenging. Additionally, with the introduction of increasing and ever-changing hygiene regulations, the work pressure on inspectors increases. This means that the logistics challenges in planning and routing inspectors need to be more efficient.

According to Heerkens and van Winden (2017) an action problem is "The discrepancy between norm and reality, as perceived by the problem owner". We thus formulate the following action problem:

'The routing time of inspectors at Eurofins should be reduced by 10%, without compromising service quality. '

1.2.2 Problem cluster

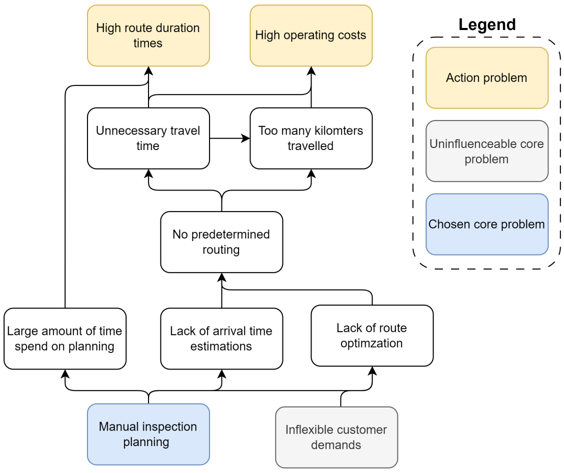

Figure 1.2: *Problem cluster for reducing route duration*

First, to tackle the action problem the root cause of that problem must be tackled. An observation study has been conducted to determine the underlying problems of the action problem, which was executed by interviewing the problem owner in this case Eurofins management and logistics employees. The underlying problems have been organized into a problem cluster to display the causal relationships between different problem statements. Heerkens and van Winden (2017) states that the goal of the problem cluster is to identify potential core problems according to the problem owner. The problem cluster is displayed in Figure 1.2.

Operations personnel and inspectors experience long route duration times or have difficulty finishing their tasks within the designated period. There is a high degree of unnecessary travel time resulting in a large amount of kilometres travelled. Some inspectors experience having no predetermined routing, the cause of having to drive excessive times to visit all customers on a given day. The routes currently get manually planned by the inspectors themselves in combination with customer wishes this creates sub-optimal routes according to management. Additionally, this necessity to plan the routes by the inspectors results in having less time to visit customers. The inspectors also have difficulty estimating when they arrive at customers causing them to not plan, since they can visit a different amount of customers each day. The demand of the customer is also a factor causing non-optimized routes, by having stringent wishes, however, this factor is determined to be non-influenceable by management.

1.2.3 Core problem

There are two core problems identified in the problem cluster. These problems are the furthest from the action problems and are the underlying causes of the other problems in the problem cluster. The first underlying problem is the stringent customer wishes for some customers. These customers have agreements with the company dictating visit times and frequencies. However, these wishes can cause the routing to be inefficient, since inspectors have obligations to visit a certain customer while visiting another would be more optimal. These decisions and agreements are made between sales and the customer and are thus non-influenceable problems and are thus left out of scope.

The other core problem, manual inspection planning is something that can be influenced. Currently, inspectors plan their routes and visit times by themselves. Due to the large amount of clients an inspector needs to visit each month, planning is challenging since there are a lot of combinations of customers to be visited each day all with different opening times, preferences, and agreements. This problem can be defined as a complex vehicle routing problem (VRP). The quality of the routes thus relies almost completely on the inspector's experience and prowess in planning.

The feasibility of planning thus takes priority over making efficient routes for a month. However, Bureau de Wit is unsure about the impact of commercial planning software and the improvement in the inspectors' routes. Given that this is an underlying problem that leads to other previously identified problems the following core problem is defined:

'Manual inspector planning is inefficient in making routes and Eurofins should use quantitatively based inspector planning.'

1.3 Research questions

Considering the problem identification, action, and core problems (Section 1.2) in cooperation with the stakeholders, the main research question is formulated. This question specifically targets the core problem and is focused on finding a solution. The reader is referred to Appendix D to get background information about the methodology and approach, which lead to these research questions. We formulate the following question:

'How can Eurofins improve their inspector planning in order to reduce routing duration times without compromising service quality at Bureau de Wit?'

To answer the main research question, the following research sub-research questions are formulated and describe the steps needed to complete the research. For each sub-research question, the main objectives and purpose are discussed. The sub research questions are:

1. *What is the current transportation network at Bureau de Wit?*

   The objective of this research question is to gain insights into the current business operations in regard to the sample transportation network. An accurate analysis of this current network helps with identifying possible improvements and evaluating solution fit, which will be discussed in the later steps. The route planning methodology, KPIs, limitations and network design have to be identified.

2. *Which methods are applicable for solving the Eurofins vehicle routing problem according to scientific literature?*

   This question focuses on finding knowledge in regards to optimization techniques related to the Eurofins situation. Scientific literature is looked into to discover relevant VRP variants that fit the current network considering the practical constraints and are compatible with the available data and choosing the most suitable one. The literature review provides an overview of techniques and relevant models to formulate a suitable solution approach.

3. *Which model and technique should be used to optimize the inspection planning at Eurofins?*

   This question aims to find a solution approach to optimize inspector planning for the chosen VRP variant. The most complex techniques are likely not feasible due to the time span of this thesis. Assumptions, requirements and model inputs should be considered in answering this research question

4. *How does the chosen solution approach perform compared against to the current situation?*

   This question covers the evaluation of the model and the impact of the proposed solution. The evaluation is done by analyzing KPIs relevant to the inspector's planning. To assess the performance of the proposed solution the change in KPIs is analysed. The other aspect is the verification of the model outcomes and model validation.

5. *What conclusions and recommendations can be made from executing the thesis at Eurofins?*

   The final step covers the main outcomes of this research and provides recommendations for Eurofins for future actions and points of interest. Avenues for future research and this research's practical limitations are identified. Finally, the practical and theoretical contributions of this work are discussed.

1.4 Scope

The logistics operations of Eurofins and its business units are very extensive as mentioned in Sections 1.1 and 1.3, so covering all aspects of this operation is infeasible. In coordination with the problem owner, a research scope has been defined taking half a semester of research time into account. As expanded upon in Section 1.1 Bureau de Wit is the business unit focused on, the other business units are thus out of scope.

This research will only take the Netherlands as the geographical scope of this research since the relevant logistics operations of Bureau de Wit are not extensive enough to warrant exploration in Belgium and Luxembourg. Furthermore, this research shall constrain itself to the planning of individual inspectors and not take the planning as a whole. Secondly, we limit the scope of this research to the monthly planning of an inspector even though many other different planning policies are possible. Most customers have a specified visit frequency (yearly, bi-yearly, quarterly, monthly), and inspectors of Bureau de Wit work based on monthly planning, it is the wish of the company to keep this planning interval. The inspector routing and planning are considered to be in the scope of this research when it comes to the logistics of Bureau de Wit. In Chapter 2 other logistics operations in Bureau de Wit are elaborated upon to get a clear picture. However, sample transportation that is not done by the inspectors is left out of scope of this research.

The reader is referred to Section 4.1 for assumptions made to formulate the mathematical model. An assumption that is made about customers is that all customer orders are known before planning and any uncertainty in customer orders is left out of scope. Due to insufficient data and the functioning of the mathematical model, this assumption was made. Additional assumptions made throughout this work are elaborated upon in their respective section.

Chapter 2

Context analysis

This chapter provides a context analysis of Eurofins's current situation in particular Bureau de Wit. As outlined in Section 1.3 the following research question is answered in this chapter *What is the current transportation network at Bureau de Wit?*. This chapter begins by summarising a preliminary observation study, where all the business units of Eurofins were analyzed based on their logistics operations, which is followed by a discussion on why Bureau de Wit is chosen as the focus of this thesis in Section 2.1. Section 2.2 explores the business unit logistics operations in particular the planning aspects in detail, as well as the characteristics of this logistics problem. This is followed by discussing the relevant KPIs to assess the performance of the solution and current logical performance in Section 2.3. Finally, we discuss the problem requirements and limitations in Section 2.4 followed by a subconclusion in Section 2.5.

2.1 Organization

In this section, the logistics of the most important business units within Eurofins are discussed. This Section is based on the preliminary research done at Eurofins and provides background information on the other business units. Due to the complex organizational structure of Eurofins, the sample transportation design varies vastly. The business units are clustered based on their logistical operations.

2.1.1 Heerenveen

Heerenveen consists of 2 business units operating out of the same laboratory, hereafter simply referred to as Heerenveen. The sample transportation activities for Heerenveen mainly consist of using the *linehaul* and Eurofins logistics. The line haul is a network that connects Eurofins labs throughout northwestern Europe. The line haul allows Heerenveen to transport samples to other Eurofins labs on the line haul and receive samples from those other labs for testing within a day. The line haul's operations are outsourced.

The other large sample transportation activity that is relevant is the use of Eurofins Logistics. Eurofins Logistics is a business unit that is not part of the same Eurofins referred to in this thesis, but another unit in the overall Eurofins organization. Eurofins logistics provides transportation, warehousing, and customer service to other Eurofins entities. Heerenveen uses Eurofins logistics to transport samples from customers to their lab.

Heerenveen also operates a dedicated line in the northern Netherlands visiting their most important customers. This line is outsourced and Heerenveen has significantly decreased these dedicated lines over the past three years. Additionally, Heerenveen is one of the two biggest labs in Eurofins and almost all samples from Bureau de Wit and C-mark are transported to and tested at this site.

2.1.2 Lab Zeeuws Vlaanderen

Similarly to Heerenveen, LZV also uses the line haul to transport and receive samples for testing from other Eurofins labs. LZV also makes use of Eurofins logistics, but only for transportation orders, it cannot fulfill itself.

LZV also operates its own vehicle fleet to pick up and take samples at customers throughout the day. The routes get planned ad-hoc based on the changing circumstances because throughout the day LZV receives requests for sample pickup and taking while the vehicles are already executing their routes. The samples get collected and taken throughout the day and are all received at the lab at the end of the day.

2.1.3 BdW and C-mark

Bureau de Wit and C-mark have almost identical sample transportation networks. First of all, both entities have inspectors that visit clients but for different types of services. During these inspections, multiple samples get taken for testing in the laboratory in Heerenveen. Throughout the day an inspector visits multiple clients taking samples at almost every visit, the samples then get dropped off at one of the many support facilities at the end of the day. The support facilities are geographically spread out throughout the Netherlands. The samples at the support facility get collected by Eurofins logistics at night and dropped off at Heerenveen before the lab's opening. See Figure 2.1 for a schematic representation of this process.

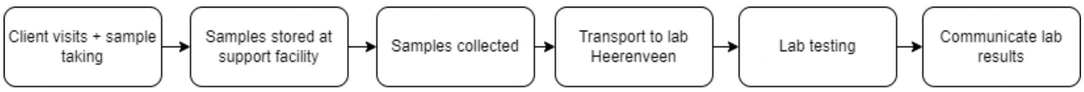

Figure 2.1: *BdW's and C-mark's sample transportation process.*

2.1.4 Other business units

The other business units have fewer sample transportation operations. Most business units opt to outsource their sample transportation activities to Eurofins logistics, Rush logistics or other third parties.

2.1.5 Bureau de Wit choice

Due to the vastly different organization of logistics in each business unit combining logistics operations is considered to be not feasible. This is the reason that together with the stakeholders in this project it has been decided to focus on one business unit and look to improve the logistics operations at that business unit. Bureau de Wit has been chosen as the focus of this thesis for multiple reasons. Bureau de Wit currently lets their inspectors plan their routes themselves, while other business units use dedicated planning software to create routes and coordinate logistics. The management believes the operational efficiency can be increased by improving the planning the inspectors currently make. Additionally, since the scope of this project would get too large if multiple business units were involved it has been chosen to focus on one, which is Bureau de Wit. Additionally, the operations of Bureau de Wit are of such scale that small improvements in the schedule can lead to large overall gains in terms of cost and customer satisfaction.

The planning of inspectors between Bureau de Wit and C-mark varies vastly. C-mark plans its inspectors based on monthly planning and gets controlled by a team of planners, who alter the routes and planning based on the circumstances. Bureau de Wit does not plan their inspectors and provides them with a list of customers to visit within the calendar month and lets the inspectors plan routes. The inspectors report to the operations of Bureau de Wit, however, at Bureau de Wit the operations team interferes almost never with the planning the inspectors make. This aforementioned situation, where inspectors create their routes is the core topic this thesis tackles as mentioned in Section 1.2.

2.2 Bureau de Wit's logistics problem

As mentioned in Chapter 1 and Section 2.1.5, we will focus on the planning of Bureau de Wit. Together with the company and stakeholders considering the limited time frame we have decided to focus on this specific business unit of Eurofins. The section begins by outlining the transportation network of Bureau de Wit focusing on the planning process and discussing the sample transportation network. The current network characteristics are discussed in Section 2.2.1 and a small illustration is provided for clarity. This is followed by outlining the characteristics of the network and discussing characteristics that can be changed within the network in Section 2.2.2. The current performance of Bureau de Wit's network and the relevant KPIs are in Section 2.2.3. These KPIs also serve as the key figures to assess the proposed solution in the later Chapters 5 and 6. Finally, the requirements and limitations for the solution methodology are presented in Section 2.2.4.

2.2.1 Sample transportation network

Bureau de Wit has a combination of a planning and routing problem, which involves visiting customers, inspecting their facilities, and taking samples to be analyzed in the laboratory. This section provides an overview of this planning and routing process and the underlying principles driving these processes. As noted earlier we focus on the Bureau de Wit's planning and routing process due to the scope, but the process described below is similar to that of C-Mark.

Planning process

Bureau de Wit currently opts for a decentralized planning process, where the inspectors create and drive their own routes. There is currently no central route planning software in place to create routes for inspectors or tools to aid in this process. Letting inspectors plan their own routes gives them a high level of autonomy and flexibility, which has its advantages and disadvantages. The advantages are that inspectors know their operating area and customers well, which is good for customer service since inspectors can customize their service based on the customers' preferences. The disadvantage however is that letting inspectors plan themselves can create sub optimal routes when approached from a logistical perspective.

In the current planning process the operations department of Bureau de Wit currently provides a monthly list of customers and tasks to each inspector needs to finish within a calendar month and gives full autonomy to the inspector to plan their own routes and schedules. The list of customers an inspector receives is based on the geographical location of the inspector's home, since this is also their starting location. The amount of visits on the customer list is around the same volume for each inspector. This planning process is provided in Figure 2.2. The process begins the week before the beginning of each month, where all customer visits for the next month get aggregated. Then the logistics department assigns these customers to an inspector based on geographic area and workload. The inspectors then receive their customer list for the month and plan the customers in their schedules. The inspectors have autonomy in their working hours as long as the customers on their list get serviced within the month and comply with any additional agreements there may be between Bureau de Wit and the customer. There are large differences in the manner an inspector plans their customers, some prefer planning on a weekly basis, some plan for the whole month, and some plan just a few days before they visit. Using an application on their smartphones inspectors report their progress back to the operations department and notify in case of unforeseen circumstances.

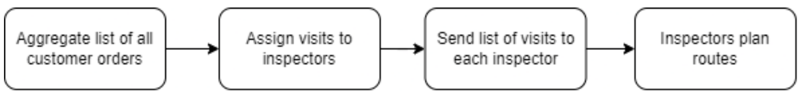

Figure 2.2: *BdW's current planning process.*

Transportation network

As mentioned in Section 2.2.1 the inspectors inspect the customers' facilities and take samples throughout the day used for testing. The samples thus need to be transported from the inspector's vehicle to the laboratory. Eurofins has a support facility network, which business units can use for their logistical operations. This support facility network is used by Bureau de Wit and its inspectors, by using them as drop-off locations for the samples collected throughout the day. In the current system inspectors visit the customers and take samples and these samples get dropped off at the end of the day before a specified time at one of these *dropoff* locations. After the dropoff deadline, Eurofins Logistics collects these samples and delivers the samples to the Heerenveen lab before the start of the morning shift using their network of hubs, vehicles, drivers, and planning software. Figure 2.3 provides a high-level schematic overview of the current network for one day, where multiple inspectors and dropoff locations are located from Bureau de Wit's point of view, and Figure 2.4 provides an example of the problem that one inspector faces over an arbitrary time period from the inspector's point of view. The latter as defined in Section 1.3 is also the scope of this research, we thus solely focus on the planning from the inspectors' point-of-view. Consequently, the depot locations and the night sample transportation logistics by Eurofins Logistics were also left out of scope.

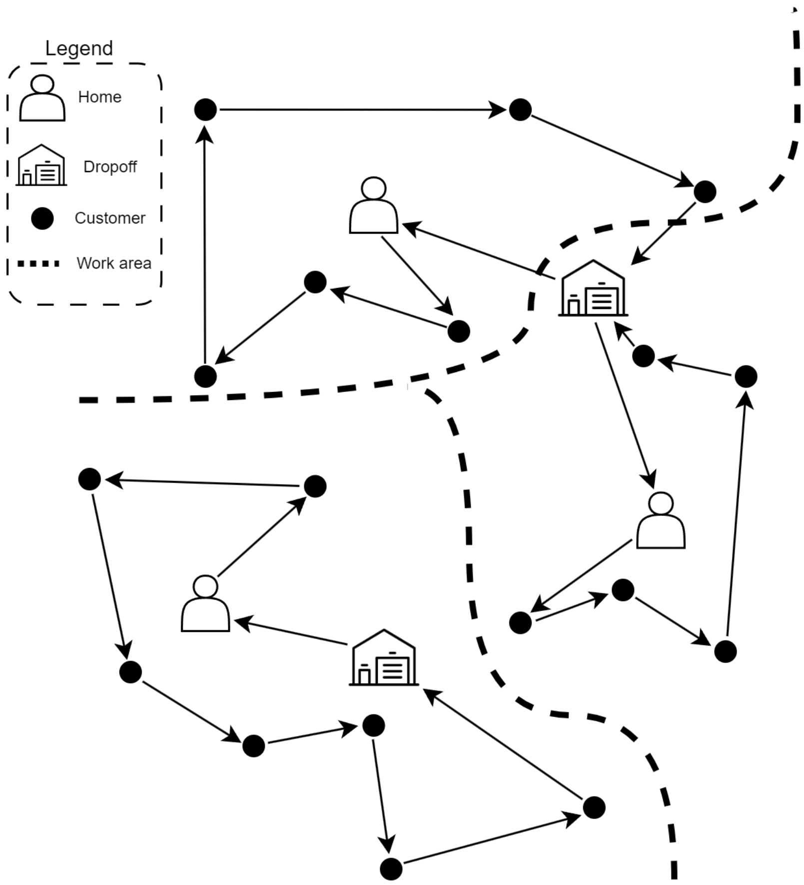

Figure 2.3: *Schematic overview of the planning for multiple inspectors on one day.*

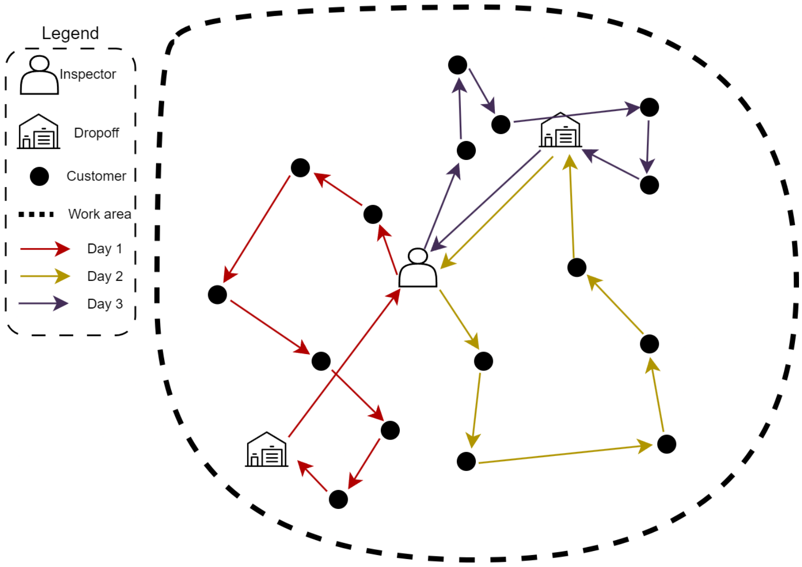

Figure 2.4: *Schematic overview of the planning for one inspector over 3 days.*

2.2.2 Network characteristics

To provide a more comprehensive understanding of Bureau de Wit's network characteristics, the key aspects that define the inspector planning of Bureau de Wit are discussed. The aspects of customer service times, customer visit times, customer visit frequency, and vehicles are discussed. These aspects are explained in the following sections and play a key role in the model formulation in later chapters.

Service times and demands

The Bureau de Wit clients request a wide array of services ranging from legionella inspections to 3-hour-long factory inspections. If a client is known and an inspector comes for an inspection the service time is uncertain, since there are varying factors that can shorten or prolong inspection time. However, Bureau de Wit has an in-house estimation tool to estimate the duration of a visit based on visit type and customer. For this thesis, this estimation tool is assumed to be accurate enough for the mathematical model discussed in Chapter 4 and according to inspectors and operations personnel, the estimations are accurate if aggregated daily. The re-inspections and new clients cause the number of visits in a month and the total orders within a month to be slightly stochastic. Re-inspections and new customers make up around X% (hidden for confidentiality) of total visits in a month, so requests are assumed to be deterministic over the span of a month. There is also the aspect of pickup demand or in this case number of samples that are taken that need to be transported to the lab. This aspect will be addressed in a later section.

Customer visit times

When a client is to be inspected they usually do not specify a specific date or time the inspector needs to do their visit. This would defeat the purpose of hygiene inspections since facilities are always supposed to be hygienic when food is prepared, so the time an inspector visits should not matter when it comes to hygiene standards. However, inspectors are subject to time windows in which they visit and inspect for certain clients. The inspectors are constrained to the opening times of their clients, which we define as a time interval in which an inspection takes place. Most clients have a wide period in which an inspection can take place, usually 8:00-17:00 with some having even larger time intervals. Some clients have shorter time windows on specified days, especially restaurants, which are closed in the mornings for example. The inspectors always take opening times into account when creating their routes.

Inspections take place throughout the week, excluding Saturdays and Sundays. The other type of client availability is opening days where certain clients are closed on certain days of the week or specific days, which are usually holiday-related. This is also an aspect that is taken into account when an inspector makes their planning.

Visit frequency

As mentioned earlier clients get planned based on a monthly basis, however, clients can of course be visited multiple times in a month due to agreements or re-inspections if a client fails an inspection. So there is a visit frequency for every client, with most clients just having one inspection planned in a month. If a client must be visited multiple times in a month, the inspections take place in different weeks or an interval of days between successive visits is assigned.

Vehicles

Bureau de Wit leases a fleet of vehicles for every inspector. Every inspector owns a vehicle, which they use to drive to customers, store inspection equipment, and store samples taken on-site at the customers. The inspector always drives this vehicle when doing inspections and the inspector can keep this vehicle at home. So routes and vehicles start at the inspector's home and they end at the inspector's home. The samples taken during the day get stored in a cooling unit in the vehicle. The physical size of the samples is so small that there is practically always enough room for the samples to be stored in the cooling unit. Since the samples always get dropped off at the support facilities at the end of the day, there are never storage capacity limitations when executing a route for a day, so pickup demand is assumed to be negligible.

2.3 Key performance indicators

The main Key Performance Indicator (KPI) for the Bureau de Wit's logistics network is the amount of time associated with servicing all clients. Cost is a main driver when looking at the performance of a logistics network, however in this case most costs are fixed or are derivatives of execution time. The main costs include personnel, vehicle, fuel, consumables, and overhead costs. Personnel costs and vehicle costs are fixed. Fuel costs are based on the distance traveled, which is highly correlated to the amount of driving hours. Consumables, items used for inspecting a client a variable cost, and most other costs are also variable. So route execution time is the main KPI to assess the inspector's planning performance since lowering this time allows to visit more customers in the same amount of time decreasing the costs associated with visiting a customer since these fixed costs get spread out over a larger amount of visits. The total time to service all customers is thus the KPI that should be decreased as much as possible and is also the most important indicator to resolve the action problem described in Section 1.2. Optimizing the route execution time leads to lowered costs and hiring pressure, which can contribute to the company's profitability and consequently competitive positioning.

The other critical KPI is the average driving time in hours per day. This indicator of time indicates how long an inspector is driving to clients per day. Driving hours are non-productive because it does not contribute to the value-adding service of inspecting, so allowing inspectors more time to inspect can increase the quality of inspectors and it allows for more customers to be visited in a day if the average driving time is lowered. Secondly, this KPI is relevant since it has an impact on the carbon emissions and travel distance of the vehicles. Since, if the vehicle is driving less time with the same average speed, the total distance travelled on a given day is lower reducing fuel consumption and consequently a reduction in fuel cost and carbon emissions.

2.4 Requirements and limitations

To formulate a solution to the Eurofins routing problem several requirements and limitations should be considered. The total amount of working hours an inspector has per week is at maximum, 40 hours per week. There is, however, flexibility in the amount of working hours per day and the time of those working hours for an inspector. Furthermore, a limitation concerning the working days for inspectors and clients is that no visits can be scheduled on Saturdays and Sundays and some holidays. The time windows specified by the clients need to be adhered to even if it results in less efficient routes. There are no vehicle capacity limitations, as well as the working hours of an inspector except the total working hours specified earlier.

The route planning should be made for each day in the calendar month for an inspector. A limitation is that now all time windows are exactly known since this is information that is communicated between inspector and client, but not communicated to the operations department. Another limitation is that not all visits are known before the beginning of a planning period, however, this is assumed to be negligible. This thesis focuses only on the operations of Bureau de Wit's inspection planning. Requests from other business units are also left out of scope.

2.5 Conclusion

This chapter explored the current logistics at Eurofins by answering the following research question *What is the current sample transportation network at the business unit level?*. First, all major logistics operations of the key business units were discussed. Every business unit has its own mix of private and outsourced methods to satisfy its logistical needs. Heerenveen uses Eurofins Logistics, an internal logistics service unit to transport their packages from their clients to their labs, a dedicated line collecting samples from important customers in one region of the Netherlands and the line-haul and transportation network connecting Eurofins labs all over Europe for inter-laboratory transportation. Lab Zeeuws Vlaanderen also uses the linehaul and some private couriers. Lab Zeeuw Vlaanderen also operates a private fleet of vehicles to collect samples at short notice from customers throughout the day. Bureau de Wit and C-mark, almost exclusively use their private fleet of vehicles to do onsite inspections at the customers where they take samples to be tested in the lab in Heerenveen. The samples are deposited at a support facility and transported by Eurofins Logistics to the Heerenveen lab at night. Finally, the other smaller entities mainly use private carriers to transport their samples from and to other customers and labs.

After deliberation, it has been decided to focus on the logistics of Bureau de Wit, in particular the monthly planning of inspectors. The Bureau de Wit sample transportation logistics are based on a network of support facilities, where samples collected throughout the day are deposited at a support facility and picked up by Eurofins Logistics which transports the samples to the laboratory. Currently, the inspectors are given a monthly list of customers that need to be visited within the month. The inspectors plan these customer visits themselves with varying degrees of efficiency. When a customer is able to be inspected is influenced by various factors including, the geographic position of the clients, time windows, open and closing days, agreements, and the drop off deadline. The inspectors need to adhere to these time windows and potential customer preferences to keep customer satisfaction and do their inspections when the facilities are opened. Additionally, the service time is a critical component in the planning of routes. The Key Performance Indicators (KPIs) for Bureau de Wit's scheduling logistics include the route execution time and average driving time per day. These 2 KPIs are crucial for understanding the schedule efficiency and optimizing the planning, ensuring visits get efficiently planned throughout the planning horizon of a month. The requirements and constraints to consider are; service times, time windows, planning horizon, clients available days, support facility locations, and inspector working hours. The limitations are that visits are known after the planning need to be inserted mid-way through the execution of the planning are not disregarded. Lastly, the scope is defined to be the scheduling of individual inspectors, the entire scheduling problem for all inspectors simultaneously is considered to be out of scope. To remain consistent throughout this thesis, the aforementioned problem will be referred to as Eurofins's transportation problem, since it is about the sample transportation for one of Eurofins's business units.

Chapter 3

Literature review

Eurofins's situation can best be described as a routing problem for which its current inspection planning requires optimization by minimizing overall route execution time. We previously referred to this problem as a scheduling and planning problem, however in the following sections it is made clear that it is also a routing problem. This Chapter will thus discuss relevant literature in regards to model approach, and solution methods for routing problems considering the characteristics of Eurofins's transportation process. So the following research question is addressed *Which model and technique should be used to optimize the inspection planning at Eurofins?* First, we introduce the vehicle routing problem in Section 3.1 and discuss various characteristics a vehicle routing problem can contain. This is followed by presenting the mathematical formulation of VRP variants and a discussion of which variants align best with the characteristics of the Eurofins's situation in Section 3.2. Heuristic and metaheuristic solution approaches that fit the mentioned VRP variants are discussed in Section 3.3. Finally, we conclude by discussing the most appropriate VRP variant and its solution approach in Section 3.4.

3.1 Vehicle routing problem

The vehicle routing problem (VRP) is a logistical problem, where a set of routes are determined to serve a set of customers by a set of vehicles. The goal of the VRP is to minimize or maximize a set objective, usually cost or distance traveled. The first VRP was introduced by Dantzig and Ramser (1959), which was called "The Truck Dispatching Problem". Danzig and Clarke aimed to find optimal routing for homogeneous gasoline trucks between a gas terminal and service stations. Five years later, the problem was generalized by Clarke and Wright (1964) as a linear optimization problem on how to serve a set of geographically dispersed customers from a central depot using a fleet of trucks with varying capacities, which is now known as the "Vehicle Routing Problem" (VRP). The VRP is known as an NP-hard problem introduced by Lenstra and Kan (1981), meaning that algorithms that can exactly solve the problem are only efficient in small-scale instances. The Clarke and Wright savings algorithm is one such algorithm, which is a step-by-step decision process for achieving local optimization (Clarke and Wright, 1964), however, this algorithm does not guarantee a globally optimal solution.

3.1.1 Taxonomy of VRPs

VRPs have increased in complexity to deal with real-life situations and do not satisfy this original formulation and definition byClarke and Wright (1964)(Braekers et al., 2016). This is also the case for the situation at Eurofins, where there an inspector needs to visit multiple locations across a month, whilst adhering to various constraints. Ni and Tang (2023) adapted the three-level taxonomy proposed in Braekers et al. (2016) and Eksioglu et al. (2009). Figure 3.1 displays the VRP taxonomy of Ni and Tang (2023) and the characteristics matching with the Eurofins situation eluded to in Chapter 2. Most modern VRPs combine several of these characteristics displayed in Figure 3.1.

- **1 Type of Study**
  - 1.1 Theory
  - 1.2 Applied methods
    - 1.2.1 Exact methods
    - 1.2.2 Classical Heuristics
    - 1.2.3 Metaheuristics
    - 1.2.4 Hyper-heuristics
    - 1.2.5 Machine learnig
    - 1.2.6 Simulation
    - 1.2.7 Real time solution methods
  - 1.3 Implementation documented
  - 1.4 Survey, review or meta-research
- **2 Scenario Characteristics**
  - 2.1 Number of stops on route
    - 2.1.1 Known (deterministic)
    - 2.1.2 Partially known,partially probabilistic
  - 2.2 Load splitting constraint
    - 2.2.1 Splitting allowed
    - 2.2.2 Splitting not allowed
  - 2.3 Customer service demand quantity
    - 2.3.1 Deterministic
    - 2.3.2 Stochastic
    - 2.3.3 Unknown1
  - 2.4 Request times of new customers
    - 2.4.1 Deterministic
    - 2.4.2 Stochastic
    - 2.4.3 Unknown1
  - 2.5 On site service/waiting times
    - 2.5.1 Deterministic
    - 2.5.2 Time dependent
    - 2.5.3 Vehicle type dependent
    - 2.5.4 Stochastic
    - 2.5.5 Unknown
  - 2.6 Time window structure
    - 2.6.1 Soft time windows
    - 2.6.2 Strict time windows
    - 2.6.3 Mix of both
  - 2.7 Time horizon
    - 2.7.1 Single period
    - 2.7.2 Multi period
  - 2.8 Backhauls
    - 2.8.1 Nodes request simultaneous pickups and deliveries
    - 2.8.2 Nodes request either linehaul or backhaul service, but not both
  - 2.9 Node/Arc covering constraints
    - 2.9.1 Precedence and coupling constraints
    - 2.9.2 Subset covering constraints
    - 2.9.3 Recourse allowed
- **3 Problem Physical Characteristics**
  - 3.1 Transportation network design
    - 3.3.1 Directed network
    - 3.3.2 Undirected network
  - 3.2 Location of addresses (customers)
    - 3.2.1 Customers on nodes
    - 3.2.2 Arc routing instances
  - 3.4 Number of points of origin
    - 3.4.1 Single origin
    - 3.4.2 Multiple origins
  - 3.5 Number of points of loading/unloading facilities (
    - 3.5.1 Single depot
    - 3.5.2 Multiple depots
  - 3.6 Time window type
    - 3.6.1 Restriction on customers
    - 3.6.2 Restriction on roads
    - 3.6.3 Restriction on depot/hubs
    - 3.6.4 Restriction on drivers/vehicle
  - 3.7 Number of vehicles
    - 3.7.1 Single vehicle
    - 3.7.2 Limited number of vehicles
    - 3.7.3 Unlimited number of vehicles
  - 3.8 Capacity consideration
    - 3.8.1 Capacitated vehicles/limited capacity
    - 3.8.2 Uncapacitated vehicles/unlimited capacity
    - 3.8.2 Uncapacitated vehicles/unlimited capacity
  - 3.9 Vehicle homogeneity (Capacity)
    - 3.9.1 Similar vehicles
    - 3.9.2 Load-specific vehicles
    - 3.9.3 Heterogeneous vehicles
    - 3.9.4 Customer-specific vehicles
  - 3.10 Travel time
    - 3.10.1 Deterministic
    - 3.10.2 Function dependent
    - 3.10.3 Stochastic
    - 3.10.4 Unknown
  - 3.11 Object
    - 3.11.1 Single
    - 3.11.2 combinations
- **4 Information Characteristics**
  - 4.1 Evolution of information
    - 4.1.1 Static
    - 4.1.2 Partially dynamic
  - 4.2 Quality of information
    - 4.2.1 Known (Deterministic)
    - 4.2.2 Stochastic
    - 4.2.3 Forecasted
    - 4.2.4 Unknown (Real-time)
  - 4.3 Availability of information
    - 4.3.1 Local
    - 4.3.2 Global
  - 4.4 Processing of information
    - 4.4.1 Centralized
    - 4.4.2 Decentralized
- **5 Data Characteristics**
  - 5.1 Data Used
    - 5.1.1 Real world data
    - 5.1.2 Synthetic data
    - 5.1.3 Both real and synthetic
  - 5.2 No data used

Figure 3.1: *Taxonomy for VRP literature (Ni and Tang, 2023) Characteristics aligning with the Eurofins situation are highlighted in red.*

The most studied VRPs are deterministic VRPs, for which customer requests are known and information is known before the execution of routes (Braekers et al., 2016). This concept extends to the number of stops, demand, and other information on a given route. One of the most common extensions of the VRP is the VRP with time windows (VRPTW). The VRPTW extends to VRP to include the constraint that customers can only be served in a specified time interval. Sometimes soft time windows are used, which are usually used to model customer satisfaction. In the VRP with soft time windows, the VRP is relaxed to a VRP with penalty costs (Ghannadpour et al., 2014). Furthermore, the capacitated vehicle routing problem (CVRP) is also known as the classical VRP and is used extensively. The CVRP operates under the additional constraint that customers have deterministic demand, which has to be satisfied with vehicles that have a limited capacity (Braekers et al., 2016). The multi-depot vehicle routing problem (MDVRP) is a more complex routing problem compared to the general VRP, as the MDVRP considers that the fleet of vehicles can depart from multiple depots, while most general VRPs only consider one depot (Montoya-Torres et al., 2015). So, when multiple depots are considered for serving a set of customers with a set of vehicles the problem becomes a MDVRP. Another characteristic and variant of the vehicle routing problem is the periodic vehicle routing problem (PVRP). This variant is applicable when the planning is made over a certain period (a month for example) and deliveries or services can be provided to the customer in different days. The PVRP allows customers to be visited once or multiple times (Campbell and Wilson, 2014).

The different variants of the VRP and relevant characteristics can be combined into one mathematical model with an accompanying solution methodology. Vieira et al. (2023) combines the characteristics mentioned above and incorporates the additional characteristics of site dependency, heterogeneous fleet, multiple drips, and fleet sizing into the mathematical formulation. A characteristic not mentioned in the Ni and Tang (2023) taxonomy is the characteristic of having multiple starting locations and a set of end locations. The multi-depot vehicle routing problem MDOVRP comes close to this characteristic, however in the MDOVRP vehicles are allowed to end on any customer node (Lalla-Ruiz et al., 2016), whilst in the Eurofins problem the inspectors need to return home, but need to visit a deposit location first. In this thesis, we limit the scope to encompass one inspector, however, in the full scale of the problem, multiple depots should be considered. The following subsection, the mathematical formulation of the VRP is provided. Later, the aforementioned variants of the VRP: VRPTW, and PVRP are discussed more extensively and the mathematical models are provided.

3.1.2 Combining VRP characteristics

The aforementioned VRP models have a wide range of characteristics. As mentioned earlier, in real-world scenarios a wide variety of characteristics need to be incorporated to get an accurate model. Additionally, the complex VRPs also need to be solved to have practical value. So, we discuss the combination of several VRP characteristics and its solution methods in this section. The periodic vehicle routing problem (PVRP) is one such model, where periodicity is incorporated in the VRP. Periodicity is a generalization of the VRP, where instead of planning for one period, multiple periods need to be planned. Cordeau et al. (1997) formulated such a PVRP and solved it using a tabu search heuristic. In this case the authors solve for the basic VRP objective namely prioritizing the minimization of vehicles followed by minimizing travel costs. Interestingly, the solution method for the PVRP also works for the MDVRP, since an MDPVRP can be formulated as a PVRP with some minor adaptations. Cordeau et al. (1997) also demonstrate that the tabu search for the PVRP also works for the PTSP, VRP and MDVRP. Cordeau et al. (2001) extended this tabu search heuristic to also include time windows within the solution approach. Consequently, similar to the relationship between de PVRP and MDVRP the same relationship is present in the PVRPTW and MDVRPTW. Yu and Yang (2011) demonstrate one of the more modern approaches to the PVRPTW, where they solve the PVRPTW using ant colony optimization (ACO). The model differs from the Cordeau et al. (2001) model, where they optimize for total transportation distance as a secondary objective instead of the total travel time. Additionally, the Yu and Yang (2011) do not include a maximum tour duration (MTD) constraint, while the tabu search based articles do. In the PVRP and PVRPTW's discussed they make use of regular visit schedules, schedules where the interval between 2 visits is equal. However, due to the practical inconvenience where a restaurant does not want to supply on consecutive days a different formulation of visit schedules is necessary. Rothenbächer (2019) addresses this issue with flexible visit schedules, where visit schedules reflect theses constraints. The formulated PVRPTW with flexible visit schedules is consequently solved using an extension of the branch-and-cut algorithm, which is an exact method.

Instead of combining VRP characteristics, combining solution approaches is also possible which Hesam Sadati et al. (2021) demonstrated. Hesam Sadati et al. (2021) solves an MDVRP and extensions of the MDVRP, the MDOVRP and MDVRPTW using a variable neighbourhood search (VNS) with tabu shaking (TS) algorithm. The tabu shaking algorithm is based on the tabu search (TS). A dropoff characteristic or characteristic, which serves as a support facility but does not contribute to the value creation has not been discussed yet. Liu (2019) uses Mixed Integer Programming (MIP) to solve a complex vehicle routing problem for meal delivery using drones. Throughout the day the battery of the drones get depleted and need to be recharged at specified recharge stations. This characteristic even though less prevalent in existing PVRP literature is a constraint used in practical problems involving drone deliveries. Another interesting note is that Liu (2019) uses a multi-tier, multi-objective objective function, where multiple business objectives are defined and quantified and sorted based on importance in a certain tier.

Article	VRP characteristics					Objective	Solution Methodology
	TW	MD	P	MTD	DL		
Cordeau et al. (1999)		✓	✓	✓		Minimize number of vehicles first and minimize total travel time	Tabu search (TS)
Cordeau et al. (2001)	✓	✓	✓	✓		Minimize number of vehicles first and minimize total travel time	Tabu search (TS)
Yu and Yang (2011)	✓		✓			Minimize number of tours first and minimize total transportation distance	Ant colony optimization (ACO)
Rothenbächer (2019)	✓		✓	✓		Minimize total costs	Branch-and-Price-and-Cut (exact)
Hesam et al. (2021)	✓	✓		✓		Based on data set used (Minimize objective)	Variable Neighbourhood Search (VNS) with Tabu Shaking (TS)
Liu (2019)	✓	✓			✓	Multi-tier, multi-objective function. The tiers are: safety, lateness, efficiency.	Mixed Integer Program (MIP)

Table 3.1: *Literature review of VRP variants and solutions.*

Table 3.1 provides an overview of the six articles referred to in this section. The objectives of the routing problems in the articles are summarised and the solution approaches are presented for each article. We discuss solution approaches later in this literature review in Section 3.3. The four characteristics of the Eurofins routing problem; time windows, periodicity, maximum tour duration and dropoffs are not all covered in one of the articles reviewed. Cordeau et al. (2001) comes the closest to the Eurofins situation with 3 out of 4 characteristics aligning.

3.2 Mathematical formulation of VRPs

In this section, we discuss the different mathematical formulations of the three VRP variants mentioned in Section 3.1. In Section 3.1.1 we provide the formulation of the classical VRP followed by the VRP with time windows in Section 3.1.2. Finally, in Section 3.1.3, we discuss the periodic VRP.

3.2.1 classic VRP

The classical Vehicle Routing Problem (VRP) is presented using an adapted formulation by Munari et al. (2016). In the Munari formulation, a two-index formulation is used, however, a multi-index formulation is required when constraints become too complex. The problem considers the classical case of having a set of customers with known demand to be served by a fleet of vehicles from a central depot. The model aims to minimize the cost objective function by constructing a set of optimal routes. The formulation is as follows:

Sets and indices

1. Set of customers: $C = \{1, ...n\}$
2. Set of all nodes: $N = C \cup \{0, n+1\}$, with 0 and $n+1$ representing the depot nodes. (start at node 0 and return to node n+1)

Paramaters

1. Q: Maximum capacity of each vehicle
2. K: Fleet size of K vehicles
3. q_i: Non negative demand associated with customer i
4. c_{ij}: Cost of travelling from node i to node j

Decision Variables

A binary decision variable x_{ij}, where

$$x_{ij} = \begin{cases} 1, & \text{if there is a route that travels directly from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_j = \text{cumulated demand on the route before visiting node } j$$

Objective Function

$$min \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \tag{3.1}$$

Constraints

$$\sum_{j \in \{0,C\}} x_{ij} = 1 \qquad \forall i \in C, i \neq j \tag{3.2}$$

$$\sum_{i \in \{0,C\}} x_{ih} - \sum_{j \in \{C,n+1\}} x_{hj} = 0 \qquad \forall h \in C, h \neq i, h \neq j \tag{3.3}$$

$$\sum_{j \in C} x_{0j} \leq K \tag{3.4}$$

$$y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}) \qquad \forall i, j \in N \tag{3.5}$$

$$q_i \leq y_i \leq Q \qquad \forall i \in N \tag{3.6}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j, k \tag{3.7}$$

In this formulation, the objective function is defined by (3.1) and imposes that the total travel costs of the routes should be minimized. Constraint (3.2) ensures that all customers are visited once. Constraint (3.3) guarantees the correct flow of vehicles, by simply stating that if a vehicle arrives at a node h it must also depart from node h. The number of routes is limited to K stated in Constraint (3.4). The capacity of the vehicles cannot exceed Q through the expressions (3.5) and (3.6). Constraint (3.5) also ensures subtour elimination. Later in this chapter, a different formulation for subtour elimination is discussed. Finally, (3.7) ensures the integrity of each decision variable by ensuring it is a binary variable.

3.2.2 VRP with Time Windows

The vehicle routing problem with time windows (VRPTW) is an extension of the classical VRP, where customers must be served within a certain time interval. These time windows can be soft or hard, where a soft time window indicates a non-binding constraint that can be violated for a certain penalty cost and a hard time window indicates it cannot violated. We continue adapting the formulation of Munari et al. (2016), where the formulation stays consistent with the formulation presented in Section 3.1. We add the following parameters, decision variables and constraints to the formulation described by constraints (3.1)-(3.7).

Parameters

1. a_i: Opening time of customer i time window
2. b_i: Closing time of customer i time window

3. t_{ij}: Travel time from customer i to customer j
4. s_i: Service time of customer i, optional parameter in case there are service times

Decision Variables

1. w_i = time instant service begins at node i

Constraints

$$w_j \geq w_i + (s_i + t_{ij})x_{ij} - M_{ij}(1 - x_{ij}) \qquad \forall i \in N \setminus \{n+1\}, \forall j \in N \setminus \{0\} \tag{3.8}$$

$$a_i \geq w_i \geq b_i \qquad \forall i \in N \tag{3.9}$$

Constraint (3.8) ensures that the customer is fully served within the time window and also ensures the decision variable w_i is calculated correctly. Constraint (3.9) ensures that each customer is visited within the allowed time window. The M_{ij} represents a sufficiently large non-negative number.

3.2.3 Periodic VRP

The periodic vehicle routing problem extends PVRP requires that customers are visited in one or multiple days within a planning period (Campbell and Wilson, 2014). Each customer has a set of feasible visit options and customers must thus be assigned to a feasible visit option. Consequently, a VRP is solved for each day within the planning period. We use an adapted formulation of the PVRP by Cordeau et al. (1997). We use the same sets and parameters as defined in the formulations of the VRP and VRPTW. The only exception is that we define the depot node only as $\{0\}$ instead of $\{0, n+1\}$. The formulation is as follows:

Sets and Indices

1. Arc set: $A = \left\{(i,j)^{k,l}\right\}$, where k refers to vehicle k and l refers to the day of visit l from node i to node j
2. Set of days: $T = \{1, ...t\}$ in the planning horizon
3. Set of allowable visit combinations for customer i: P_i

Parameters

1. c_{ijkl}: Cost of travelling from node i to node j using vehicle k on day l
2. D_k: Daily maximum route execution time for vehicle k

Decision Variables

$$x_{ijkl} = \begin{cases} 1, & \text{if vehicle } k \text{ visits node } j \text{ directly after node } i \text{ on day } l \\ 0, & \text{otherwise} \end{cases}$$

$$a_{rl} = \begin{cases} 1, & \text{if day } l \text{ belongs to visit combination } r\text{, with } r \in P \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ir} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to visit combination } r \in P_i \\ 0, & \text{otherwise} \end{cases}$$

Objective Function

$$min \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{l \in T} c_{ijkl} x_{ijkl} \tag{3.10}$$

Constraints

$$\sum_{r \in P_i} y_{ir} = 1 \qquad \forall i \in C \tag{3.11}$$

$$\sum_{j \in N} \sum_{k \in K} x_{ijkl} - \sum_{r \in P_i} a_{rl} y_{ir} = 0 \qquad \forall i \in C, \forall l \in T \tag{3.12}$$

$$\sum_{i \in N} x_{ihkl} - \sum_{j \in N} x_{hjkl} = 0 \qquad \forall h \in N, \forall k \in K, \forall l \in T \tag{3.13}$$

$$\sum_{j \in C} x_{0jkl} \leq 1 \qquad \forall k \in K, \forall l \in T \tag{3.14}$$

$$\sum_{i \in N} \sum_{j \in N} q_i x_{ijkl} \leq Q_k \qquad \forall k \in K, \forall l \in T \tag{3.15}$$

$$\sum_{i \in N} \sum_{j \in N} (c_{ijkl} + d_i) x_{ijkl} \leq D_k \qquad \forall k \in K, \forall l \in T \tag{3.16}$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijkl} \leq |S| - 1 \qquad \forall k \in K, \forall l \in T, S \subseteq V \setminus \{0\}, |S| \geq 2 \tag{3.17}$$

$$x_{ijkl} \in \{0, 1\} \qquad \forall i, j \in N, \forall k \in K, \forall l \in T \tag{3.18}$$

$$y_{ir} \in \{0, 1\} \qquad \forall i \in C, r \in P_i \tag{3.19}$$

The objective function (3.10) has been changed to include a four-indices decision variable. Constraint (3.11) ensures one allowable visit combination is assigned to each customer and constraint (3.12) ensures that the customer is only visited on days corresponding to that combination. Constraints (3.13) and (3.14) are similar to constraints (3.4) and (3.5) respectively but are specifically tailored to the PVRP. Limits on daily capacity and route duration are ensured through constraints (3.15) and (3.16) respectively. A more general subtour elimination formulation is provided by constraint (3.17). Finally, constraints (3.18) and (3.19) ensure the integrity of each decision variable by ensuring it is a binary variable.

3.3 Heuristics and metaheuristics

Numerous ways have been developed to find optimal solutions to vehicle routing problems, which are known as exact algorithms (Braekers et al., 2016). The Clarke and Wright heuristic (Clarke and Wright, 1964) and integer linear programming, a mathematical optimization program are one of the first approaches to solve the VRPs. Lenstra and Kan (1981) were the first to prove that the VRP is an NP-hard problem, meaning that as the number of nodes increases, so does the complexity of the problem, making it computationally hard to solve. Exact algorithms thus become inefficient at solving large problem instances, so heuristics and meta-heuristics were developed to get near-optimal solutions in a reasonable amount of time. In many real-world instances, VRPs with often complex constraints and variants of the VRP, exact methods are too time-consuming to solve.

There are two main types of exact heuristics namely constructive and improvement heuristics. The improvement heuristics improve solutions to local optima, a solution that cannot be improved further within its neighbourhood. In contrast, constructive heuristics make an initial solution to the problem (Laporte, 2009). Metaheuristics can also be divided into two main categories; local search and population-based metaheuristics. Local search explores a solution space by moving each iteration to another solution in the solution space. Secondly, population-based metaheuristics take a population of solutions and aim to improve the population through various techniques mimicking biologic evolution (Laporte, 2009). Campbell and Wilson (2014) introduced the following key constructive heuristics for PVRPs; Clarke and Wright savings algorithm, nearest neighbour, and two-stage heuristics were generally the first explored heuristics and are now mostly used for creating initial solutions. The main improvement heuristics are based on simple operations such as moving, swap and 2-opt. Laporte (2009) outlined the distinction between intraroute and interroute moves, where intraroute moves aim to improve each route separately and interroute aims to improve multiple routes simultaneously.

Metaheuristics to solve instances of the VRP are a staple in most VRP research (Braekers et al., 2016). Due to the nature of the PVRP and its relationship to the MDVRP solution methods including metaheuristics are proven to be effective in both cases (Campbell and Wilson, 2014). In the same review Campbell and Wilson (2014) identify that the most commonly used metaheuristics to solve PVRPs are tabu search (TS) and Variable Neighbourhood Search (VNS). Other significant metaheuristics are ant colony optimization (ACO), genetic algorithm and greedy randomised search procedure (GRASP), however, due to the scope of this research these techniques will not be discussed further. For a comprehensive overview of these metaheuristics, the reader is referred to Gendreau et al. (2008). Sections 3.3.1-3.3.2 discuss these main metaheuristics for PVRP and Section 3.3.3 discusses a construction heuristic.

3.3.1 Tabu search

Tabu Search (TS) is a local search metaheuristic developed by Glover (1986) that explores the solution space and escapes local optima through iterations. Tabu search has been applied to many variants of the VRP, with Cordeau et al. (1997) applying it to PVRP, MDVRP and PTSP. Three years later, Cordeau et al. (2001) would adapt its tabu search algorithm to include VRPs with time windows, with the MDVRPTW and PVRPTW as illustrations. The TS algorithm starts with an initial solution and explores its neighbouring solutions by applying local moves, like reinsertion, 2-opt and swap (Cordeau et al., 2001; Campbell and Wilson, 2014). The solution including the neighbouring solutions is then assessed based on the objective function or other KPIs. To avoid cycling back to the same solution, recent solutions are declared tabu. So, if a solution is tabu the algorithm cannot choose that solution unless a certain aspiration level or criterion is met. After each iteration, the tabu list is updated and goes to the next iteration until a stopping criterion is met, which is a maximum number of iterations or achieving a solution satisfying a certain objective. The tabu search algorithm allows for intermediate infeasible solutions to diversify the solution space. A simplified tabu search heuristic is based on the Cordeau et al. (1997) heuristic is provided in Algorithm 5 in Appendix A.

3.3.2 Variable neighbourhood search

Variable neighbourhood search (VNS) is a metaheuristic, whose main idea is to systematically change neighbourhoods. VNS thus exploits the solution space by descending to local minima and diversifies the search by systematically changing neighbourhood, which is escaping the valleys that the local minima are situated in (Hansen et al., 2010). VNS has numerous extensions and can have a high level of sophistication to exploit a VRPs unique characteristics. A basic VNS algorithm based on Hansen et al. (2010) is provided in Algorithm 6 in Appendix A. One of the advantages of VNS is that it requires relatively few parameters making it relatively easy to tune compared to other metaheuristics like TS. VNS has many extensions, for instance, variable neighbourhood descent (VND). In VND the current solution is improved for all its neighbourhoods, while in the basic VNS explores the neighbourhoods and only performs a local search.

3.3.3 Greedy algorithm

Greedy algorithms are heuristics that focus on making locally optimal decisions. Greedy algorithms thus often do not produce globally optimal solutions, but can quickly yield locally optimal solutions in a short amount of time (Zhang et al., 2016). Greedy heuristics are most often used in construction heuristics like insertion. Algorithm 7 in Appendix A provides a greedy insertion heuristic for the initial route solution. Greedy insertion treats the decision of inserting a customer in a route as the local decision and inserts customers one by one into the routes (Liu et al., 2023). Greedy algorithms are often used for generating initial solutions, due to their efficiency. The solution space is much smaller in greedy algorithms because it only makes decisions based on small local decisions like choosing which customer to visit next instead of trying to find the optimal route for that day. Incorporating a greedy algorithm for construction can thus quickly yield feasible solutions.

3.4 Conclusion

This chapter addresses the following research question *Which methods are applicable for solving the Eurofins transportation problem according to scientific literature?*. The characteristics of the Eurofins routing problem were quickly identified to be a complex vehicle routing problem. Through exploration of the VRP literature and looking for variants applicable to the Eurofins situation, a small gap remains to match the characteristics of the VRP variant to the Eurofins situation. Namely, the dropoff locations or fixed locations at the end of the route in combination with time windows, and periodicity were not found. There are ways of incorporating these characteristics using constraints. Out of the VRP variants reviewed (VRP, VRPTW, PVRP), the closest variant was found in Cordeau et al. (2001), which combines the PVRP and VRPTW to create a Periodic Vehicle Routing Problem with Time Windows (PVRPTW). The variation shares most characteristics with the Eurofins situation, however, it does not incorporate the dropoff aspect. This shows the complexity and specificity of the Eurofins situation making a tailored approach necessary. This chapter also discussed solution methods for solving the Eurofins situation and it become clear that a metaheuristic approach is preferable over exact methods. Limited exact methods were developed for this problem and only problems to the complexity of the PVRP are shown to be successful. Additionally, the scale of the problem of around 80 to 100 customers makes exact methods undesirable, due to their computational requirements due to the nondeterministic polynomial growth of the solution space with the addition of more variables and constraints. This makes metaheuristics the most suitable solution approach since they can find solutions within a realistic amount of time considering the problem scale. Metaheuristics are a more high-level heuristic that can balance the solution quality and computational time, providing a reasonable solution within a realistic amount of time. For the PVRPTW, two main metaheuristic approaches emerge as candidates for solving the Eurofins situation, tabu search (TS) and variable neighbourhood search (VNS).

For this thesis tabu search is chosen as the main metaheuristic approach. VNS is also a very suitable metaheuristic since it can systematically explore multiple neighbourhood structures allowing an extensive search of the solution space. Tabu Search even though it explores a more limited neighbourhood structure, can exploit characteristics of the PVRPTW better, namely choosing the visit schedule of a customer. Because choosing the schedule of a customer is likely to be the most influential decision on route quality, the Tabu Search can solely focus on the neighbourhood that represents this characteristic the best, whilst VNS also explores less influential neighbourhoods. This schedule allocation is likely to be the determining factor determining the quality of the solution. Tabu search can focus on this particular neighbourhood structure and get to a good quality solution quicker than VNS since it only explores this neighbourhood structure. The complexity of the Eurofins problem makes Tabu Search a good approach and can be adapted to include the dropoff characteristic of the Eurofins problem. TS in combination with a neighbourhood structure to exploit the periodicity component of the Eurofins situation is crucial for achieving a good quality solution to the problem.

The literature provides insights into a wide range of VRP variants and its solution methodologies, however, a variant that addresses all characteristics of the Eurofins situation was not found necessitating a novel approach. Due to the closeness of the Eurofins situation to the PVRPTW, solution methodologies that are effective on the PVRPTW are also identified as potential effective solution methodologies for the Eurofins situation. Tabu Search is chosen as the solution method, due to its effectiveness in large-scale PVRPTW scenarios and its adaptability to include more complex constraints and variables. This thesis thus aims to solve the complex routing problem by using a tabu search algorithm adapted to Eurofins's needs.

Chapter 4

Mathematical model: PVRPTWD

The literature review concludes that Eurofins's inspector planning problem contains several characteristics found in the literature related to vehicle routing problems (VRPs). This chapter discusses the model, which is used to answer the research question *What is an effective solution design for the Eurofins transportation process?*. This chapter starts with presenting the transportation model for Eurofins, including the model's underlying requirements and assumptions in Section 4.1. Section 4.2 presents the mathematical formulation of the model. Section 4.3 discusses the main solution methodology for the model with a Tabu Search Algorithm being the core metaheuristic approach. Finally, Section 4.4 provides a conclusion.

4.1 Model description

The Eurofins transportation can be described simply as a periodic vehicle routing problem with time windows and drop-offs (PVRPTWD) and can be defined as follows.

Assume $G = (V, A)$ to be a complete directed graph, where $C = \{1, ...n\}$ is the set of nodes that contains all customer nodes, and $F = \{0\}$ represents the home of the inspector and $D = \{n+1, ..n+m\}$ represents drop-off locations. So, we define the node set $V = C \cup F \cup D = \{0, 1, .., n+m\}$ and the arc set A, defined by $A = \{(i, j) : i, j \in V, i \neq j\}$. Finally, let $T = \{1, ..., \tau\}$ be the set of time periods in the planning horizon with a time period being equal to one working day and the horizon one month. Vertex set D represents the set of uncapacitated dropoff locations and N represents the set of customers to be served. The vehicles are assumed to be incapacitated. For each customer $i \in C$ a predefined set of allowable visit combinations P_i is associated, where a combination $p \in P_i$ consists of the specific days a customer can be visited with $p \subseteq T$. The days a customer can be visited is dependent on the customer's visit frequency e_i and other constraints. For instance if we have a planning horizon of 3 days and customer i needs to be visited 2 times, $P_i = \{(1,2), (2,3), (1,3)\}$. For each arc (i, j) we associate travel cost c_{ij}, which is equal to the travel time from travelling from node i to j. Furthermore, for each customer $i \in C$ a non-negative service time s_i is defined. A specified time interval a customer must be served within, $[a_i, b_i]$, which we refer to as a time window. The dropoff locations D, also have a specified time window but have no service time. In case a vehicle arrives before the opening time the vehicle must wait until the time window opens at time a_i before serving customer i. On a given day the total time to execute a route cannot exceed S_{max}.

The PVRPTWD aims to select a visit schedule for each customer and design routes to minimise the total time spent executing the routes. The model aims to achieve this objective while satisfying the requirements listed below in Section 4.1.1 and considering the model assumptions discussed in Section 4.1.2.

4.1.1 Requirements

This section lists the requirements for the VRP. The requirements listed below are included in the model formulation using constraints.

1. Each customer must be visited once or multiple times on different days according to the visit frequency within the planning horizon.
2. Each customer has to be fully served when visited.
3. The time windows of the customers, as well as the dropoff points, have to be adhered to.
4. Each vehicle starts and ends each trip at home.
5. Each vehicle must visit a dropoff location before returning home.
6. The total time to execute a route on a given day does must not exceed 10 hours unless specified otherwise.

4.1.2 Assumption

The following assumptions have been made to effectively model the Eurofins situation.

1. The service times of the customers are considered deterministic.
2. The opening times of all customers are assumed to be fixed.
3. The travel time is considered to be the ratio between the distance in kilometres and a constant average speed in kilometres per hour.
4. The geodetic distance is considered the distance between 2 nodes.
5. The routes are scheduled immediately before the execution of the first route begins.
6. All customer orders (number of visits, time windows and service times) are known within the planning horizon.
7. The number of visits for a customer is considered deterministic.
8. The drivers have no limitation in working hours and are limited only to the maximum amount of working hours each day.
9. The capacity of the vehicles is infinite.
10. The inspections take place on normal working days, Monday to Friday.
11. The dropoff locations are considered to be uncapacitated.

4.2 Mathematical formulation

The mathematical formulation of the Eurofins problem is based on the models presented by Ahmadi Basir et al. (2024) and Cordeau et al. (1997). Our proposed model below takes the MTZ-based (Miller et al., 1960) formulation proposed in Ahmadi Basir et al. (2024) as the basis of the mathematical model and adds additional constraints to reflect the requirements and assumptions. The model is formulated as a Mixed Integer Linear Program (MILP) and begins by setting the parameters and defining the set and indices of the Eurofins inspection planning problem. Consequently, the decision variables, objective function and constraints are defined based on the requirements (Section 4.1.1).

4.2.1 Sets and indices

Notation	Description	Value
C	Set of customer nodes	$\{1, 2, ..., n\}$
F	Set of inspector home node	$\{0\}$
D	Set of dropoff nodes	$\{n + 1, ..., n + m\}$
V	Set of all nodes	$C \cup F \cup D$
T	Set of days in the planning horizon	$\{1, ..., \tau\}$
P_i	Set of possible schedules of customer i	$p \in P_i, p \subseteq T$
K	Set of homogeneous vehicles	$\{1, ..., \kappa\}$

Table 4.1: *The sets and indices of the MILP model*

4.2.2 Parameters

Notation	Description	Range
c_{ij}	Travel time from node i to node j	$c_{ij} \geq 0$
S_{max}	Maximum route execution time of inspectors	10, unless specified otherwise
a_i	Start time of customer i's time window	
b_i	Start time of customer i's time window	$b_i > a_i$
r_i	Service time of customer i	$r_i \geq 0$
e_i	Visit frequency of customer i in time horizon	$1 \leq e_i \leq \tau$
M	Sufficiently large positive constant	

Table 4.2: *Parameters of the MILP model*

4.2.3 Decision variables

The decision variables of the model are defined as follows:

$$x_{ijt} = \begin{cases} 1 & \text{if vehicle travels directly from node } i \text{ to node } j \text{ on day } t \\ 0 & \text{otherwise} \end{cases}$$

$$s_{ip} = \begin{cases} 1 & \text{if schedule } p \in P_i \text{ is chosen to visit customer } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_{it} = \begin{cases} 1 & \text{if customer } i \text{ is visited on day } t \\ 0 & \text{otherwise} \end{cases}$$

w_{it} = start time of serving customer i at day t

4.2.4 Objective function

The function to minimise total time spent on executing the routes can be formulated as:

$$min \sum_{i \in V} \sum_{j \in V} \sum_{t \in T} c_{ij} x_{ijt} + \sum_{i \in C} r_i f_i \tag{4.1}$$

The objective function is minimizing the total route execution time. The travel costs c_{ij} represent the time it takes to go from node i to node j and r_i represents the time of visiting customer i.

4.2.5 Constraints

The constraints which are defined in line with the model requirements and assumptions for the Eurofins routing problem are listed below. We subject the model to the following constraints:

$$\sum_{p \in P_i} s_{ip} = 1 \qquad \forall i \in C \tag{4.2}$$

$$\sum_{p \in P_i : t \in p} s_{ip} = y_{it} \qquad \forall i \in C, \forall t \in T \tag{4.3}$$

$$x_{ijt} \leq \frac{y_{it} + y_{jt}}{2} \qquad \forall i, j \in C, t \in T \tag{4.4}$$

$$\sum_{j \in C} x_{0jt} \leq |K| \qquad \forall t \in T \tag{4.5}$$

$$\sum_{j \in C \setminus \{i\}} x_{ijt} = y_{it} \qquad \forall i \in C, \forall t \in T \tag{4.6}$$

$$\sum_{j \in C \setminus \{i\}} x_{jit} = y_{it} \qquad \forall i \in C, \forall t \in T \tag{4.7}$$

$$\sum_{i \in C} x_{i0t} = 0 \qquad \forall t \in T \tag{4.8}$$

$$w_{it} + r_i + c_{ij} - w_{jt} \leq M\left(1 - x_{ijt}\right) \qquad \forall i \in V, \forall j \in V \setminus F, \forall t \in T \tag{4.9}$$

$$a_i \leq w_{it} \leq b_i \qquad \forall i \in N \setminus F, \forall t \in T \tag{4.10}$$

$$\sum_{i \in V} \sum_{j \in V} \left(r_i + c_{ij}\right) x_{ijt} \leq S_{max} \qquad \forall t \in T, i \neq j \tag{4.11}$$

$$x_{ijt} \in \{0,1\} \qquad \forall i, j \in V, \forall t \in T, i \neq j \tag{4.12}$$

$$s_{ip} \in \{0,1\} \qquad \forall i \in C, \forall p \in P_i \tag{4.13}$$

$$y_{it} \in \{0,1\} \qquad \forall i \in C, \forall t \in T \tag{4.14}$$

In this formulation constraint (4.2) ensures that a visit schedule is selected for each customer. Constraint (4.3) relates the schedule selection variable with the customer visit variable. Constraint (4.4) guarantees that only arcs of customer pairs assigned to the same day can be used. The number of vehicles, which in the Eurofins case of the PVRPTWD is equal to one, is ensured in Constraint (4.5). Constraints (4.6) and (4.7) ensure that each customer is visited exactly once on a given day, which also implies flow conservation. Constraint (4.8) ensures that the route can only return to the inspector via the dropoff node by forbidding travel from a customer node to the home node. Constraints (4.9) and (4.10) together guarantee that the time windows are respected in the routes. Note that the large positive constant M may be equal to $max_{(i,j) \in A} \{a_i + r_i + c_{ij} - b_j\}$. Additionally, Constraint (4.9) ensures subtour elimination. Constraint (4.11) ensures that on a given day the maximum amount of working hours is not exceeded. Finally, constraints (4.12)-(4.14) ensure the integrity of the decision variables.

4.3 Solution methodology

The problem is formulated as a Mixed Integer Linear Program (MILP). The MILP is formulated using 14 constraints, 3 binary decision variables and 1 continuous variable. Due to this complexity and increase in the number of customer nodes and potential dropoff nodes, the complexity and computation time increase exponentially. The problem is thus NP-hard making the problem likely solvable in small instances, but infeasible to solve to optimality in larger instances. For example, in the real-world scale of this problem of having 2 dropoff locations, and 97 customers over a time horizon of 20 days. We have 200000 binary x_{ijt} variables (100 x 100 x 20), 2000 y_{it} variables, 2000 w_{it} variables, and around 2000 s_{ip} variables depending on the amount of allowable schedules. This encompasses a scenario of up to 208000 variables for the real-world scenario. Because of this complexity in the larger instances using exact methods, tabu search as a metaheuristic approach is expected to be more efficient. Due to the closeness of the PVRPTWD, to the PVRPTW a Tabu Search metaheuristic is identified as a suitable heuristic for solving the PVRPTWD. Tabu Search is proven to be one of the most effective metaheuristics when it comes to periodic vehicle routing problems and their extensions.

4.3.1 Initial route construction

The first step in the solution methodology is constructing initial routes. The initial route creation serves as a relatively quick way to find a feasible initial solution to reduce solving time since the tabu search heuristic does not need to spend excessive time on finding an initial feasible solution. The initial solution is crafted as follows and is inspired by the construction heuristic presented in Cordeau et al. (2001), customers i are first assigned a random combination $p \in P_i$ and a random dropoff location is selected $d \in D$ for every day $t \in T$. Then, for each day the customers (that are to be visited based on visit combination p) and the random dropoff location are sorted based on the angle between the inspector's home and dropoff location θ, with the random dropoff location being the last element in the sorted list. Starting with the first element of this list, the locations are inserted into the route for that day one by one, minimizing the increase in total travel time for the route. The pseudocode presented in algorithm 1 displays this procedure.

Algorithm 1 Basic initial route construction

```
InitialRoutes ← ∅
VisitCombination ← AssignRandomVisitCombination(P)
for day ∈ T do
    route ← ∅
    CustomersForDay ← AddCustomer(customer, VisitCombination)
    DropoffDay ← RandomDropOff(D)
    SortedCustomers ← SortByAngle(CustomersForDay)
    for customer ∈ SortedCustomers do
        position ← FindBestInsertion(customer, c, route)
        route ← Insert(position, customer)
    end for
    InitialRoutes(day) ← AddRoute(route)
end for
InsertInitialRoutes(InitialRoutes)                          ▷ Inject initial solution
```

Algorithm 1 has a lot of randomness in the creation of the solution especially in assigning visit combinations. This creates a high probability of having some long and some very short days. Additionally, since a depot gets randomly assigned to a day we improve the algorithm by pairing customers to the depot based on geographic positioning. Because the depots determine the end of the routes, we can exploit this characteristic to potentially get better solutions. To address the issue of random depot allocation and daily route times with high variance, a customized algorithm is developed based on Algorithm 1 that is specifically tailored to address these weaknesses. The concept of assigning combinations based on spreading service times comes from Chao et al. (1995).

The idea of exploiting the dropoff location is created by observing the performance of Algorithm 1. The proposed algorithm is presented in Algorithm 2.

Algorithm 2 Customized initial route construction

InitialRoutes $\leftarrow \emptyset$
CustomerDepot $\leftarrow AssignCustomerToDepot(customers, depot)$
DepotDays $\leftarrow AssignDaysToDepot(CustomerDepot, Days)$
$VisitCombination \leftarrow AssignVisitCombinationEqualSpread(CustomerDepot, DepotDays, P)$
for $\forall t \in T$ **do**
    $route \leftarrow \emptyset$
    $CustomersForDay \leftarrow AddCustomer(customer, VisitCombination)$
    $DropoffDay \leftarrow RandomDropOff(D)$
    $SortedCustomers \leftarrow SortByAngle(CustomersForDay)$
    for $customer \in SortedCustomers$ **do**
        $position \leftarrow FindBestInsertion(customer, c(i, j), openwindow(i), route)$
        $RouteInsert(position, customer)$
    end for
    $InitialRoutes(day) \leftarrow route$
end for
$InsertInitialRoutes(InitialRoutes)$ ▷ Inject initial solution

Algorithm 2 is similar to 1, however, there is a more deterministic methodology behind assigning a visit schedule to a customer. Firstly, customers get assigned to the closest depot in *CustomerDepot*. This is followed by assigning random days to the depots based on the number of customers assigned to that depot, so depots with more customers assigned to them get assigned more days in *DepotDays*. Finally, instead of a visit combination getting randomly chosen for a customer, a two-phase process is used in *AssignVisitCombinationEqualSpread*. The goal of this function is to find a schedule for each customer that would result in an even spread of total daily service times for each depot in order to have less variance in daily route times. In this process, the customers get sorted based on decreasing visit frequency. In the first phase, based on the days assigned to the depot the customer is also assigned to valid visit combinations get filtered. A valid visit combination is a combination whose days are also in the days assigned to the depot. In case no such schedule can be found a random combination is selected. In the second phase, the visit schedule is chosen based on the schedule that results in the minimum amount of service time. The visit combination selection for a customer is finalised by adding the service times corresponding to the visit schedule to the respective days. So for example, if an arbitrary customer has the decision to choose between schedule 1: (2) and schedule 2: (5) the aggregated service times for days 2 and 5 are 4.5 hours and 5.5 hours respectively. Schedule 1 gets chosen because it results in the lowest aggregated service time, in case a schedule with multiple days needs to be chosen it is the sum of these aggregated service times.

4.3.2 Tabu search

The tabu search algorithm is the metaheuristic that is implemented to solve the mathematical model. The tabu search heuristic described here is an adapted version of the tabu search heuristic presented in (Cordeau et al., 2001). For the tabu search, we define the following: $s \in S$, where S represents the set of solutions s formed by routes that satisfy all constraints except the maximum route duration constraint and time window constraints (Constraints (4.11) and (4.10) respectively). These constraints may be violated at a certain penalty cost. To include these violations in the mathematical model, we formulate the following evaluation function based on concepts introduced in Chao et al. (1995).

$$min\ f(s) = c(s) + \alpha * dv(s) + \gamma * twv(s) \tag{4.15}$$

Here, $c(s)$ represents the total travel time of the routes for a solution s and is equal to the Equation (4.15), $dv(s)$ is the total violation of the duration constraints and $twv(s)$ represents the violation of the time window constraints. The hyperparameters α and γ influence the TS to choose solutions that violate these constraints and are dynamically adjusted in a later explained mechanism. A duration constraint violation is simply equal to the non-negative difference in route execution time for a day and the maximum route duration, S_{max}. A time window violation is expressed as the difference between arrival time and closing time $(w_{it} - b_i)^+$, with $x^+ = max\{0, x\}$, x is in this case equal to $w_{it} - b_i$. If a vehicle arrives before the open window a_i the vehicle in the model waits for $o_i = w_{it} - a_i$. We integrate these constraint violations into the tabu search heuristic using non-negative slack variables. The slack variables are there to quantify the violation of individual constraints. So we define the slack variables for time windows violations as $SlackTW_{it}$ for $\forall i \in C, \forall t \in T$ and for duration violations as $SlackDuration_t$ for $\forall t \in T$ according to the aforementioned equations and descriptions. Based on constraints (4.11) and (4.10) we define the following equations for the tabu search:

$$a_i \leq w_{it} \leq b_i + SlackTW_{it} \qquad \forall i \in C, \forall t \in T \tag{4.16}$$

$$\sum_{i \in V} (w_{it} + c_{ij}) \leq S_{max} + SlackDuration_t \qquad \forall t \in T \tag{4.17}$$

$$SlackTW_{it} \geq 0 \qquad \forall t \in T, \forall i \in V \tag{4.18}$$

$$SlackDuration_t \geq 0 \qquad \forall t \in T \tag{4.19}$$

Consequently, we define the total duration violation (4.20) and time window violations (4.21) used in Equation (4.15) equal to:

$$dv(s) = \sum_{t \in T} SlackDuration_t \tag{4.20}$$

$$twc(s) = \sum_{t \in T} \sum_{i \in C} SlackTW_{it} \tag{4.21}$$

The tabu search algorithm takes the solution produced in the initial route construction and begins iterating over the following steps. The tabu search chooses the best non-tabu solution in its neighbourhood $N(s)$. After choosing a solution α and γ get updated based on whether or not the duration constraint and time window constraints are violated. α gets multiplied by $1 + \delta$, with δ being a non-negative number if the solution adheres to the duration constraints and gets divided by $1 + \delta$ if it does not. For γ the same process applies but with respect to the time window violations. To determine if constraints are not violated, we check the total slack of the constraints, so $dv(s)$ or $twv(s)$ is equal to 0. The tabu search is repeated until a termination condition is met, in this case, a time limit. During this process, the best solution and costs $s*$ and $c*$ are stored.

Furthermore, we define θ to be the tabu list length. After selecting a new solution s, where customers i get reinserted into different days t the insertions (i, t) get assigned the tabu attribute and put at the front of the tabu list, which is based on the insertion presented in Algorithm 4. The use of insertion (i, t) will thus be forbidden for the iterations when (i, t) is in the tabu list. (i, t) will lose its tabu status when it is at the back of the tabu list and a new attribute gets inserted while the list is full. If a solution is found that would result in a smaller cost than the overall best solution, the tabu status of the attributes that would prohibit choosing that solution is revoked. The aforementioned procedure is called the aspiration criterion. The final element of the tabu search is a diversification mechanism using the penalty factor of a solution $p(s)$, we evaluate solutions based on the equation (4.15), $f(s)$ plus the penalty factor $p(s)$. The penalty factor is part of the diversification strategy, where we incentivise the search to look into unexplored or lesser-explored regions when the search finds itself trapped in local optima. The penalty cost is based on the attribute ρ_{it}, which represents the number of times attribute (i, t) is added to the solution in the tabu search. The formula for the penalty cost is provided in Equation (4.22):

$$p(\bar{s}) = \lambda c(\bar{s})\sqrt{nK} \sum_{(i,t)\in B(\bar{s})} \rho_{it} \tag{4.22}$$

Here in Equation (4.22), λ represents the strength of the diversification mechanism. The other components scale the penalty cost relative to the objective cost and problem size. In case the solution $\bar{s}$ would result in a better evaluation cost $f(\bar{s})$ than the current solution $f(s)$ we set $p(\bar{s}) = 0$. Algorithm 3 presents the tabu search (pseudo)code.

Algorithm 3 Tabu Search

```
Current route: s ← InitialRoutes                                      ▷ Algorithm 1
α ← 1
γ ← 1
tabu list ← ∅
tabu list lenght: θ
Best solution: s*
Best cost: c(s*)

if s is feasible then
    s* ← s
    c(s*) ← c(s)
else
    c(s*) ← ∞
end if

while time < TimeLimit do
    N(s) ← ExploreNeighbourhood(s)
    NewSolution ← ∅
    NewSolutionCost ← ∞
    for s̄ ∈ S do
        NeighbourCost(s̄) ← f((s̄) + p(s̄))
        if NeighbourCost(s̄) < c(s*) then                          ▷ Aspiration criteria
            NewSolution ← s̄
            NewSolutionCost ← NeighbourCost(s̄)
        else if s̄ IsNotTabu then                ▷ Check if solution uses move on tabu list
            if NeighbourCost(s̄) < NewSolutionCost then
                NewSolution ← s̄
                NewSolutionCost ← NeighbourCost(s̄)
            end if
        end if
        if NewSolutionCost < c(s*) s̄ is feasible then
            s* ← NewSolution
            c(s*) ← NewSolutionCost
        end if
        UpdateAlpha(NewSolution, slack duration, δ)
        UpdateGamma(NewSolution, slack timewindow, δ)
        UpdateTabuList(θ, NewSolution)
        s ← NewSolution
    end for
end while
return s*
```

4.3.3 swap and reinsert

The Algorithm 4 displays the basic operation used in the TS used in this solution approach. The algorithm consists of two main parts. The core idea behind the algorithm is to explore the neighbourhood of feasible solution s by reinserting customer i into different visit combinations r. For this transformation to work and keep track of the tabu status we define $B(s) = \{(i,t) \mid \text{customer } i \text{ is visited on day } t\}$. We also define b_{pt}, which is equal to 1 if day t belongs to visit combination r.

The first part of the algorithm replaces the visit combination p of customer i with another visit combination called $p' \in P_i$. After this replacement, we remove the customer or add the customer based on this new visit combination p'. See appendix C Figures C.1 and C.2 for a solution representation.

Algorithm 4 Swap visit combination

```
Feasible solution s:
Customers n:
P: all possible combinations of customers and visit combinations
i ← RandomCustomer(n)
p ← CurrentVisitCombination(P_i)
p' ← NewRandomVisitCombination(P_i)
for t ∈ T do
    if b_pt = 1 & b_p't = 0 then
        x_jit, x_iht = 0 ∀j, h ∈ V            ▷ Remove customer from route
        x_jh = 1                              ▷ Link predecessor and successor
    else if b_pt = 0 & b_p't = 1 then
        Insert(i, t)                          ▷ Insert customer i on day t, minimise increase in f(s)
    end if
end for
return B(s)
```

4.4 Conclusion

In this chapter, we discussed the proposed solution methodology to address the Eurofins's situation as well as the underlying mathematical model describing this situation. This chapter answers the research question: *Which model and technique can be implemented to solve the Eurofins's transportation problem?*. The Eurofins's routing problem was first formulated as a Periodic Vehicle Routing Problem with Time Windows and Dropoffs (PVRPTWD). The model adheres to a set of assumptions and requirements which are the following; each customer must be visited a specified number of times within the planning horizon. A customer can only be visited once per day and each customer is fully served when visited. Each vehicle starts at home and ends at home on a given day and each vehicle must visit a dropoff location before returning to home. The vehicle is constrained so that the total time to execute a route on a day must not exceed 10 hours. Finally, the customers must be served within a specified time interval called a time window.

Furthermore, assumptions have been made to create a feasible solution and for its practicality. Firstly, service times and opening times are fixed. Secondly, the travel time between 2 points is considered to be the ratio between the geodetic distance and an average speed and is thus also deterministic. All customers and orders are known before scheduling. Inspectors need to take and transport samples, however, if the samples are small enough in size, the capacity of the vehicle can be considered infinite. The inspections take place during normal working hours from Monday to Friday. The solution approach is using a Mixed Integer Linear Program (MILP) as the base formulation of the problem, and a tabu search metaheuristic to solve the larger instances. Tabu search is used to solve the model since it provides good diversification and exploitation mechanics to solve relatively large problem instances. An initial solution is constructed and applied before initiating the tabu search to speed up optimization times. The initial solution is created by assigning random visit combinations to customers and applying a greedy insertion heuristic to find an initial solution. The core principle of the tabu search is to quickly explore the solution space through simple neighbourhood operations and move out of local optima through a diversification strategy using penalties. Additionally, a tabu list, a list with forbidden moves is used to prevent cycling. This overall solution approach addresses the Eurofins routing problem and demonstrates how a tabu search heuristic can be applied to a real-world vehicle routing problem.

Chapter 5

Solution evaluation

This chapter evaluates the proposed solution methodology of using a Tabu Search heuristic to solve the Eurofins routing problem. The aim of this chapter is to assess the solutions created by the solution methodology against the current performance of Eurofins. We thus address the following research question: *How does the chosen solution approach perform compared against to the current situation?*. We begin this chapter by outlining the experiment design in Section 5.1. Section 5.2 describes the data instances used in the experiments, covering both artificial and real-world data instances. The model parameters for the tabu search are covered in Section 5.3 followed by the results from the testing using these parameters in Section 5.4.

Note: to maintain confidentiality values in Section 5.4 have been multiplied by an arbitrary number.

5.1 Experiment Design

The experiment design is split up into two distinct phases. Labeled $P1$ and $P2$ respectively. The first experiment $P1$ is dedicated to the parameter tuning for the tabu search and MILP. The MILP is solved using a combination of exact and approximate solution techniques by using a general-purpose solver, Section 5.1.1 provides details of this solver. In this chapter when we refer to the MILP the solution approach using the MILP solver is meant. Parameter tuning is an important part of the experiment design since good parameter tuning can aid in finding good quality solutions within a feasible timeframe. This first phase goal is to find a suitable time parameter for the tabu search and MILP, other parameters are also discussed. Furthermore, parameters for the initialization phase are also addressed. For the parameter tuning phase, 10 artificial data sets are created based on the company's data. These data sets are created to have a larger sample size and test the general applicability of the model on a more diverse set of data as opposed to the similar sized data sets from Eurofins.

Secondly, after the parameter tuning phase has concluded, the algorithms are applied to three real-world data instances from Eurofins in experiments $P2$. These data instances are based on the planning of three inspectors over a month. These data instances are used to evaluate the inspector's planning performance against the proposed algorithms. The aim is thus to have the solution method's effectiveness evaluated in terms of planning to reduce the number of hours to execute a monthly workload. Finally, an analysis between the KPIs from the current inspectors' planning and the solution methodology's planning is performed to look for patterns explaining potential discrepancies in KPI values.

5.1.1 Technical specifications

The experiments and solving the data instances were conducted on a laptop equipped with an AMD Ryzen 5 5600H processor, 3301 MHz and 16GB RAM. The MILP and tabu search heuristic were both implemented and used, using the PyCharm Community Edition 2023.2.1 IDE to integrate

Python 3.11.5 64-bit. The MILP is implemented using a general purpose solver, Gurobi Optimizer version 11.0.1 using an academic license.

5.2 Data instances

This section discusses the data instances used for the experiments. We use a combination of artificial and real-world data for the aforementioned experiments. The data instances used for parameter tuning are artificially created. These data instances range from 10 to 100 customers, and from 2 to 20 days to test the proposed solution methodology and smaller and larger problem instances. This wide range of data instances creates an environment, where the model can be tested on its robustness exposing potential weaknesses and showing its strengths, unlike the real-world data, which are all the same problem size avoiding overfitting. Correctly tuning the parameters is also crucial for model performance, where we can effectively weigh computational performance against solution quality. This parameter tuning process is discussed in Section 5.3 using artificial data. Secondly, the three real-world instances are discussed by providing details on the parameter calculation and characteristics of the instances. The parameters are based on the real-world data representing actual customer time windows, number of visits and other characteristics. The instances are used to test the current situation against our proposed solution methodology.

5.2.1 Artificial data

For tuning for the tabu search parameters artificial data instances ranging from 10 to 100 customers and 2 to 20 days were created based on the real-world data instances provided by Eurofins. The number of days is proportional to the amount of customers, with 5 customers accounting for one day. The customer node locations are randomly generated within a 50 by 50 kilometer grid. The service times and time windows are also generated based on the Eurofins customers. In table 5.1 the 4 basic drop off location configurations can be found. A drop off configuration is a geographic description of the positioning of the dropoff locations compared to the home location of the inspector.

Configuration ID	**Number of dropoff locations**	**Geographic description**
C1	1	Dropoff location is the same as the inspector's home
C2	1	Dropoff location is located between 20-30 kilometers from inspector's home
C3	2	Dropoff locations are located between 20-30 kilometers from inspector's home
C4	3	Dropoff locations are located between 20-30 kilometers from inspector's home

Table 5.1: *Overview of drop off configurations for the artificial data instances, and real-world data instances.*

The time windows are chosen based on the opening and closing times from Eurofins client list. This methodology also applies to the number of customers and possible visit combinations, where we can exclude certain visit combinations based on if a client is closed on Friday for example. Distances between 2 nodes are calculated based on the Euclidean distance between these 2 points. In the next section we will use a different method to calculate the distance which takes the curvature of the earth into account. The distortion caused by using the Euclidean distance is however minimal due to the relatively small area. Finally, we use varying average speeds ranging from 30 km/h to 50 km/h. An overview of the data instances used can be found in table 5.2 with each data instance representing a different number of customer nodes, days, dropoff configuration, and speed.

ID	Configuration ID	Number of customer nodes	Days	Speed (km/h)
A1	*C2*	10	2	30
A2	*C1*	20	4	35
A3	*C4*	30	6	50
A4	*C3*	40	8	40
A5	*C2*	50	10	35
A6	*C1*	60	12	50
A7	*C4*	70	14	40
A8	*C3*	80	16	35
A9	*C1*	90	18	40
A10	*C2*	100	20	45

Table 5.2: *Overview of the 10 artificial datasets used for parameter tuning and performing experiments.*

5.2.2 Real-world data instances

In order to find an optimal routing strategy for the Eurofins transportation problem, 3 real-world instances are used. The data is extracted from three different inspectors with three different working areas, in terms of customer density, dropoff location positioning and average speeds. The real-world instances reflect a monthly planning an inspector gets at the beginning of each month, the data used are April or May 2024. The data includes customer locations, visit frequencies, service times, time windows and open days. The data instances are based on the monthly planning of an inspector, which amounts to an average of 20 working days, including holidays and other responsibilities where inspectors do not inspect. 3 monthly plannings from different inspectors were chosen to be analysed.

For practicality, only dropoff points that actually get used by that inspector are taken into account. The closing time of dropoff points is 18:00 unless specified otherwise. The customer data locations, which are only provided in addresses are converted into GPS coordinates using the Google Maps API tool. Additionally, the visit frequencies are based on the number of times a customer visited in a month by the inspector on different days. Spacing between visits is also taken into account by ensuring that visits to the same customer must be in different weeks. The data sets customer nodes and dropoff configuration are provided in table 5.3, the amount of customer nodes for all data sets is similar in size ranging from 88 to 100 customer nodes. Each data instance is from a different geographic of the Netherlands with a different amount of customer nodes, with different time windows, visit frequencies and allowed visit days. The variation in node density, average speed, and geographic positioning of nodes make each data instance unique and allow the proposed solution approaches to be tested in various real-world scenarios.

The average speed of the vehicles is calculated by using the GPS data from the vehicles and extracting the average speed, for time periods the vehicle is non-stationary. Since, this average speed is over the actual road distance, instead of the absolute distance used in the model, the average speed needs to be adjusted to take the lower absolute distance into account. The rule of thumb is that the actual distance on the road is 30% higher than the Euclidean distance (Boyacı et al., 2021), however due to the high road density of the Netherlands this factor is lower. Berens (1988) estimates that optimal factors for western European counties with high road density lie between 1.20 and 1.25, for this thesis, we take a detour factor of 1.25 to compensate for using Euclidean distance instead of actual road distances.

ID	Number of Customers	Dropoff ID	Average Speed (km/h)
I1	101	*C2*	48.33
I2	94	*C1*	48.50
I3	87	*C3*	62.24

Table 5.3: *Overview of real-world data instances. (values multiplied by arbitrary number for confidentiality)*

5.3 Paramater tuning

This section covers the parameter tuning of the tabu search and MILP. For both the tabu search and MILP a maximum computation time parameter is implemented using the artificial data instances discussed in Section 5.2.1. The other parameters for the tabu search itself are discussed in Section 5.3.2. Additionally, the initialization algorithm for the tabu search is chosen in Section 5.3.1. The maximum computation time for the tabu search and MILP are discussed separately in Sections 5.3.3. and 5.3.4 respectively.

5.3.1 Initialization choice

In Section 4.3.1 of the solution methodology, two algorithms were proposed for constructing an initial solution. The first algorithm randomly assigns a visit combination to each customer and uses a greedy insertion heuristic to insert customers into the routes. The second algorithm uses a more complex mechanism to assign visit combinations to customers based on the dropoff locations and the customer locations and service times. The choice of algorithm is important for selecting a visit combination, which influences the tabu search by allowing it to start in a more favourable position. So, through experiments, the most effective construction heuristic is chosen. We define the following:

- *Experiment H1*: Randomised visit combination selection. This construction heuristic creates different results each time due to randomly selection visit combinations. So, this experiment is conducted 5 times to account for variability in the solution output quality.
- *Experiment H2*: Customer-dropoff based visit combination selection. This method also contains some randomness in assigning the days to the depots. Consequently, this experiment is also conducted 5 times to account for variability in the solution output quality.

For these experiments the following values are reported; minimum, maximum and average costs. The average computation times are also reported. The results of the experiments are presented in Table 5.4. The first column corresponds to the data instance on which the algorithms were experimented. The Following 4 columns present the values corresponding with experiment $H1$ and the last 4 columns with experiment $H2$. The first 3 columns of each experiment provide the minimum, average and maximum objective values with the fourth column providing the average computation time. Note that the time reported is the time it takes to construct the routes, the time it takes to assess and inject this solution in the tabu search is not included.

ID	*H1*				*H2*			
	Objective			***Time***	***Objective***			***Time***
	Min	**Avg**	**Max**	**Time (s)**	**Min**	**Avg**	**Max**	**Time (s)**
A1	24,26	43,78	61,85	0,001	22,94	**33,59**	48,5209	0,001
A2	38,94	77,09	110,92	0,001	36,64	**44,32**	52,545	0,001
A3	59,32	98,02	150,43	**0,001**	71,28	**73,34**	75,7641	0,002
A4	123,20	163,54	195,36	**0,001**	110,01	**111,59**	113,231	0,002
A5	210,16	230,18	238,65	0,002	180,99	**190,12**	194,795	0,002
A6	115,53	149,49	165,55	0,003	96,78	**97,89**	98,513	0,003
A7	172,49	191,87	222,06	**0,007**	117,23	**120,82**	124,308	0,009
A8	295,82	338,24	391,75	0,009	189,01	**198,50**	209,844	0,009
A9	262,08	286,27	340,02	**0,010**	160,71	**175,60**	187,209	0,011
A10	303,06	332,13	358,44	0,010	216,91	**220,61**	225,881	0,010
Average	160,49	191,06	223,50	**0,0045**	120,25	**126,64**	133,06	0,0050

Table 5.4: *Summary of experiments $H1$ and $H2$. The lowest average objective and lowest average computation time are displayed in bold.*

Table 5.4 shows differences in performance between the randomised visit combination selection (*Experiment H1*) and customer-dropoff based visit combination selection (*Experiment H2*). Both algorithms have different ways of constructing the initial routes for the tabu search impacting the solution quality and time to reach a certain solution quality itself.

It is clear that the customer-dropoff-based visit combination selection (*Experiment H2*) consistently, outperforms the randomised visit combination selection (*Experiment H1*) in terms of minimizing the initial objective value. There are insights into these algorithms through these experiments. Firstly, even though there is a limited sample size for each instance of 5 experiments for each algorithm, it is clear that the randomised selection algorithm displays more variance in its solution quality, sometimes producing better results than the customer-dropoff-based algorithm. The experiments done on dataset $A3$ showcase this pattern where the randomised algorithm finds an initial solution of 59.32, while the other algorithm's best initial solution is equal to 71,28. Overall, it is clear that the customer-dropoff-based algorithm has a small range of initial solution values, while the randomised selection algorithm has a very broad range of initial solution values. This behaviour is expected, because even though both algorithms contain some kind of randomness, the random selection algorithm has considerably more randomness in selecting the visit schedules, as opposed to the dropoff based algorithm.

The random selection algorithm, however, seems to have a slight edge in computational time compared to the customer-dropoff-based algorithm across a couple of instances. However, this edge in computational time is incredibly minimal only accounting for a few milliseconds. The sample size is, however, too limited to comment on the kind of pattern the computational time experiences with an increasing amount of nodes.

In conclusion, the customer-dropoff-based visit combination selection algorithm ($ExpirimentH2$) clearly appears to be the better construction heuristic for initializing the tabu search. This algorithm produces better initial solutions compared to the randomised visit combination selection in almost all instances, while the difference in computation time is almost negligible. The customer-dropoff-based algorithm suggests that incorporating key features of the routing problem can enhance initial solution quality significantly, where in this case exploiting the positioning of the dropoff locations and spreading the amount of service time.

5.3.2 Tabu search paramaters

The tabu search has 4 main parameters to set; δ, λ, θ and $TimeLimit$. Additionally, there are initialization parameters of α and γ, which are dynamically adjusted based on route duration and time violations respectively. Because there are no clear measures available to evaluate the effectiveness of most parameters, the parameter tuning for the tabu search comes down to the evaluation of the researchers. The parameters for the tabu search for the Eurofins problem will thus, be based on the parameter values presented in Cordeau et al. (1997), since extensive testing and experimentation for these parameters was done for the unified tabu search heuristic this solution methodology is based on.

Experimentation has been conducted to validate these Cordeau parameters by observing if the search behaviour each parameter induces is evident and if it is present to its desired extent. In the following section, each parameter is covered and argumentation is provided if the desired influence each parameter induces is satisfactory. Setting the parameters for the tabu search well has a great impact on the solution quality and influences the search process significantly. So setting these parameters well to extract the desired behaviour of the tabu search is crucial for having an effective solution methodology.

Diversification: λ

The λ is responsible for diversifying the search process. This is the scalar that penalises repeating moves. Using a too small value of λ results in not having the desired diversification of the search and results in choosing solutions close to the initial solution. On the other hand, having a too large value of λ results in the tabu search choosing low quality solutions because it quickly disregards solutions in which previously made moves were used. After observing the experiments, the parameter value in Cordeau seems to have this desired behaviour of diversification, while not taking on low-quality solutions. Cordeau recommends a λ of 0.01 to 0.02, for the Eurofins problem there is a wide range of high-quality solutions, so a more aggressive diversification is warranted. We thus set $\lambda = 0.02$

Initialization: α and γ

These first two parameters have the least impact on the tabu search since these parameters are dynamically adjusted based on route duration and time window violations respectively. As discussed in Section 4.3.3 high values of α or γ constraints the search to look for solutions that do not violate any constraints and low α and γ values allow the search to take on solutions that violate these constraints by a larger extent. A range of α and γ was experimented upon with α and γ having a range of 0.1 to 10. If the δ is tuned well, these parameters will reach their desired values in a couple of iterations. Because in the first couple of iterations, the researcher prefers the tabu search to accept solutions that do not violate any constraints, the following initialization parameters are set to $\alpha = 5$ and $\gamma = 5$.

Violation adjustment: δ

This parameter is an important parameter since it influences α and γ and thus the aggressiveness of looking for solutions that violate the route duration and time window constraints. Having a too small value results in having almost no diversification, since the search cannot take on solutions that would violate these constraints. Contrarily, a too high value results in the search taking on extremely infeasible solutions. Cordeau settles on a value of 0.5 to achieve this balance between highly infeasible solutions and diversification. In the solution methods tabu search this balance also seems to show when using this value. An interesting revaluation for the value of δ is that it follows a kind of three phase pattern. In the first phase, α and γ increase significantly, since the initial solution almost always violates the constraints. In the second phase, α and γ stabilise in a certain range, with α and γ increasing or decreasing in value with a random probability. In the final phase, where the improvements to the best solution are minimal α and γ almost exclusively decrease. So we set $\delta = 0.5$.

Tabu list length: θ

The tabu list length parameter θ, determines the maximum number of moves within the tabu list at any given time. The best value of θ increases with problem size, since in large instances there are more possibilities to cycle back to a local optimum than in smaller instances. Cordeau recommends a value of θ of $[7.5 * \log_{10} n]$, with n being the number of customers to account for this characteristic of increasing problem size. However, in our methodology, a list length is used instead of setting a move as tabu for the θ number of iterations. Since, with swapping customers, there is a chance that multiple attributes get added after an iteration the tabu list length needs to be adjusted to compensate. A tabu attribute gets set to non-tabu quicker if the list length stays the same, but multiple attributes can be added in one iteration. After numerous observations, a θ of $[10 * \log_{10} n]$ seems sufficient to prevent cycling. So we set $\theta = [10 * \log_{10} n]$.

5.3.3 Tabu search runtime

The proposed solution methodology and Cordeau et al. (2001) tabu search diverge on the stopping criterion of the TS. In our solution methodology, the tabu search is terminated after a certain amount of runtime. This parameter is the most crucial parameter since it influences the optimally of the result the most. A low runtime can be effective in finding a decent solution in a short time but has a high chance of settling on a solution that is low quality. A high runtime, however, can cause the tabu search to needlessly waste computation time, to find very minor improvements in the solution or not to find a better solution at all. Thus, selecting an appropriate runtime for the tabu search that balances solution quality against runtime is paramount.

To find the optimal runtime the following series of experiments ($R1$ - $R3$) have been conducted. The experiments range from allocating 10 minutes, 30 minutes and 120 minutes of runtime to the tabu search using the parameters set in Section 5.3.2. The experiments have been conducted on the artificial data instances $A1 - A10$ outlined in Table 5.2. As discussed in Section 5.3.1 the customer-drop-off based visit combination selection construction heuristic has been used for creating the initial solutions to these data instances. The improvement, compared to the initial objective has been reported across all data instances, as well as the initial values are reported in Table 5.5.

ID	**Initial result**	*R1*			*R2*			*R3*		
		Result	**Time**	**Δ%**	**Result**	**Time**	**Δ%**	**Result**	**Time**	**Δ%**
A1	36,985	**15,955**	600	56,86%	**15,955**	1800	56,86%	**15,955**	7200	56,86%
A2	41,672	**33,525**	600	19,55%	**33,525**	1800	19,55%	**33,525**	7200	19,55%
A3	64,644	**51,708**	600	20,01%	**51,708**	1800	20,01%	**51,708**	7200	20,01%
A4	91,756	62,281	600	32,12%	62,160	1800	32,26%	**61,994**	7200	32,44%
A5	191,671	96,383	600	49,71%	94,605	1800	50,64%	**93,468**	7200	51,24%
A6	96,781	79,261	600	18,10%	79,261	1800	18,10%	**79,261**	7200	18,10%
A7	127,636	93,376	600	26,84%	91,343	1800	28,43%	**91,249**	7200	28,51%
A8	199,159	162,615	600	18,35%	124,724	1800	37,37%	**123,483**	7200	38,00%
A9	189,935	131,468	600	30,78%	125,795	1800	33,77%	**125,655**	7200	33,84%
A10	224,093	161,065	600	28,13%	155,213	1800	30,74%	**153,980**	7200	31,29%
Avg.	126,433	88,764	600	30,05%	83,429	1800	32,77%	**83,028**	7200	32,98%

Table 5.5: *Summary of experiments $R1 - R3$. The best objective for each instance is given in bold.*

Table 5.5 shows the outcome of experiments $R1 - R3$. The objective, computation time and improvement for each experiment is provided. The improvement is based on the difference between the objective value and the initial solution value, which is also provided. The tabu search improves the initial solution by 18% to 57% depending on the data instance solved and solve time. In general the longer the computation time the better the improvement.

This improvement in solution quality comes at a cost of increased computation time. The $R2$ instance of allowing for a computation time of up to 30 minutes seems to strike the best balance between solution quality and computation time. On average the solution gets improved by 32.77% in the $R2$ experiments against the on average 32.98% improvement of the $R3$ experiments, whilst having four times as much computation time allocated. The $R1$ experiments also show promising results with an average 30,05% improvement demonstrating that tabu search can quickly improve from the initial solution. However, since $R2$ strikes an acceptable balance between solution quality and computation time, we use a computation time of 1800 seconds for the tabu search.

5.3.4 MILP runtime

To find the optimal runtime for the MILP a balance between solution quality and computation time is made. The following three experiments are created to find the best balance between computation time and solution quality. We define the following three experiments $L1$, $L2$ and $L3$, which allow us to test for varying computation times, from 10 minutes to 120 minutes.

- *Experiment L1* allows for a short computation time of 10 minutes.
- *Experiment L2* allows for a moderate computation time of 30 minutes.
- *Experiment L3* allows for a long computation time of 120 minutes.

ID	*L1*			*L2*			*L3*		
	Result	**Time**	**Gap**	**Result**	**Time**	**Gap**	**Result**	**Time**	**Gap**
A1	**15,955**	0,3	0,00%	**15,955**	0,3	0,00%	**15,955**	0,3	0,00%
A2	**33,525**	600	0,68%	**33,525**	1800	0,42%	**33,525**	3495	0,00%
A3	**51,708**	57,3	0,00%	**51,708**	57,3	0,00%	**51,708**	57,3	0,00%
A4	62,301	600	2,88%	62,301	1800	2,77%	**61,900**	7200	1,84%
A5	92,786	600	4,75%	91,985	1800	3,88%	**91,315**	7200	3,09%
A6	79,273	600	3,09%	79,058	1800	2,57%	**79,017**	7200	2,47%
A7	175,266	600	48,90%	92,011	1800	2,64%	**91,715**	7200	2,32%
A8	139,085	600	13,50%	124,878	1800	3,34%	**123,470**	7200	2,24%
A9	135,154	600	12,60%	126,410	1800	5,76%	**125,654**	7200	4,90%
A10	191,265	600	29,70%	155,839	1800	5,72%	**154,299**	7200	4,68%
Average	97,632	485,76	11,61%	83,367	1446	2,71%	**82,856**	5395	2,15%

Table 5.6: *Summary of experiments $L1 - L3$. The best objective for each instance is given in bold.*

Table 5.6 shows the outcome of experiments $L1 - L3$. The objective, computation time and optimality gap for each experiment are provided. For instance, $D1$ and $D3$ optimality is achieved across all experiments, terminating the MILP optimization before its allocated computation time and thus displaying a lower value in Table 5.6 than the allocated runtime. for instance, $D2$ no optimality gap is shown at the extended computation time. For data instances $D4 - D10$, the optimality gap consistently decreases with an increasing amount of computation time allocated. Especially for larger instances, the optimality gap decreases significantly with an increasing amount of computation time between experiments $R1$ and $R2$. There is still an expected decrease in the optimality gap between experiments $R2$ and $R3$, however, the decrease is significantly lower.

The average best solutions are found in the instances where a prolonged computation time is allocated. However, experiment $R2$ shows a better balance between solution quality and computation time. When comparing the optimality gap between experiments $R2$ and $R3$, a decrease of 0,56% is observed against a quadrupling of computation time. A computation time of 600 seconds is not preferable, since the optimality gap especially in larger instances is large. Given the average objective of 83,3671 and an optimality gap of 2,71%, a computation time of 1800 seconds is the most reasonable option for solving the MILP.

5.4 Evaluation of real-world scenarios

This section evaluates the real-world scenarios described in Section 5.2.2 using the MILP and tabu search. The parameters for both these solution methodologies have been discussed in the previous Section 5.3 and all experiments are consequently done using these parameters. The results for both the MILP and tabu search will be compared, to determine the strengths and weaknesses of both solution methodologies. Furthermore, for each instance, the best result from the two solution methodologies is compared against the real-world KPIs to determine to which extent the proposed solution methodologies are superior to the planning of the inspectors. Lastly, trends and patterns in regard to the solution methodologies solution are presented. The trends and patterns explain the difference in solution quality between inspectors compared to the computer made planning and identify characteristics of an efficient monthly planning.

Note: values are multiplied by an arbitrary number for confidentiality.

5.4.1 MILP and tabu search solutions

The experiments of phase 2 as outlined in Section 5.1 assess the inspector's planning against the planning made from the 2 solution methodologies. The data instances are the real-world Eurofins data, where each instance represents a monthly list of orders that need to be fulfilled by the end of the month. The MILP and tabu search are performed on the three instances described in Section 5.2.2 and the results are presented in Table 5.7 and Table 5.8 respectively. For the parameters used in the following experiments please see Section 5.3.

The results for the MILP for the three instances are presented in Table 5.7. For each instance, the data instance ID, nodes, dropoff configuration (Table 5.1) and number of days are given. In subsequent columns, the model costs are represented in the Objective column followed by the optimality gap given as a percentage. Next, the computation time is provided. In the last columns, insights into solutions are displayed by displaying the travel time in hours, the distance travelled in kilometres and providing how many days each depot is used.

The results for the tabu search are presented in Table 5.8. Since the tabu search uses an initial solution as a starting point to improve from the initial solution is also reported. Another change from the MILP is that the tabu search reports the improvement compared to the initial solution as opposed to reporting an optimality gap. The other reported values are similar to those of the MILP.

The MILP and tabu search are applied to 3 real-world data instances, as discussed in Section 5.2.2. The aim is to make a comparison between the planning quality of the inspector against the solution quality of the MILP and tabu search. Based on the outcomes a small analysis is provided by discussing trends and patterns in the tabu search algorithm. The MILP and tabu search use the parameters described in Section 5.3.

5.4.2 Results of real-world instances

The tabu search and MILP are applied to the real-world scenario data instances $I1 - I3$. The results for the tabu search are provided in Table 5.8 and the results for the MILP are presented in Table 5.7.

The MILP results on data instances $I1 - I3$ are detailed in Table 5.7. For each instance, the data instance, number of nodes, the dropoff configuration and number of data are provided. The Following columns provide the route time, and the total amount of time it takes to execute the monthly workload. The optimality gap and computation time are also provided. The other columns provide insights regarding the solution such as the travel time, distance travelled, number of times dropoff 1 is visited and number of times dropoff 2 is visited.

Instance	*Nodes*	*Dropoff ID*	*Days*	*Gap*	*Time (s)*	*Route Time (h)*	*Travel Time (h)*	*Distance (km)*	*D1*	*D2*
I1	103	C2	20	5.81%	1800	213,32	45,74	1804,11	20	-
I2	96	C1	19	2,28%	1800	214,70	32,50	1302,96	19	-
I3	90	C3	20	10,80%	1800	220,66	58,11	2999,91	7	13
Average	-	-	-	6,54%	1800	216,23	45,44	2032,59	-	-

Table 5.7: *Overview of the MILP results on real-world instances $I1 - I3$. (values are multiplied by an arbitrary number for confidentiality)*

The MILP failed to identify the optimal solutions as expected and an optimality gap of 2.28% to 10.80% is present. This reinforces the notion of the MILP's computational limitations in achieving optimality in larger problem size. Especially in case $I3$ the MILP has problems identifying the optimal solution, which likely is because this is the most complex data instance due to its dropoff configuration. The average route time calculated by the MILP is 216,23 hours.

Instance	*Nodes*	*Dropoff ID*	*Days*	*Initial result*	*Improvement*	*Time (s)*	*Route Time (h)*	*Travel Time (h)*	*Distance (km)*	*D1*	*D2*
I1	103	C2	20	295,86	29,06%	1800	209,86	42,28	1666,59	20	-
I2	96	C1	19	249,66	12,80%	1800	217,71	32,98	1322,53	19	-
I3	90	C3	20	269,44	21,84%	1800	210,61	48,06	2472,03	7	13
Average	-	-	-	271,66	21,23%	1800	212,53	41,12	1821,05	-	-

Table 5.8: *Overview of the TS results on real-world instances $I1-I3$. (values are multiplied by an arbitrary number for confidentiality)*

The tabu search outcomes are shown in Table 5.8. For each instance, the same information as the MILP is displayed except for the optimality gap, which is replaced by the initial solution created by the construction heuristic and improvement by the TS expressed as a percentage.

The tabu search shows varying results for each instance. The tabu search has an average total route time of 212,53 hours with an average improvement of 21,23% demonstrating its capability to improve solutions substantially from an initial solution. In the $I3$ instance the tabu search opts to visit the second dropoff locations 13 times and the first location 7 times demonstrating that having 2 dropoff locations can improve routing times.

Solution method analysis

Tabu search shows in almost all instances a significant improvement from the initial solution. In contrast, the MILP was not able to solve the large-scale instances within the 1800 second computation time. The TS was able to improve from the initial solution by an average of 21,23% while maintaining the same runtime. With equal runtimes, the tabu search beat the MILP by on average of 3,70 hours, which is a substantial improvement. The tabu search outperforms the MILP in instances $I1$ and $I3$ and was outperformed in instance $I2$. Considering that $I1$ and $I3$, were the more complex instances where the customer nodes are spread throughout a large area as opposed to instance $I2$, where the customers are clustered in a small area. The tabu search is likely to be more effective in these complex cases, while the MILP is effective in the more basic data instances. The MILP is likely outperformed by the Tabu Search in instances $I1$ and $I3$, because it was unable to find the optimal solution. It is unknown if the TS finds the optimal solution since the lower bound of the solution space is unknown. This indicates that the tabu search ability to find near-optimal solutions in large scale scenarios.

Another noteworthy observation is that the tabu search tends to converge faster to a near-optimal solution compared to the MILP. A larger sample, size is necessary to quantify this behaviour. With the tabu search being set to 1800 seconds this is an inflexible parameter, using a different termination criterion could result in reduced runtimes by simply terminating when an improved solution is not found after an arbitrary amount of time for example. Overall, the values from the TS and MILP are relatively close to each other. There are explanations for this behaviour. Firstly, the MILP resulted in better solutions than the tabu search. By changing certain Gurobi parameters, as well as the formulation of constraints, the MILP finds better solutions in less time. Especially, using an efficient mathematical formulation of the subtour elimination presented in Chapter 4.2.5 decreased runtimes significantly. On the other hand, the tabu search is able to find on average better solutions than the MILP, but still has significant room for improvement. Especially, the insertion mechanism and subsequent solution evaluation are expensive computationally, which significantly increases the time of one tabu cycle. Other optimizations in regard to the programming of the TS can significantly decrease computation time. In conclusion, the TS marginally outperforms the MILP solution approach, however, the TS computation is likely to decrease significantly by using more efficient programming and calculation approaches.

Comparison total travel time

The inspector's planning is compared against the outcomes of the MILP and tabu search solution approaches. By comparing the travel time of each inspector against the travel time outcomes in the MILP and TS conclusions are drawn. Figure 5.1 displays a graph comparing these solution methods and the planning of the inspectors.

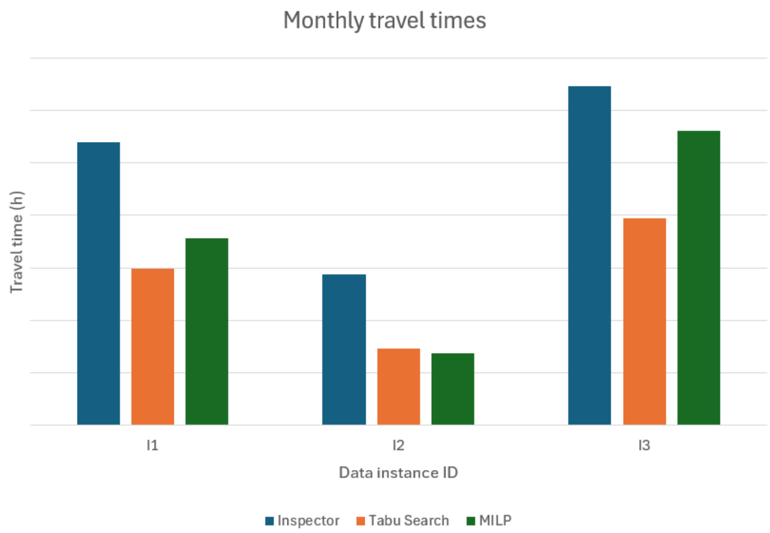

Figure 5.1: *Graph showing the travel times for each data instance and planning method. (values hidden for confidentiality)*

The monthly time spent on travelling for the MILP, TS and the actual inspector planning are plotted in Figure 5.1. The blue bar represents the actual planning of the inspectors, the orange bar represents the tabu search results, and the green the MILP results. It is clear that TS and MILP consistently outperform the inspector-made planning. The figures suggest a potential travel time reduction of 8% to 25% if Eurofins starts optimizing the inspector planning. The scenario however an optimistic estimation as the scenarios represent the historic planning of one month for an inspector. Throughout the month the inspectors got extra client inspections planned, which the inspectors could not anticipate. The tabu search results indicate a consistent time saving of around 22% compared to the inspector-made planning. If we take the best result from the TS and MILP the average travel time gets reduced by 23,86% Both the MILP and TS solution approaches indicate improvements over the current planning methodology. The MILP shows on average lesser results compared to the tabu search for the same computation time, indicating that TS outperforms the MILP in higher-complexity instances. However, it is difficult to make claims about whether TS is superior to MILP since we use 2 different measures of performance, optimality gap and improvement in Section 6.3 we discuss this further. In similar-sized instances, TS is thus recommended compared to the MILP, which can be too computationally time-consuming to run if more complexity is added.

5.4.3 Trends and patterns

This section discusses the trends and patterns observed in the solutions provided by specifically tabu search. The solutions to instances $I1 - I3$ are provided in Appendix B in Figures B.1, B.2, B.3.

The routes constructed by the tabu search seem to follow a cluster pattern, where the TS clusters certain customers together to visit in one day. This pattern is clear when looking at Figure, where the TS proposes to visit all customers in Groningen (North), in one day (Day 16 Route). This is in contrast to the inspector, that comes to these locations on three different days. Furthermore, the tabu search prefers to have high variability in daily routing times, opting to have as long as possible route each day and compensating it with a shorter route on the following day. This is likely because the tabu search looks for possibilities to visit all customers in one area even if the route takes longer than average. An extended allowed maximum route time per day can thus decrease the total travel time. This behaviour can best be observed in Figure B.2. There are clear outliers in this work area and the tabu search creates a route denoted in grey, to visit all the outliers in 2 days. All the outliers could not be visited in one day due to the maximum daily route time. The final observation is that the routes created by TS rarely follow an erratic pattern. It prevents creating routes that move back and forth in areas but makes routes that almost make a continuous line from the homes through the customers to the depot. This characteristic is not present in the planning made by the inspectors where usually they visit one area and there is limited reasoning behind the order of visits.

5.5 Conclusion

This chapter discusses the data instances, parameter tuning process and evaluation of the MILP and TS using various experiments. This chapter aimed to compare the proposed solution approaches against the current Eurofins situation.

This chapter first discussed the setup for the experiments followed by the creation of data sets for parameter tuning and comparison to the real-world instances. There were 2 data sets created one artificial dataset, whose instances were based on the Eurofins data. The artificial instances varied in size by each data instance having a varied number of nodes, dropoff configuration, days in the planning horizon and average speed. Additional to the artificial data set a real-world data set was created. The instances of the real-world dataset were created based on the monthly planning of an inspector and were created to compare the results of the TS and MILP against the actual KPIs.

The experiments were conducted in 2 phases. The first phase was to tune the parameters of the proposed solution methodologies and the second phase was the actual comparison against real-world scenarios. The first phase focussed on tuning the parameters of the MILP by first determining a maximum computation time. The parameters of the tabu search were consequently tuned, as well as the maximum computation time. The customer-dropoff-based visit combination selection was chosen as the construction heuristic for the tabu search due to its consistency in producing initial routes. The computation times for both the MILP were set to 1800 seconds because it has a great balance between computation time and solution quality. Similarly, the computation time for the tabu search was also set at 1800 seconds.

In the second phase, the TS and MILP were applied to the real-world data instances. Using the parameters set in Phase 1 the Tabu Search outperforms the MILP due to the complexity of the real-world instances. The MILP is effective in finding optimal solutions, especially in small-scale instances ($\leq$ 40 nodes), however, it has difficulty achieving the optimal solution In large-scale instances. The TS significantly decreases the total travel time of Bureau de Wit's inspectors by recommending optimized routes by having a more systematic approach to planning and being able to consider the whole planning horizon. The TS shows an average decrease in travel time of 25% compared to the current Eurofins situation while performing the same services.

These results provide an indication to Eurofins about the efficiency of inspector planning at Bureau de Wit. It is clear that using vehicle routing to plan routes for inspection is critical to minimizing travel times in order to allocate more time to the inspections. The analysis shows that using prolonged routes, clustering and using multiple depots can significantly decrease travel times. Because the problem size is similar for almost every inspector the TS is recommended, since it outperforms the MILP in the complex inspector routing planning scenarios. For relatively simple inspector planning problems the MILP is recommended.

Chapter 6

Conclusions, recommendations and discussion

This chapter provides concluding remarks in regards to this thesis by providing final conclusions, recommendations and insights based on the findings presented in the thesis from addressing the Eurofins transportation problem. Thus in this chapter, the last research question is discussed namely, *What conclusions and recommendations can be made from executing the thesis at Eurofins?*.

This chapter begins by providing final conclusions followed by a discussion that outlines the limitations of this research as well as the theoretical and practical contributions of this work. Avenues for future research are discussed aswell. Finally, we provide some recommendations for Eurofins.

6.1 Conclusions

This thesis set out to address the knowledge gap for Eurofins to test the impact of having dedicated planning tools on their operations. The main goal was to find a solution method to effectively plan an inspector's monthly workload in the shortest time frame, and to increase the planning effectiveness by executing the same workload in a shorter time span. Through reviewing existing literature to create a foundation for the solution methodology a periodic vehicle routing problem with time windows and dropoffs (PVRPTWD) was formulated. 2 solution methodologies for this mathematical formulation were developed, a Mixed Integer Linear Program (MILP) and a tabu search (TS). The solution methods were customised to fit the characteristics of the Bureau de Wit's planning and routing problem, such as time windows, visit frequencies, route duration and customer availability. To tune the MILP and the tabu search for applicability to the Eurofins problem, 10 artificial data instances were created based on the Eurofins data. The solution methodologies were tuned with the goal of handling scenarios with varying amounts of nodes and days to ensure the models work in a wide range of scenarios. After the tuning process, the two solution methods were applied to real-world scenarios provided by Eurofins. After extensive experimentation, the tabu search proved to be superior in large-scale scenarios and in improving the inspectors' planning efficiency. The tabu search produces plannings that reduce the travel time by 25% compared to the current inspector-made planning. The experiments demonstrated that routing based on the most constrained points creates effective planning. The MILP as the tabu search has extended computation times, however, the tabu search is preferable due to it discovering high-quality solutions quicker than the MILP.

6.2 Discussion

This section discusses the theoretical and practical contributions of this work. Lastly, this section outlines the limitations of this work.

6.2.1 Theoretical contribution

This research conducted a literature study on VRP variants, however existing formulations that incorporate nodes that must be visited at the end of a period have not been found. Numerous studies have been conducted on the PVRP and its variants Campbell and Wilson (2014), however, the aforementioned characteristic, which is almost similar to an open vehicle routing problem has not been found. This created a need to develop a novel formulation by adapting the formulation presented by Ahmadi Basir et al. (2024) and adding the in this research defined dropoff constraint.

The contribution lies in adapting the well-known Tabu Search methodology presented in Cordeau et al. (2001) to this specific variation of the PVRPTW. This research also uses a novel construction heuristic by using the concept presented in Chao et al. (1995) to spread customers across the planning horizon and exploit that with the PVRPTWD characteristics of the depot locations. Through assigning customers to dropoff locations and subsequently assigning certain days to dropoff locations and only then assigning visit combinations to customers based on spreading the service times a new construction is created. This customer-dropoff location-based visit combination selecting mechanism outperforms the randomised construction heuristic based on the Cordeau et al. (2001) initialization heuristic.

Finally, this research contributes by providing a case study of using tabu search as an adaptable and effective metaheuristic for a complex practical problem. Additionally, This case study provides evidence that using tabu search in logistics, particularly for problems with a scheduling component is effective in large-scale instances compared to traditional exact methods.

6.2.2 Practical contribution

This research is performed at Eurofins Food, Feed and Water, superficially at Bureau de Wit the hygiene inspection division of Eurofins. The practical contribution for Eurofins is the insights into the planning quality of their inspectors. Using the insights of the model and current KPIs of the inspectors an opportunity to improve routing efficiency is identified. The tabu search algorithm demonstrated that travel times can be reduced by 25% encouraging an adoption of software of similar capabilities into their operations.

The algorithm also demonstrated how tactical and strategic level decisions impact operational performance. The impact of selecting geographically efficient dropoff locations as well as the impact of adding new customers and agreements with customers in regard to visit frequency and time windows. This effect also works vice versa, where these operational insights can be used for tactical and strategic level decisions when it comes to customer selection and dropoff location selection. Additionally, the use of tabu search and by extend other metaheuristics algorithms as a scalable solution for vehicle routing and scheduling problems can serve as helpful tools for companies experiencing increasing amounts of complexity in their logistics operations.

6.2.3 Limitations and future research

To limit the complexity of this research a scope and assumptions were defined. The research thus has multiple limitations, but also provide avenues for future research. The following limitations are defined:

- This research has taken the monthly planning from one inspector as its scope as described in Chapters 1 and 2. The full-scale problem of having multiple geographically dispersed inspectors and dropoff locations throughout the Netherlands is too large of a scope considering a planning horizon of one month. Future research could be conducted to address this full scale problem or can be used to create a strategic level model that determines dropoff locations. The working areas of each inspector could be investigated using this methodology as well.
- This research is performed from a operational logistics perspective. However, tactical and other strategic level decisions are assumed to be fixed. Future research is needed to investigate tactical and strategic level decisions on the efficiency of the inspector planning, such as

flexibility of working hours, (electric) vehicle selection, customers specific geographic pricing and geographically based hiring policies.

- The customer visits are determined to be deterministic, however, this ignores the dynamic behaviour of customer requests throughout the month. The dynamism is mainly due to new customers being added throughout the month and re-inspections. However, the lack of data on these occurrences limits our ability to investigate their effect on planning efficiency.
- For the parameter tuning of both the MILP and tabu search the parameters could have been more extensively tuned. Since the adjustment of parameters for the tabu search is a subjective process, the parameters could have been tuned better through the use of statistics. Another limitation of tabu search is that is unknown if the solution space is sufficiently explored, because unlike the MILP there is no known best objective bound.
- The limited sample size of only three monthly plannings provides limited statistical evidence of the true efficacy of the proposed solution methods. Future research could increase the sample size by using the plannings of more inspectors and also take plannings from different months and assessing how seasonal demand influences optimized planning.
- For the mathematical formulation the following assumptions were crucial including the distances, vehicle speeds, service times and time windows. Future models can use real-world data to determine travel times. Additionally, opening times were taken as time windows, but are not always equal to when a customer can or wants to be visited. These characteristics that add more complexity to the model, but create a more accurate representation are characteristics to include in future research.

6.3 Recommendations

Based on the outcomes of this research, it is strongly advised that Eurofins use software to plan their inspectors. Additionally, the company should encourage flexible working hours for their inspectors, especially in working areas with highly geographically dispersed customers. Finally, to have better communication with sales and logistics a logistics check should be performed to see what the impact of inserting a customer into the planning is before making agreements. In case Eurofins wants to extend the TS heuristic efforts can be made to increase the efficiency of the algorithm decreasing computation times. Another direction for the TS algorithm is to extend the model to incorporate more complex constraints and add other characteristics to the model. The main focus for improving the solution approaches should be to focus on increasing the data quality. Firstly, the service times are now approximations, however, using historical data from different sites and previous visits a more accurate service time estimation can be made.

The development or integration of commercial routing software is recommended for daily operations. A well-designed software allows the company to incorporate real-time changes, such as unexpected sick leave, customer requests and unforeseen traffic circumstances into their daily decision-making. The greatest reduction in route execution time can be achieved through integrating the tabu search or other effective algorithms to create routes for the inspectors. For problems with a low amount of nodes, less than 40 the MILP can also be considered as an effective tool to integrate.

Furthermore, we recommend for future research Eurofins focuses on the other aspects of the inspectors' planning. First, there is the problem of selecting dropoff locations, so analysis of the addition and removal of certain dropoff locations can increase the efficiency of the network by decreasing non-productive kilometres compared to the cost of operating a dropoff location. Secondly, we recommend Eurofins investigate the possibility of dynamic work areas, where instead of assigning all customers within one area to one inspector, we assign customers based on an optimization model, where a customer can be assigned to multiple inspectors based on costs.

Bibliography

S. Ahmadi Basir, G. Şahin, and G. Özbaygın. A comparative study of alternative formulations for the periodic vehicle routing problem. *Computers & Operations Research*, 165:106583, 5 2024. ISSN 0305-0548. doi: 10.1016/J.COR.2024.106583.

W. Berens. The suitability of the weighted/p-norm in estimating actual road distances. *European Journal of Operational Research*, 34:39–43, 1988.

B. Boyacı, T. H. Dang, and A. N. Letchford. Vehicle routing on road networks: How good is Euclidean approximation? *Computers & Operations Research*, 129:105197, 5 2021. ISSN 0305-0548. doi: 10.1016/J.COR.2020.105197.

K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 9 2016. ISSN 0360-8352. doi: 10.1016/J.CIE.2015.12.007.

A. M. Campbell and J. H. Wilson. Forty years of periodic vehicle routing. *Networks*, 63(1):2–15, 2014. doi: 10.1002/net.21527. URL `http://dx.doi.org/10.1002/net.21527`.

I. Chao, B. L. Golden, and E. Wasil. An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):25–44, 8 1995. ISSN 0028-3045. doi: 10.1002/net.3230260104.

P. Chapman. CRISP-DM 1.0: Step-by-step data mining guide. In *CRISP-DM 1.0: Step-by-step data mining guide*, 2000. URL `https://api.semanticscholar.org/CorpusID:59777418`.

G. Clarke and J. W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581, 1964. ISSN 0030364X, 15265463. URL `http://www.jstor.org/stable/167703`.

J.-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 9 1997. ISSN 0028-3045. doi: 10.1002/(SICI)1097-0037(199709)30:2¡105::AID-NET5¿3.0.CO;2-G.

J.-F. Cordeau, G. Laporte, and A. Mercier. A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *The Journal of the Operational Research Society*, 52(8):928–936, 2001. ISSN 01605682, 14769360. URL `http://www.jstor.org.ezproxy2.utwente.nl/stable/822953`.

G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem. *Management Science*, 6(1): 80–91, 10 1959. ISSN 0025-1909. doi: 10.1287/mnsc.6.1.80.

B. Eksioglu, A. V. Vural, and A. Reisman. The vehicle routing problem: A taxonomic review, 11 2009. ISSN 03608352.

Eurofins Food Feed Water Benelux. About us, 2024. URL `https://www.eurofinsfoodtesting.com/about-us`.

M. Gendreau, P. Jean-Yvesand, O. Bräumlaysy, G. Hasle, and A. Løkketangen. Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. In S. Golden Bruce
}and Raghavan and W. Edward, editors, *The Vehicle Routing Problem: Latest Advances and*

New Challenges, pages 143–169. Springer US, Boston, MA, 2008. ISBN 978-0-387-77778-8. doi: 10.1007/978-0-387-77778-8-7.

S. F. Ghannadpour, S. Noori, R. Tavakkoli-Moghaddam, and K. Ghoseiri. A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing*, 14(PART C):504–527, 1 2014. ISSN 1568-4946. doi: 10.1016/J.ASOC.2013.08.015.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1 1986. ISSN 0305-0548. doi: 10.1016/0305-0548(86)90048-1.

P. Hansen, N. Mladenović, and J. A. Moreno Pérez. Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010. ISSN 15729338. doi: 10.1007/s10479-009-0657-6.

H. Heerkens and A. van Winden. *Solving Managerial Problems Systematically.* Noordhoff Uitgebers, 2017.

M. E. Hesam Sadati, B. Çatay, and D. Aksen. An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems. *Computers & Operations Research*, 133:105269, 9 2021. ISSN 03050548. doi: 10.1016/j.cor.2021.105269.

N. Hotz. What is CRISP DM?, 3 2024. URL https://www.datascience-pm.com/crisp-dm-2/.

E. Lalla-Ruiz, C. Expósito-Izquierdo, S. Taheripour, and S. Voß. An improved formulation for the multi-depot open vehicle routing problem. *OR Spectrum*, 38(1):175–187, 1 2016. ISSN 14366304. doi: 10.1007/s00291-015-0408-9.

G. Laporte. Fifty Years of Vehicle Routing. *Transportation Science*, 43(4):408–416, 11 2009. ISSN 0041-1655. doi: 10.1287/trsc.1090.0301.

J. Lenstra and A. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2): 221–227, 1981. doi: 10.1002/net.3230110211.

F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, and M. Yuan. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. 3 2023. URL http://arxiv.org/abs/2303.04147.

Y. Liu. An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research*, 111:1–20, 11 2019. ISSN 03050548. doi: 10.1016/j.cor.2019.05.024.

C. E. Miller, A. W. Tucker, and R. A. Zemlin˜. Integer Programming Formulation of Traveling Salesman Problems*. *Journal of the ACM (JACM)*, 7(4):326–330, 1960.

H. Mintzberg. Structure in 5's: A Synthesis of the Research on Organization Design. *Management Science*, 26(3):322–341, 3 1980. ISSN 0025-1909. doi: 10.1287/mnsc.26.3.322.

J. R. Montoya-Torres, J. López Franco, S. Nieto Isaza, H. Felizzola Jiménez, and N. Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers and Industrial Engineering*, 79:115–129, 2015. ISSN 03608352. doi: 10.1016/j.cie.2014.10.029.

P. Munari, T. Dollevoet, and R. Spliet. A generalized formulation for vehicle routing problems. 6 2016.

Q. Ni and Y. Tang. A Bibliometric Visualized Analysis and Classification of Vehicle Routing Problem Research. *Sustainability*, 15(9):7394, 4 2023. ISSN 2071-1050. doi: 10.3390/su15097394.

A.-K. Rothenbächer. Branch-and-Price-and-Cut for the Periodic Vehicle Routing Problem with Flexible Schedule Structures. *Transportation Science*, 53(3):850–866, 5 2019. ISSN 0041-1655. doi: 10.1287/trsc.2018.0855.

C. Schröer, F. Kruse, and J. M. Gómez. A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181:526–534, 2021. ISSN 18770509. doi: 10.1016/j.procs.2021.01.199.

B. S. Vieira, G. M. Ribeiro, and L. Bahiense. Metaheuristics with variable diversity control and neighborhood search for the Heterogeneous Site-Dependent Multi-depot Multi-trip Periodic Vehicle Routing Problem. *Computers & Operations Research*, 153:106189, 5 2023. ISSN 0305-0548. doi: 10.1016/J.COR.2023.106189.

B. Yu and Z. Z. Yang. An ant colony optimization model: The period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 47 (2):166–181, 3 2011. ISSN 13665545. doi: 10.1016/j.tre.2010.09.010.

J. Zhang, H. Huang, and X. Wang. Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 64:23–42, 4 2016. ISSN 1084-8045. doi: 10.1016/J.JNCA.2015.12.018.

Appendices

Appendix A

Algorithms

Algorithm 5 Basic tabu search heuristic

Initial solution s: $s \leftarrow InitialSolution(x)$
Tabu list: $T \leftarrow \emptyset$
while $k \leq k_{terminate}$ **do** $s' \leftarrow BestSolution(N\,(s)$
    if $s' \notin T$ **then**
        $S \leftarrow S'$
        Update T
    else if $s' \in T$ & $AspirtationCriteriaSatisfied$ **then**
        $S \leftarrow S'$
        Update T
    else
        Decrease T
    end if
end while
return $s*$ ▷ point with best evaluation so far

Algorithm 6 Basic VNS algorithm

Neighbourhood: k
current solution: x
while $t \leq t_{max}$ **do**
    $k \leftarrow 1$
    while $k \leq k_{max}$ **do**
        $k \leftarrow k + 1$
        $x' \leftarrow Shake(x, k)$
        $x'' \leftarrow LocalSearch(x')$
        $Neighbourhoodchange(x, x'', l)$ ▷ Change neighbourhood
    end while
end while

Algorithm 7 Greedy insertion

Customers: $C = \{1, ..n\}$
Route: $R \leftarrow \emptyset$
InsertionList: $L \subseteq C$

while $L \neq \emptyset$ **do**
  CheapestInsertion(L, R) ▷ Find cheapest insertion in list L
  Insert(R, i)
  $L \leftarrow L \setminus i$
end while
return R

Appendix B

Tabu search routes

B.1 Tabu search results I1

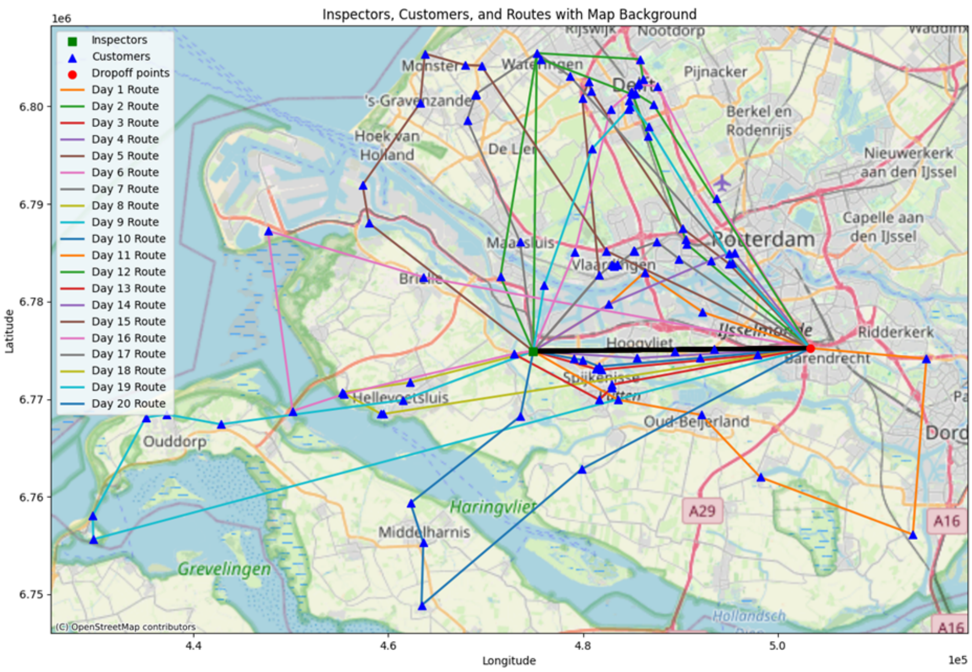

Figure B.1: *Tabu search routes for data instance I1. Background from OpenStreetMap*

B.2 Tabu search results I2

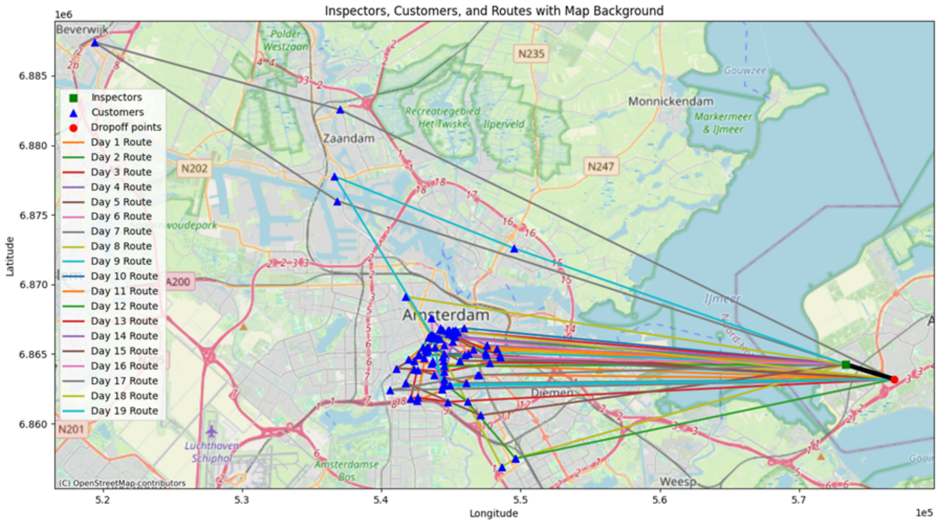

Figure B.2: *Tabu search routes for data instance I2. Background from OpenStreetMap*

B.3 Tabu search results I2

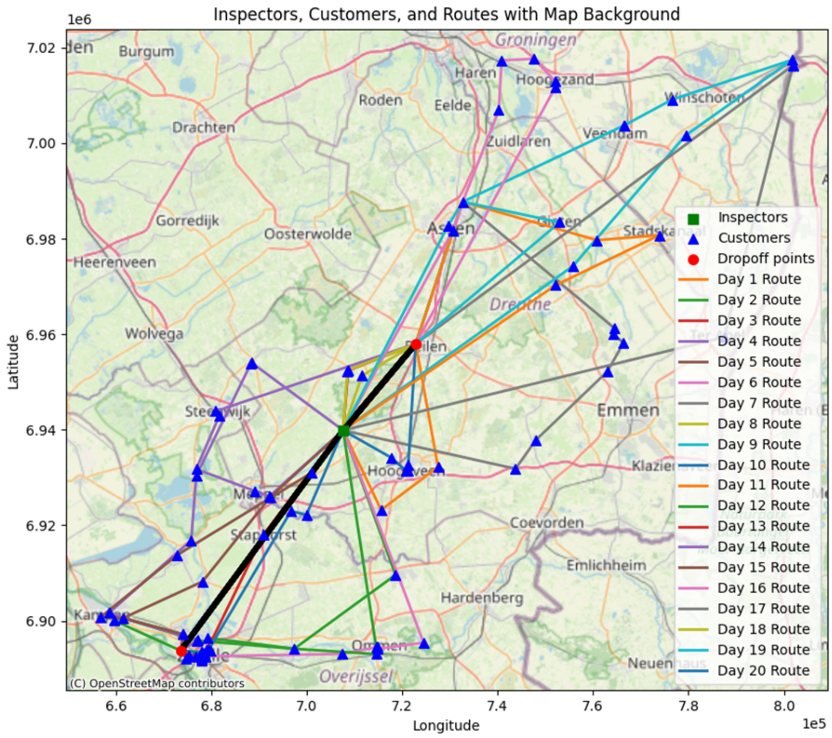

Figure B.3: *Tabu search routes for data instance I3. Background from OpenStreetMap*

Appendix C

Solution example

Figures C.1 and C.2 provide a schematic representation of a visit schedule swap. In this example the schedule of customer 7 gets swapped where it needs to be visited on day 2 instead of day 3. So $p = \{3\}$ and $p' = \{2\}$. The consequent attributes that get changed are thus that attribute $(7, 3)$ gets removed from the solution $B(s)$ and $(7, 2)$ gets added, since customer 7 does not get visited on day 3 anymore, but gets visited on day 2 instead. The other attributes in $B(s)$ stay the same. The full encoding of the solution $B(s)$ before the swap is thus:

$$B(s) = \begin{Bmatrix} (1,3),\ (2,3),\ (3,3),\ (4,1),\ (5,1),\ (5,2),\ (6,3),\ \mathbf{(7,3)},\ (8,1), \\ (9,1),\ (10,1),\ (11,2),\ (12,2),\ (13,1),\ (13,2),\ (14,2),\ (15,2),\ (16,1) \end{Bmatrix}$$

The solution after the swap can consequently be defined as (change highlighted in bold):

$$B(s') = \begin{Bmatrix} (1,3),\ (2,3),\ (3,3),\ (4,1),\ (5,1),\ (5,2),\ (6,3),\ \mathbf{(7,2)},\ (8,1), \\ (9,1),\ (10,1),\ (11,2),\ (12,2),\ (13,1),\ (13,2),\ (14,2),\ (15,2),\ (16,1) \end{Bmatrix}$$

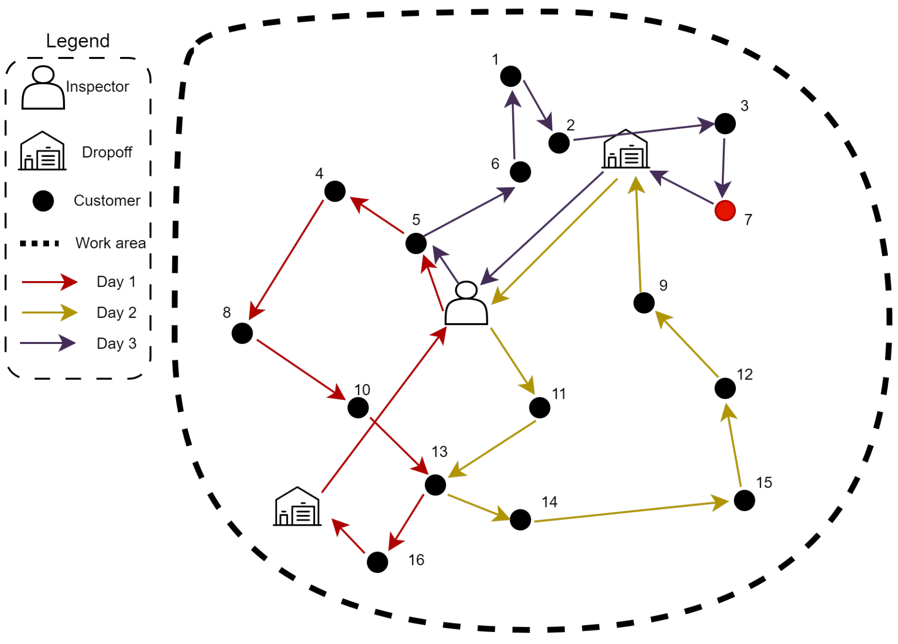

Figure C.1: *Schematic representation of solution before swapping visit combination. The swapped customer is highlighted by the red dot.*

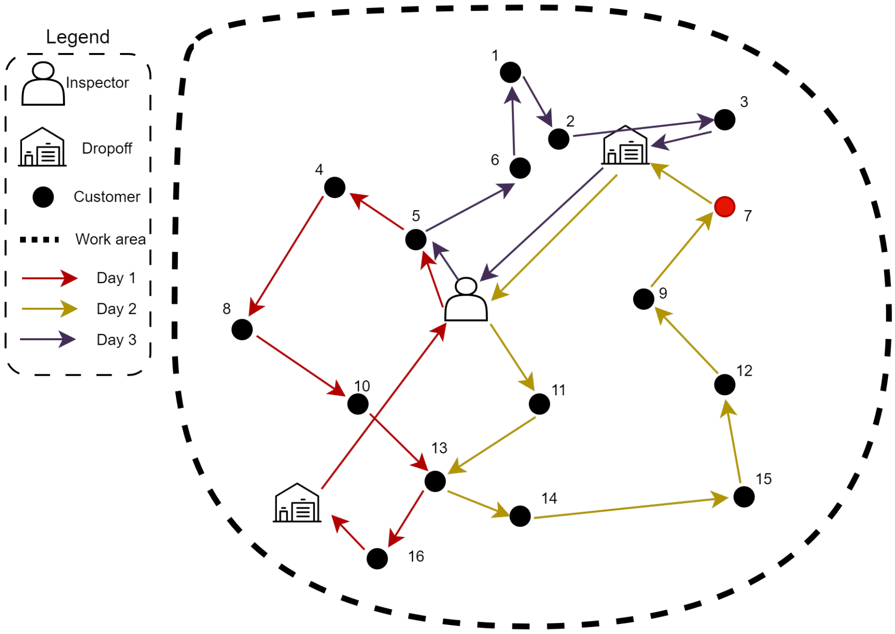

Figure C.2: *Schematic representation of solution before swapping visit combination. The swapped customer is highlighted by the red dot.*

Appendix D

Research design

This Appendix discusses the main research methodology, CRISP-DM and the research approach and outline. The research methodology is described in Appendix D.1 and the research approach is discussed in Appendix D.2.

D.1 Research methodology

The structure of this thesis is based on the CRoss Industry Standard for Data Mining methodology (CRISP-DM) as described in Chapman (2000). This methodology consists of six major phases; business understanding, data understanding, data preparation, modelling, evaluation, and deployment (Schröer et al., 2021). These steps form an iterative process as shown in Figure D.1 and moving between particular phases is possible and beneficial. CRISP-DM is chosen because the phases in CRISP-DM fit the approach for evaluating new sample transportation logistics networks. The success of the data-driven model is highly dependent on the interaction between the data preparation and modelling phases of this research since the usefulness of the data-driven model is dependent on the quality of the data chosen in the data preparation phase. Because CRISP-DM is the de facto standard in data mining research and is highly regarded as a methodology for incorporating customer and business needs within the process to create actionable improvements (Hotz, 2024) this methodology is selected.

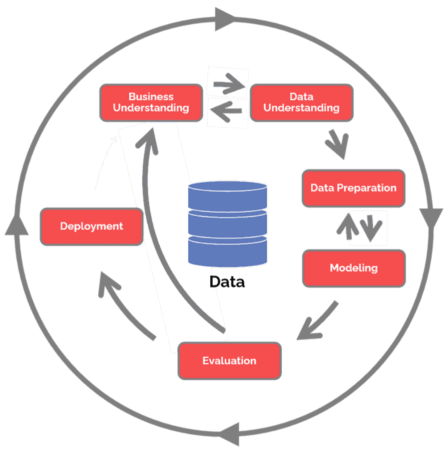

Figure D.1: *the CRISP-DM methodology (Hotz, 2024).*

D.2 Research approach

As mentioned in Section previously CRISP-DM is the foundation of this research design. The formulated sub-questions in Section 1.3.3 are thus linked to different phases of the CRISP-DM cycle. In this section, we cover the approach to answer each sub-research question and give an outline of the structure of this thesis.

Firstly, the current situation at Eurofins is discussed in Chapter 2. This chapter covers the first phase of the CRISP-DM cycle, business understanding. It begins by elaborating on the current logistics networks operated by Eurofins. Followed by elaborating on the network characteristics and organization. The relevant KPIs to an inspector's planning are discussed as well as requirements and limitations of the network. Secondly, literature research is performed to explore relevant methods for vehicle routing problems with constraints applicable to the Eurofins situation and theories for and how to minimize the relevant costs in Chapter 3. Time windows, planning horizons and service times are vehicle routing constraints and are the inclusion criteria that are taken into account in this literature review. Furthermore, this literature review provides an understanding and foundation for the mathematical model and solution techniques that can be used in the modelling phase. The third research question is answered by creating a mathematical formulation of the vehicle routing problem and creating a solution approach based on the literature review. Requirements and assumptions are formulated, on which the mathematical model is based. Using this mathematical model, a solution technique is created. This process is discussed in Chapter 4. Next, in Chapter 5 the model performance is tested through a quantitative analysis, by comparing the model results against the current situation. Before the model is tested against the current situation the parameters for the solution methodologies are tuned to balance solution quality and computational time. Lastly, final conclusions are drawn in Chapter 6 based on the results of the solution outcome and analysis. This chapter provides conclusions and recommendations to the company in regard to the outcomes of this research. The limitations and contributions are discussed as well.

It should be noted that the final phase of the CRISP-DM cycle, deployment, is not included in this research approach. It is unlikely that a completely new planning approach can be implemented within the time span. However, In Chapter 6 recommendations for deployment are provided.

Appendix E

Use of AI

The following tools were in this thesis that (may) utilize Artificial Intelligence (AI):

- ChatGPT 4o
- GitHub CoPilot
- Grammarly
- Mendeley

The AI used in ChatGPT is mainly used for programming assistance in Python, understanding concepts, LaTeXassistance and minor writing assistance such as; looking up synonyms, spelling check, small parts of text, and punctuation. Grammarly was also used to be a spelling check, punctuation and finding appropriate synonyms. The rephrasing or tone suggestions of Grammarly were not used. GitHub CoPilot was used as a programming aid in Python. Mendeley may use AI in the creation of references or the bibliography. Mendeley was used to organise and track the sources in this report efficiently.

After using an AI tool, the content was thoroughly reviewed and edited as needed. We take responsibility for the contents of the final content.