Master Assignment

Using Tabu search to investigate the utilization of stackable crates in a large retailer's supply chain

Supervisors:
Dr. P.C. Schuur,
University of Twente
R. van Ringelestijn
External member
Dr. D. Demirtas
University of Twente

Author:
Stan Lettink

2019

[PUBLIC VERSION]

List of Tables

List of Figures

Management Summary

The retailer is a dutch department store with over 750 stores in nine countries. Part of the retailer is the bakery business unit. This business unit supplies most stores with fresh pastries on a daily basis. To be able to replenish the stores, the retailer has a supply network consisting out of a central bakery and six decentral bakeries. From these bakeries, a fleet of trucks is used to deliver the pastries to the stores. When transported in a truck, the pastries are kept on plastic plates that are shoved into a metal rack on wheels. The retailer realized that by transporting the pastries in this manner, the utilization of the truck is not always at a desired level. For this reason and several other reasons, the retailer wants to do an investigation into another means of transporting the pastries to the stores. The retailer proposed an alternative method that switches the plate and metal rack for a stack of crates on a rolling platform. Since the stacks of crates weigh a lot less, the trucks will be exchanged for smaller vehicles, sprinters. These sprinters are cheaper to drive, but are smaller. In this research, the situation in which the replenishment routes are driven with trucks containing metal racks is referred to as the current situation. Similarly, the intended situation in which the replenishment routes are driven with sprinters and stacks of crates, is referred to as the proposed situation. The goal of this research is to investigate the differences between the current and proposed situation in terms of transportation costs.

Currently the retailer employs a fleet of similar trucks from a decentral bakery to replenish the stores. All decentral bakeries deliver pastries to the stores 6 times per week. The decentral and bakeries also deliver pastries on Sunday, and therefore replenish 7 days per week. The stores have to be visited within certain time windows. Based on these time windows the stores are replenished in one of two shifts. Either the evening or night before, or in the morning. To take better advantage of their fleet, the retailer uses the same vehicles for the evening/night shift as for the morning shift. The transportation costs of the replenishment process consists out of three components: fuel costs, driver costs and vehicle costs. The sprinters consume on average 1 liter diesel per 6 km, while the trucks consume 1 liter diesel per 3 km. The fuel costs per kilometer are therefore lower for the sprinter. In addition to this, the driver of the sprinters will be a bakery production employee, who's salary is lower than that of a certified truck driver. The driver costs per hour are therefore lower in the proposed situation. However the costs of leasing/owning a sprinter are higher due to their lower technical lifespan compared to a truck. The vehicle costs per vehicle are therefore higher in the proposed situation. To estimate the total transportation costs in both situations, the routing schemes in both situations have to be determined. Using the routing schemes the transportation costs in both situations have to be determined.

To find an appropriate model to generate the routing schemes in both situations

with, a literature review is done. Following the investigation in the current situation of the retailer, the routing problem is generalized to a vehicle routing problem with time windows (VRPTW). The routing problem should be solved with primary objective to minimize the total transportation costs. Additionally, the VRP includes three main constraints: a maximum route duration, a maximum loading capacity per vehicle and the time windows. The added complexity of the situation of the retailer is the fact that a fleet at a decentral bakery is used twice per day, once in the evening/night and once in the morning.

With the specifics of the routing problem known, the next step is to select an appropriate model to determine the routing scheme and accompanying transportation costs with. An exact method for finding the optimal routing schemes is found to be unfitting due to the complexity of the constraints and the limited existence and availability of exact solvers. Instead of finding an exact solution, the optimal replenishment routes between the decentral bakeries and their allocated stores will be approximated by a heuristic. The selected heuristic is the tabu search heuristic reported in [4], due to its its simplicity and adaptability to the case of the retailer while still providing good quality solutions.

After adapting the heuristic to fit the VRP, the routing schemes in both situations could be determined. Since the average order size on Saturday is significantly higher than during the rest of the week, a different routing scheme has to be used on Saturday than during the rest of the week. As a result, four different routing schemes are determined using the routing model: routing schemes for the weekdays and Saturday in the current situation and routing schemes for the weekdays and Saturdays in the proposed situation. After generating all routing schemes for the transportation, the corresponding transportation costs are calculated.

The results, shown in Figure E.11, show that the fuel costs and driver costs in the proposed situation are lower compared to the current situation. On the contrary, the vehicle costs in the proposed situation are higher due to the fact that the fleet size in the proposed situation is larger and the costs per vehicle are also higher. The degree of savings vary per decentral bakery. The decentral bakery in Zwolle achieves an estimated saving of 2.81% while Dordrecht achieves an estimated saving of 22.30% by changing to the proposed situation. The only difference between the weekdays and Saturday is an increase in average order size on Saturday. Comparing the routing schemes of a weekday with those of Saturday shows that the savings on Saturday are lower than the ones achieved on a weekday. Nevertheless, overall the proposed situation seems to be performing better in terms of transportation costs. The estimated savings when changing to the proposed situation is approximately 13.15% of the estimated total yearly transportation costs in the current situation.

Following the results of this investigation, the retailer should first perform another investigation into the qualitative aspects of implementing the change. This analysis should focus on the effects this change might have on any stakeholder of the replenishment process. After an analysis into the qualitative aspects of the change, the retailer could make a more informed decision. If the retailer decides to implement the change, the next phase should be a pilot at one decentral bakery. If the results of this pilot are as expected and no deal-breaking disadvantages are discovered, the retailer can move on the next phase. The final phase concerns the gradual introduction of the changes to the supply chain. First the crates should be introduced after which, depending on the lease contracts, the sprinters can be introduced into the supply chain.

Contents

,

Chapter 1

Introduction

The research for this report was conducted at a large dutch retailer for the completion of the MSc. Industrial Engineering and Management at the University of Twente. This report contains a description of the research conducted regarding the investigation into the utilization of stackable crates within a part of the bakery supply chain.
In the first section, the retailer is introduced, after which a brief description of the relevant logistical processes is given to provide the context for this research. The motivation for this research and the problem statement is provided in the second section. The third section contains the main research question and the framework with which this question is answered. Section 4 contains the scope statement followed by Section 5 that provides the research approach and outline of this report.

1.1 Research context and company description

The Dutch department store has over 750 stores in nine countries and with 19000 employees serving over six million visitors every week. From these 750 stores 257 are franchises. The store is centrally managed with four main business units: Service office, Distribution Centre, Bakery and Sales. The bakery business unit supplies most of the stores in the Netherlands with pastries. These stores sell the pastries to the customers at their bakery department or serve them to the customers in their restaurant located in some of the larger stores.
The bakery business unit is comprised out of three main departments:

1. The production office
2. The central bakery
3. The decentral bakeries

The bakery business unit is managed by the production office. Additionally, bakery related tasks such as purchasing, finance and administration are handled by this office. The production office is located in Almere and shares the same facility with the central bakery and one of the decentral bakeries. The central bakery produces half-fabricates and does the stock keeping for the decentral bakeries. The stock at the central bakery consists out of refrigerated half-fabricates and the raw materials needed to finish the half-fabricates. This stock is used to replenish six decentral bakeries throughout the Netherlands. As was mentioned before, one of these decentral bakeries is located in the same facility as the central bakery. The other five decentral bakeries are located in Sneek, Helmond, Zwolle, Doetinchem and Dordrecht. Figure 1.1 shows the central and decentral bakeries on a map of the Netherlands.

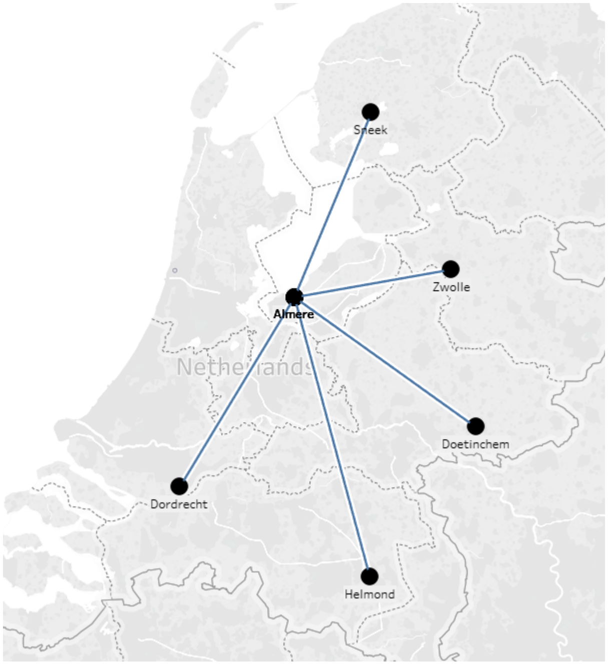

Figure 1.1. The central and decentral bakeries displayed on a map of the Netherlands

The half-fabricates and raw materials usually arrive at the decentral bakeries in the morning. Once the replenishment arrives, the production employees at the decentral bakeries start finishing the products. Once finished, these pastries are combined with other finished pastries originating from external suppliers and placed on plastic food-grade plates. These plates are shoved in large metal racks to provide protection and enable transportation. Once the metal racks are filled with the ordered pastries, they are placed in trucks that transport them to the corresponding stores. This process is repeated every day of the week at all decentral bakeries to supply 337 stores in the Netherlands daily. The details of the production and transportation process together with the exact specifications of the metal racks are elaborated upon in Chapter 2. Figure 1.2 shows a flowchart representing the general flow of products and containers between the central bakery, the decentral bakeries and the stores.

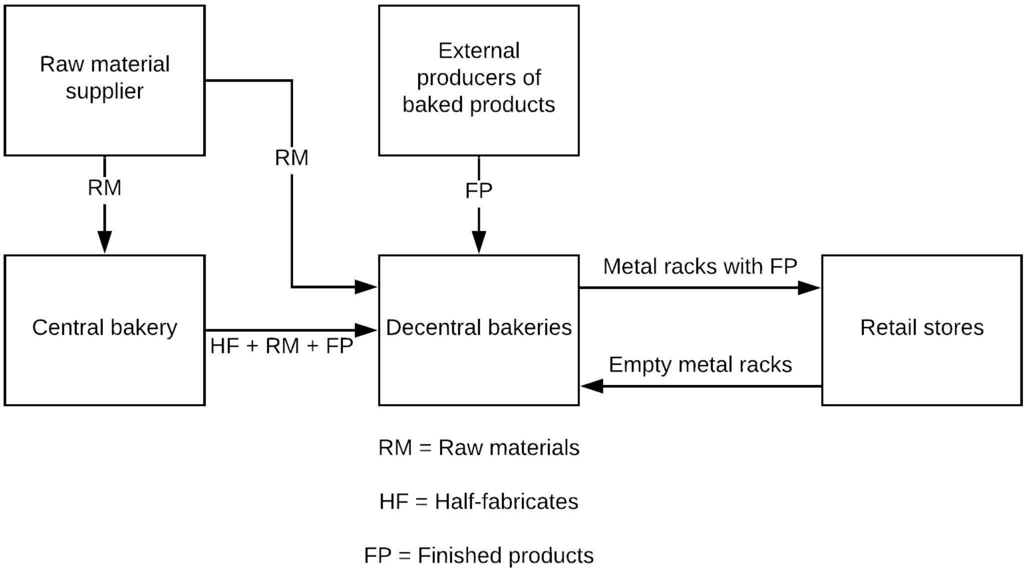

Figure 1.2. A flowchart displaying the general flow of materials and products

1.2 Problem statement and research motivation

The management of the bakery business unit wants to change the transportation process and came to this conclusion for two reasons. The first reason is that the management expects that, due to external factors, the current way of transporting the pastries may become infeasible. One of the most important external factors is that local law concerning noise limits and pollution levels in certain areas is becoming stricter over time. The trucks the store is currently transporting the metal racks in are large. Because of their size, they cause a relatively large amount of pollution and make a lot of noise. The metal racks used for the transportation of pastries also cause a lot of noise, especially when pushed over uneven streets. The unloading process mostly happens during the night and is therefore likely to exceed the noise limits and raise complaints from the surrounding inhabitants.

The retailer expects that efficiency and the business value added by logistics will become more of a deciding factor in the future. Therefore, the second reason to change the transportation process is to increase the competitiveness of the bakery's logistical process. Internal research shows that the current trucks utilize, depending on the day of the week, between 40% and 80% of their capacity every trip from the decentral bakeries to the stores. A large part of this inefficiency is caused by the size of the metal racks relative to the volume of pastries they contain. For example, if only one plate of baked products has to be transported to a certain store, a full-sized metal rack that could contain 48 plates has to be used. The retailer intends to reduce transportation costs by increasing the utilization of the truck capacity each trip between the decentral bakery and the stores.

For these reasons, the retailer intends to change two aspects of the transportation process.

- Replace the metal racks used to transport to pastries with a stack of crates that is moved on a dolly.

- Replace the trucks currently used to drive the replenishment routes with a smaller vehicle referred to as a sprinter.

By changing these aspects of the replenishment process, the retailer hopes to improve upon the current process and achieve several advantages. Some of these expected advantages are:

1. Pastries can be transported more space-efficiently between the decentral bakeries and the stores.
2. The vehicles used for transportation of the baked goods can access the city centers more easily.
3. Unloading creates less noise and disturbance of the surroundings.
4. The bakery will be better prepared for future changes in laws regarding the limitation of pollution in cities and sound caused by unloading products.
5. Easier to find drivers, due to the them not needing a C(E) driver's license. This makes it easier
6. A reduction in the amount of space needed for storing the transportation containers in the decentral bakeries and stores.
7. A reduction in the amount of pollution caused by their vehicle.

Both the type of crate and sprinter have already been selected by the management. Throughout this research, a distinction will be made between two situations: the current situation and the proposed situation. The **current situation** is the situation in which a single type of metal rack is utilized as a container for the pastries. The replenishment routes are driven by a single type of truck in this situation. In the **proposed situation** a single type of stackable crate will be used as container for the pastries. The replenishment routes are driven by a single type of sprinter in this situation. The specifications of the containers and vehicles used in both situations are known and will be discussed in a later chapter.

Before making the investment to replace the vehicles and containers, the retailer wants to research whether the combination of new vehicles and containers will actually save money and, in case it does, estimate how much money it will save. Therefore, the goal of this research is to *quantitatively analyze the differences in terms of transportation costs, achieved by utilizing the intended stackable crates and smaller vehicles.*

1.3 Research Questions

Following the research goal, the following main research question is formulated:

> *What are the differences in terms of transportation costs, achieved by replacing the current containers and vehicles with the intended stackable crates and smaller vehicles for the transportation of pastries?*

To answer this research question, a number of sub-research questions are formulated. These questions not only give insight into the aspects of the main problem, but also provide a framework used to structure this research.

The first step in this research is to acquire a clear understanding of the existing production and logistical process between the decentral bakeries and the stores. By comparing both the current and proposed situation, it is important to understand what factors make up the overall transportation costs and how these cost factors differ in both situations. As a result, the following sub-research questions can be formulated:

1. What logistical processes are present between the decentral bakeries and the stores?
2. How does a store order translate into required vehicle capacity for the replenishment routes between the decentral bakery and the stores?
3. What are the cost factors that make up the overall transportation costs between the decentral bakeries and the stores?
4. How do these cost factors change between the current and proposed situation?

The next step is to select a method to determine the routing while minimizing transportation costs in both situations. The available literature will be reviewed in order to select the most appropriate method in the case of the bakeries. Concerning the model selection, the following sub-research questions are formulated;

5. What routing models are most appropriate to determine the routing that minimizes the transportation costs for the decentral bakeries?

With the model selected, the routing in both situations can be determined. Both the current transportation costs and the transportation costs in the proposed situation will be determined using the routing resulting from the model. The actual incurred transportation costs, from the annual report, will solely be used for verification purposes. Hence, the following sub-research questions can be formulated:

6. What is the routing in the current situation while minimizing transportation costs?
7. What is the routing in the proposed situation while minimizing transportation costs?
8. What is the difference between the estimated transportation costs in the current situation and the estimated transportation costs in the proposed situation?

1.4 Scope Statement

The scope for this research is determined in conversation with the problem owner at the retailer, and the University of Twente. A time frame of approximately 6 months is available for this research. Due to the limited amount of time, not all aspects of the problem can be visited in this research. The scope of this research is therefore limited to the logistical processes between the decentral bakeries and the stores. Figure 1.3 shows the scope of this research outlined in a flowchart.

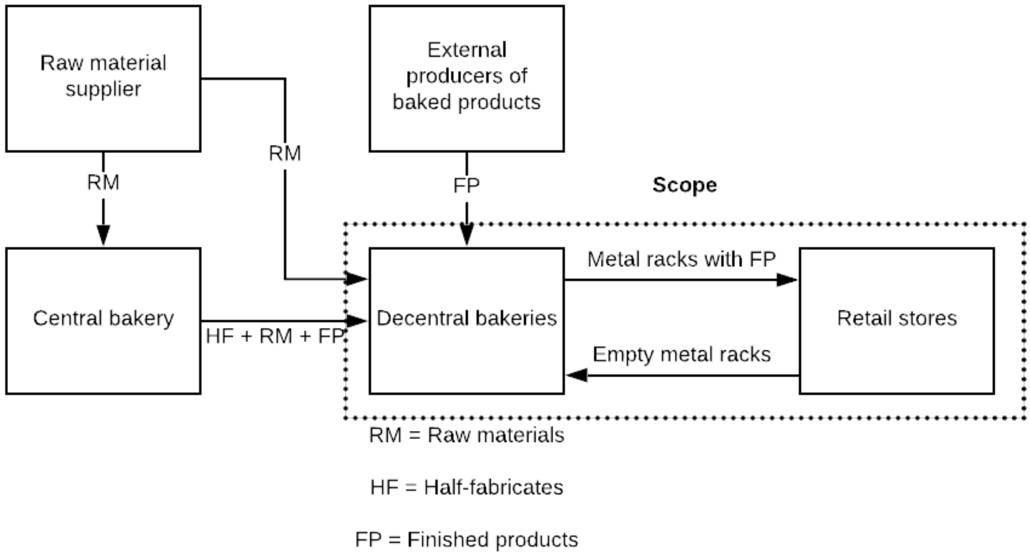

Figure 1.3. The scope of this research marked in a flowchart

Furthermore, this research will focus on estimating the transportation costs in both situations, and the analysis of the difference between these. Other aspects such as noise and environmental impact of unloading in the new situation will therefore not be taken into account. As was mentioned earlier in this chapter, two situations will be compared. Other situations such as one with multiple types of vehicles (truck and sprinters) or multiple types of containers (metal racks and stackable crates) will not be taken into account. As was mentioned before, the goal of this research is to provide the bakeries with an investigation into difference in transportation costs resulting from changing to utilizing the intended smaller vehicles and crates for the transportation of pastries. To achieve this goal, a routing scheme that minimizes the transportation costs in both situations will be determined.

The deliverable at the end of this research will be an analysis on the transportation costs in the current and proposed situation. The transportation costs in both situations are determined based on the modeled supply network in both situations. Based on the results of the modeling efforts, several recommendations will be given relating to the proposed changes to the replenishment process.

1.5 Research approach and outline of the report

The research will be conducted according to the structure provided by the sub-research questions. The first step in this research is to describe the current logistical processes and how these translate to the transportation costs. In addition to this, the proposed situation will also be described and how it differs from the current situation. While doing the research in this step, data is being collected that can be used for the comparison later in this research.

The second step is to review the available literature for routing models that fit the case of the the bakeries. First, the aspects of the routing problem in the case of the bakeries have to be determined. Secondly, methods for solving the routing problem will be evaluated on performance and relevance to the aspects of the bakery routing problem. The final step is, once an appropriate model has been found, to design this model so it can be applied to the case in this research.

The third step is to collect and structure the data so it can be applied to the model. In this part of the report, the data sources and the data collection process are described. To create an overview, the acquired data will be displayed.

The fourth step is to apply the model and determine the routing for two situations: the current and proposed situation. Based on these routing schemes, the transportation costs will be estimated, analyzed and compared. In the final step, the results from the previous step are used to answer the main research question and a recommendation for implementation will be formulated. Figure 1.4 shows the research approach and their sequence.

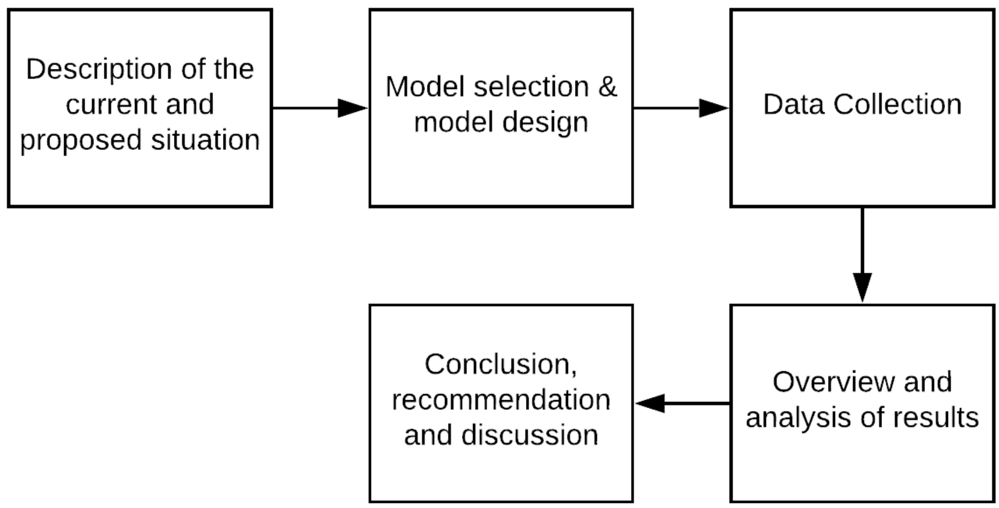

Figure 1.4. The research approach

The outline of the report is similar to the research approach. First, a description of the current and proposed situation and which cost factors make up the transportation costs is given in Chapter 2. The literature review is reported in Chapter 3 and the model design in Chapter 4. Chapter 5 contains the results from the data collection step process and shows a visualization of the data that will be used as input for the routing model. The results of applying the model are analyzed and visualized in Chapter 6. The conclusion and a recommendation for implementation are given in Chapter 7. Finally, Chapter 8 contains a brief discussion about further research and the limitations of this research.

Chapter 2

Current situation and proposed changes

In this section, the current situation and the proposed situation are described. The focus in this chapter is on the replenishment process and accompanying cost factors. This chapter should provide sufficient information so that a relevant and appropriate routing model can be selected. Firstly, a brief description of the production process is given to provide some context to the transportation process. Once the context is described, the second section contains general information about the transportation process. Topics such as the calculation of order volume and time windows per store will be discussed in this section. The second section concludes with a description of the cost factors that make up the transportation costs. The third section reports the specifications of the vehicles and containers in both situations. Additionally, some comments regarding the average utilization of a single container in both situations are made in this section. The final part of this chapter contains a brief conclusion.

2.1 The production process in the decentral bakery

Most decentral bakeries receive a shipment with stacks of crates containing raw materials, half-fabricates and finished pastries from the central bakeries six times per week. Once arrived, the shipment is directly placed in the freezers where it is stored until needed. Depending on the decentral bakery, the stores send a replenishment order six or seven times per week. Based on these orders a production planning is made. Based on this production planning, the pastries are retrieved from the freezer and finished on the production line if necessary. Depending on the product, the products are either put on paper plates or in a box. These pastries are then placed on plastic plates, put in metal racks and transported to the outlet-fridge. A metal rack containing plates is shown in Figure 2.1. In the outlet-fridge employees have lists containing the orders for every store for the next day. Based on these orders, the employees place the products from the metal racks onto other metal racks that contain the orders for a certain store. Pastries destined for different stores are not allowed to be placed in the same metal rack. At the end of the shift, the outlet fridge should contain metal racks categorized under headers that mark the different stores. The metal racks containing the pastries are now ready for transportation. Appendix A shows a flowchart of the production process in the decentral bakeries.

Figure 2.1. A metal rack containing plates with doors open (left) and with doors partially closed (right)

2.2 Transportation

There are two replenishment shifts, one in the morning and one in the evening/night. During the evening/night shifts, the orders for the next day are delivered. During the morning shifts, the orders for that same day are delivered. For example, the orders for Tuesday are delivered during the Monday evening shift and the Tuesday morning shift. Each store is replenished once per day. Therefore a store is either replenished during the morning shift or the evening shift. A large advantage of working in this manner is that the fleet size can be reduced since a vehicle used in the evening shift can also be used in the morning shift. At the start of each shift, the truck drivers gather at the decentral bakery. Each truck driver is assigned a route for that shift. Based on the markers in the outlet fridge, they place the metal racks that belong to the stores in their route into their truck. Once loaded, the truck drivers take the keys required for supplying the stores outside opening hours and start with their route.

2.2.1 Transportation volume

The metal racks currently used for transportation have the advantage that they allow for excellent protection of the pastries and clear visibility of the contents. However, one of the major disadvantages for the retailer is the significant volume and weight these racks have relative to their contents. This volume especially becomes disadvantageous during transportation, where the capacity of the truck transporting the pastries is limited. To calculate the volume a store order requires, the following information is needed per pastry:

- How many of this type of pastry are allowed on one plate?
- How many plates with that pastry can fit in a metal rack.

The number of a certain type of pastry that fits on a single plate is determined by the width and length of the pastry. The number of plates that fit in a metal rack is determined by the height of the pastry on the plates. There is room for 48 plates in one metal rack, 24 in height and two in depth. However, due to the height of some of the pastries, some of the slots have to be skipped. Figure 2.2 shows a close-up of the plates in their slots.

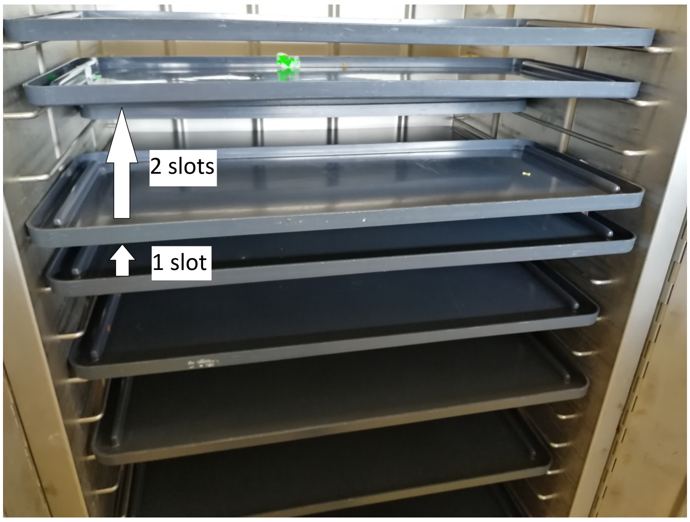

Figure 2.2. A close-up of a metal rack with plates

For this reason, to determine the transportation volume, the bakeries register the height in number of required "slots" for every product. An example of the listed information per pastry is shown in Table 2.1.

Table 2.1: *An example of the information required per product*

Type	**Amount per plate**	**Plates per metal rack**
Dripcake	3	16
Carrot cake	14	24
Apple pie	2	48

Using this information the required volume for an order can be estimated. The following equation is used to calculate the amount of (partial) metal racks needed per pastry per store:
$V_{i,j}$ = (partial) metal racks needed for pastry i to store j
$Q_{i,j}$ = amount of pastry i ordered by store j
P_i = amount of pastry i that fits on one plate
M_i = amount of plates with pastry i that fit in one metal rack (based on the height of the pastry)

$$V_{i,j} = \frac{Q_{i,j}}{P_i} * \frac{1}{M_i}$$

The order volume is calculated by taking the sum of the required volume of all pastries in the order ($\sum_i V_{i,j}$). The order volume has to be rounded to the next integer since only pastries of one order may be present in one metal rack. The number of metal racks needed for a route is the sum of the required metal racks per store order in that route. To give an example of how a store order translates into required capacity consider the store order in Table 2.2. In this example the information in table 2.1 is used.

Table 2.2: *An example of a store order*

Pastry	**Amount ordered**	**(Partial) plates**	**(Partial) metal racks**
Dripcake	11	3.67	0.23
Carrot Cake	50	3.57	0.14
Apple Pie	30	15.00	0.31

As the table shows, this particular store order results in $0.23 + 0.14 + 0.31 \approx 0.69$ metal racks. Because only one store order may be present per metal rack, the resulting required volume for this store order is one metal rack.

2.2.2 Order volume and routing

The drivers that replenish the store, drive according to a routing scheme that is made beforehand. Most bakeries have a different routing scheme for the replenishment routes during the week than on Saturday. This is caused by the significant difference in order volume between weekdays and Saturdays. To illustrate this difference, the order volume for the decentral bakery in Almere is shown in figure D.1.

The decentral bakeries in Dordrecht and Almere also replenish their stores on Sunday. Since not every store is opened on Sunday, these bakeries also have a different routing scheme for Sundays. Figure D.1 illustrates the difference between demand on Sundays and the rest of the week for the decentral bakery in Almere.

2.2.3 Time windows

The time windows of some of the stores also have a large impact on the replenishment routing. As mentioned before, the replenishment routes for a certain day are driven both the evening the day before or the morning of the day itself. The main reason why not all stores are replenished during the night is that some stores can only be supplied during the morning. There are three possible reasons that delivery at a store is restricted to a time window:

- Local law prohibiting access to a zone during certain time periods.
- Local law prohibiting unloading during certain time periods.
- The preference of the owner of the franchise store.

While designing the routing scheme, these time windows are a limiting factor and have to be taken in to account. Based on both the order volume and time windows per store, the retailer determines the routing schemes using routing software. Once the routing schemes are generated, the decentral bakery managers alter the routing schemes over time, based on the feedback of the store owners and truck drivers.

2.2.4 Transportation costs

The transportation costs for the replenishment of pastries at the stores consist out of a three main factors: driver costs, fuel costs and vehicle costs. The driver costs consist of the salaries paid to the drivers of the vehicles. The retailer employs two type of employees to drive the trucks: permanent employees and temporary employees. The percentage of temporary employees varies per bakery. Some bakeries utilize mostly drivers with contracts at the retailer while some have twice as many temporary drivers as permanent drivers.

The vehicle costs consist of the maintenance, insurance and lease/depreciation costs. Concerning the depreciation and lease costs, the retailer does not buy new vehicles. It rather leases these vehicles for several years. After this period, the retailer buys these vehicles for a predetermined amount. Naturally, after buying the vehicle, the lease costs for the vehicle are replaced by the depreciation costs of that vehicle. Throughout the lease and depreciation period, the maintenance and fuel costs are paid by the retailer. Figure 2.3 visualizes the composition of the transportation costs.

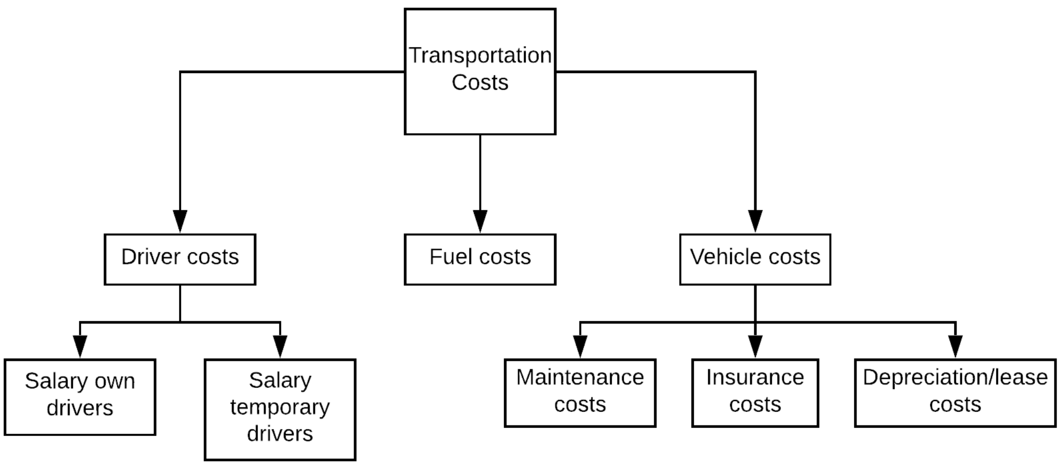

Figure 2.3. The composition of the transportation costs

The decentral bakeries in Helmond and Dordrecht are different in their transportation process in the sense that most or all of their replenishment routes are outsourced. These two decentral bakeries do therefore not incur the same transportation costs. Rather they pay an external company for the transportation of pastries between the decentral bakery and the stores. However, for the sake of comparison in this research, the routing for these bakeries in the current and proposed situation will be determined as if they handled the replenishment themselves.

2.3 Cost factors in the current and proposed situation

Both the crates and sprinters to be used in the proposed situation have already been selected by the management. Changing to using these crates and sprinters will affect the overall transportation costs in some way. This section starts with a description of the slight changes in the production process, after which the specifications of the containers and vehicles in both the current and proposed situation are given. Subsequently, using the specifications, a preliminary analysis is performed concerning the utilization of a single container in both situations.

2.3.1 The production and transportation process in the proposed situation

In addition to the transportation process, the production process inside the decentral bakeries will be (slightly) impacted by the change to new containers. Currently, once the products are finished and packaged at the production line, the employees place these pastries on plastic plates in the metal racks. In the proposed situation these metal racks would be exchanged for stacks of crates. Once the stackable crates are stacked up to 14 crates high, the stacks are moved to the outlet fridge and distributed to other stacks of crates destined for the stores. When empty, these crates can be folded. This causes the empty crates to take up less space than the empty metal racks. The reduction in wasted space is likely to bring some advantages such as more oversight on the production floor and less time spend sorting through the containers. Whether this advantage also results in a costs reduction is hard to measure and likely negligible, since there is currently no reported problem with empty containers.

Changing to new containers and vehicles will have the most impact on the transportation process. The sprinters can be driven by any employee with a BE driver's license. In the current situation, the driver needs a C driver's license to drive the truck. The C license is owned by far fewer people and requires relatively expensive re-certification every few years. Currently there is a scarcity for truck drivers in the market, making it harder for the retailer to acquire truck drivers. On the other hand, the sprinter can be driven by anyone with a BE drivers license, which could be a regular decentral bakery production employee. The hourly salary of a production employee is generally lower than that of a truck driver, bringing some potential financial advantages. Whether the implementing the proposed situation actually brings financial advantages is depended on the overall transportation costs. The next section reports the specifications of the vehicles and containers in both situations, and aims to provide some insight into the cost saving potential of the combination of crates and sprinters.

2.3.2 Specifications

Table D.1 and D.2 show the specifications of the containers and the vehicles, in both the current and proposed situation. The sources and procedures used to determine the specifications of the vehicles and containers will be extensively discussed in Chapter 5.

The maximum volume listed in Table D.1 is based on the number of containers that fit in one vehicle concerning its weight and volume constraints. Furthermore, the average salary for a sprinter driver is based on the average salary of a production employee. The average capacity listed in Table D.2 is based on the information that: (i) one crate can hold the same amount of pastry as one plate, regardless of the height of the pastries; (ii) a plate in a metal rack has an average height of "2 slots".

Using the averages listed in Table D.2, the average capacity per vehicle measure in number of plates can be calculated. According to these averages a sprinter can hold up to 560 plates and a truck can hold up to 924 plates. In practice, the average number of plates containing pastries placed in a single metal rack or stack of crates might be lower than its maximum capacity. This is because once the containers for a certain store are being filled, the last container is likely to not be filled to its maximum capacity. This brings us to one of the cost saving potentials of the stack of crates. Because only a single store order can be present in one stack or rack, the remaining capacity of the container remains unused in case the store order doesn't fill a round number of metal racks. This is also the reason why a conversion factor cannot be used to calculate the required number of crate stacks using the required number of metal racks. The unused capacity in a stack is on average less, compared to the unused capacity in a metal rack, because the stacks have a smaller capacity. To illustrate, using the capacity per container and the average demand per week, the average utilization of a single type of container can be calculated. The average utilization of a single type of container is estimated and displayed in Figure E.1

The graph seems to confirm that, because the stacks can hold fewer plates worth of pastries, the average utilization seems to be higher. However, the increased average utilization per container may not directly translate into lower overall transportation costs. The transportation costs are largely determined by the routing of the vehicles and the accompanying driver, fuel and vehicle costs.

2.4 Conclusion

The goal of this chapter is to gather all relevant information required for finding an appropriate model with which to determine the routing in the current and proposed situation. Concerning the replenishment process, there are three constraints that have to be taken into account when determining the routing: vehicle capacity, route duration and the time windows. Due to the time windows, some stores can be replenished during the evening and night, while others have to be replenished the next morning. For this reason the retailer decided to use two shifts for the delivery of pastries to the stores: an evening and a morning shift. To reduce the total fleet size, a single vehicle is used twice per day, in both the evening and the morning shift.

Due to the difference in average order size throughout the week, the retailer drives according to three different routing schemes each week. One for the Monday till Friday, one for Saturday and one for Sunday. The transportation costs of these routing schemes consists out of three components: fuel costs, driver costs and the vehicle costs. Because of the lower fuel consumption and lower hourly rate, the fuel costs and driver costs are expected to be lower when using the intended sprinters in the proposed situation. The vehicle costs per vehicle are higher in the proposed situation caused by the shorter technical lifespan of the sprinters.

A brief analysis of the average utilization of a single metal rack and a single stack of crates is performed based on the demand. On average a stack of crates seems to have a higher utilization than a metal rack. Whether this translates into a lower transportation costs has to be determined based on the overall transportation costs.

In the next chapter a routing model is selected with which the routing for each decentral bakery can be determined. The routing should be designed such that the transportation costs are minimized while taking into account the vehicle capacity and store time windows. The routing model should also take into account that vehicles can be used two times during the total delivery window, once in the evening and once in the morning.

Chapter 3

Model Selection

The first step in estimating the transportation costs for the current and proposed situation is selecting an appropriate model with which to determine the routing. Determining the replenishment routes from a depot to serve a set of customers is a well-reported problem in the literature. This type of problem is referred to as a Vehicle Routing Problem (VRP). The objective of the vehicle routing problem varies, but the most common objective is to minimize the transportation costs of all replenishment routes while complying with the following constraints:

- The sum of demand of the customers visited in one route may not exceed the vehicle capacity
- All routes have to start and end at the depot.
- All stores have to be replenished using the available vehicles.

Since first described in [7], hundreds of models and algorithms have been proposed for solving various versions of the VRP optimally[14].

The first step in finding an appropriate routing model is to determine what kind of routing problem needs to be solved. For this reason, Section 3.1 creates an overview of all the aspects of the retailer VRP. Subsequently, Section 3.2 and 3.3 evaluate different exact and approximate methods for solving the retailer VRP. The chapter ends with a brief conclusion.

3.1 The general aspects of the VRP

Because hundreds of models and algorithms have been proposed in literature, a framework is needed the identify the key aspects on which to differentiate the numerous vehicle routing problems. A vehicle routing problem can be categorized by the following aspects:

1. One vehicle or multiple vehicles per depot.
2. A single type of vehicle or multiple vehicle types per fleet.
3. Multiple depots or a single depot.
4. Type of transportation jobs per vehicle.
5. The constraints imposed on the problem.

6. Type of decisions made by the planner.
7. The management objective.

To replenish all stores on a daily basis, multiple vehicles are required. These vehicles are of a single type, making the fleet homogeneous. Therefore, the retailer VRP includes multiple vehicles of a single type in both the current and proposed situation.

At first sight it seems that the retailer VRP concerns multiple depots, because multiple decentral bakeries replenish the stores. In practice, each store is served by only one bakery. In conversation with the retailer is decided to split the VRP into multiple single depot routing problems. This decision has been made because the retailer has no intention to change the store-to-bakery allocation and to greatly reduce the complexity of the computations. Therefore, the current store-to-bakery allocation will remain unchanged and the problem can be transformed into multiple single depot vehicle routing problems. Table D.4 shows the number of stores allocated to each of the decentral bakeries.

Concerning the type of transportation jobs, the trucks deliver the complete order of pastries in a single delivery per day. In practice, the driver of the truck takes back the empty metal racks after delivery. The retailer reported that taking back the empty containers never caused any issues regarding capacity. Therefore taking the transport of empty containers into account is redundant, and will therefore not be included in the routing model. Additionally, the retailer decided that order-splitting is not allowed. This means that delivery at a store always includes the full order.

The next aspect concerns the constraints imposed on the routing problem. The following three constraints are the most important ones while determining the routing for the decentral bakeries:

- Each vehicle has a fixed capacity in terms of weight and volume. The total order volume and weight during each route may not exceed this capacity.
- The drivers can only drive/work for a maximum of 9 hours. In addition to this, a mandatory breaks of 30 minutes has to be included in case the shift is longer than 6 hours.
- A replenishment vehicle may only start unloading within the time windows of each store.

Many stores have time windows that are the result of several factors described in Chapter 2. Most time windows dictate that stores have to be visited early in the morning, when unloading is allowed and before the store opens. The time window in which all stores have to be served, varies per decentral bakery. This time window is referred to as the *overall delivery window*. Stores allocated to most decentral bakeries are usually served between 18:00 and 8:00 the next day. For the decentral bakeries in Dordrecht and Almere the overall delivery window is slightly different. The stores allocated to these decentral bakeries can be replenished between 18:00 and 9:00 the next day.

The nature of the decisions made is also a relevant aspect of any VRP. In the case of the retailer, the route planning is made offline, meaning that the route planning model does not update the route while the replenishment route is being driven.

Rather, the model determines the routing based on the average order volume in a normal week (with no holidays or special events), with all possible routes and costs known in advance. For the sake of simplicity, we consider the travel times and therefore the traveling costs to be constant/deterministic.

Finally, the objective of the VRP in this research is to minimize the transportation costs based on the routing and the number of vehicles used. First, the routing is determined while minimizing transportation costs. As was mentioned in the previous chapter, most vehicles are used twice per day, once in the evening and once in the morning. Therefore the fleet size is not simply equal the number of replenishment routes driven from each decentral bakery. The model to be applied on the retailer's case has to include a rule with which the vehicle costs can be determined based on the amount of evening and morning routes being driven. Once the number of evening and morning routes are known, the maximum number of routes between the two is the number of vehicles needed for the replenishment of the stores. A summary of the aspects of the retailer VRP is given in Table 3.1

Table 3.1: *A summary of the aspects of the VRP in this research*

VRP aspect	**VRP in this research**
One or multiple vehicles	Multiple vehicles
Single or multiple depots	Single depot
One or multiple vehicle types	One type of vehicle
Type of transport jobs	Delivery only, no order splitting
Constraints	Capacity, duration and time window constraints
Decisions	Offline/static routing, constant/deterministic traveling costs
Objective	Minimize transportation costs based on the fuel, driver and vehicle costs

3.2 Exact formulations for the VRPTW

The way the VRP is formulated is dependent on the algorithm used for solving and the aspects mentioned in the previous section. Considering the aspects listed in Table 3.1, the most relevant standard formulation is that of the vehicle routing problem with time windows (VRPTW). Several surveys have been conducted regarding the exact formulations and methods for the VRPTW. One of these formulations is the two-index vehicle flow formulation [6]. This problem formulation is briefly discussed to provide an example of the formulation of the objective function and its constraints. The formulation is as follows:

Sets:
N = Set of stores with pastries (N = 0,....,n) with v_0, v_{n+1} = decentral bakery
K = Set of vehicles K = (1,....,m)
Parameters
c_{ij} = costs of traveling from store i to j with $(i, j) \in N$
t_{ij} = duration of traveling from store i to j with $(i, j) \in N$
d_i = demand for store i $\in N$
C = capacity each vehicle $k \in K$
s_i = service time for store i $\in N$
E = earliest possible arrival at depot

L = latest possible arrival at depot
a_i = earliest possible arrival at store i $\in N$
b_i = latest possible arrival at store j $\in N$

Decision variables

$$x_{ijk} = \begin{cases} 0 & \text{if vehicle } k \text{ does not drive between store } i \text{ and } j \\ 1 & \text{if vehicle } k \text{ drives between store } i \text{ and } j \end{cases}$$

w_{ik} = start of service at store i when replenished by vehicle k

Objective Function
Minimize $\sum_{k \in K} \sum_{i \in N} \sum_{j \in N} x_{ijk} c_{ij}$

Constraints

$$\sum_{k \in K} \sum_{j \in \Delta^+(i)} x_{ijk} = 1 \quad \forall i \in N \quad (1)$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \quad \forall k \in K \quad (2)$$

$$\sum_{i \in \Delta^- j} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{jik} = 0 \quad \forall k \in K, \forall j \in N \quad (3)$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \quad \forall k \in K \quad (4)$$

$$w_{ik} + s_i + t_{ij} - w_{jk} \leq (1 - x_{ijk}) M_{ij} \quad \forall k \in K, \forall (i,j) \in A \quad (5)$$

$$a_i(\sum_{j \in \Delta^+(i)} x_{ijk}) \leq w_i k \leq b_i(\sum_{j \in \Delta^+(i)} x_{ijk}) \quad \forall k \in K, \forall i \in N \quad (6)$$

$$E \leq w_{ik} \leq L \quad \forall k \in K, \forall i \in \text{0,n+1} \quad (7)$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq C \quad \forall k \in K \quad (8)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, \forall (i,j) \in A \quad (9)$$

$$x_{ijk} = 0 \text{ or } 1 \quad \forall k \in K, \forall (i,j) \in A \quad (10)$$

The objective function in this formulation represents the objective to minimize the traveling costs. This function contains the sum of the costs of each traversed arc in the solution. In the case of the retailer, this objective function should be altered to contain the fixed costs of utilizing a vehicle for a route. Methods have been proposed to include the vehicle costs in the objective function [8] and to allow vehicles to remain at the depot at zeros costs [6]. Unfortunately, these methods would not directly fit the situation of the retailer, since a vehicle used in the evening routes could also be used for the morning routes. It is therefore important to note that the notation for vehicle k does not characterize a single vehicle but rather represents a route driven by one vehicle.

Constraint (1) ensures that each store is assigned to exactly one route/vehicle. Constraints 2 to 4 characterize the flow on the path to be followed during by vehicle k. Constraint (2) ensures that each route includes leaving the depot exactly one time. Constraint (4) ensures that each route includes arriving at the depot exactly one time. Constraint (3) restricts each route to leave each store that they arrive at and arrive at each store that they leave.

Constraints 5 till 8 guarantee schedule feasibility concerning the time windows and capacity constraints. Constraint (5) ensures that in case the route traverses arc (i, j), the time of service at store j is equal to the time of service at store i plus the driving time from store i to j and service time at store j. Additionally, this constraints forces $w_{ik} = 0$ whenever customer i is not visited in route k. Constraint (6) ensures that the time of service at a certain store falls between the time window for that store. Constraint (7) also enforces time windows, but for the depot exit (v_0 and the depot entrance v_{n+1}. Constraint (8) restricts the sum of the demand of all stores visited in a route to be less than the capacity of the vehicle driving that route.

3.2.1 Exact algorithms for the VRPTW

Over time, several exact methods have been proposed to solve the VRPTW. Some recent reviews of these methods have been reported in 2000 [6], in 2008 [13] and in 2012 [1]. These reviews show the most successful exact methods, at the time of writing, are based on the set partitioning formulation, such as the method reported in [9]. The key factor of success is the effective combination of the set partitioning formulation with families of cuts into column generation based algorithms [1]. Unfortunately, the number of cases in which the problem could be solved is limited. Whether a problem can be solved depends on, among other things, the number of customers and the width of each of the time windows. The width of a time window is the time between the earliest and latest allowed unloading time. Exact methods seem to be limited to small-medium sized problems, even with large amounts of computation time. The retailer VRP is of medium size, ranging from 39 to 102 customers per decentral bakery. Because of the time-complexity of the VRPTW, most research has been dedicated to proposing new heuristics rather than exact methods. Instead of solving the problem optimally, heuristics aim to approximate the optimal solution as best as possible. The goal of this research is to compare the transportation costs of the retailer's distribution network in the current situation with that in the proposed situation. A heuristic that is able to find a good approximation of the optimal solution will therefore be sufficient. The wide time windows, the practical features of the retailer VRP together with the limited accessibility to solvers that may be capable of solving these problems. Therefore, instead of using exact methods, heuristics will be used to approximate the routing for each decentral bakery.

3.3 Heuristics for the VRPTW

Heuristics are often used to solve time-complex problems that are hard to solve using exact methods. Two types of heuristics can be differentiated. The first type of heuristics are the constructive heuristics. Constructive heuristics start from nothing and incrementally add a 'building block' to the solution. In the VRP this translates to determining at every iteration, which store to visit next. An example of a constructive heuristic is the 'nearest neighbour heuristic'. In this heuristic, every iteration, the next store to visit is the one closest to the current one. This process is repeated until no unassigned stores are left. The second type of heuristics are the local search heuristics. These heuristics start with an initial solution and evaluate, each iteration, the 'neighbourhood' of the current solution. The neighbourhood is defined differently per heuristic. To provide an example, consider a solution consisting out of stores assigned to routes. In this example the neighbourhood of that solution is defined by changing, for one store, the route it will be visited in. Following this rule, the neighbourhood of the solution consists out of all solutions in which one store is assigned to a different route compared to original solution. Once the neighbourhood of a solution is determined, the local search heuristic selects the best solution out of that neighbourhood. This solution is referred to as the candidate solution. The candidate solution is compared with the current best solution and, if it is better, will be marked as the new best solution. Now starting with the candidate solution, this process is repeated until a local optimum is reached.

3.3.1 Heuristic evaluation

Heuristics can be evaluated using the framework reported in [5]. According to this framework heuristics are evaluated on the following four attributes:

1. **Accuracy:** Measures the degree of difference of a heuristic solution value from the optimal value.
2. **Speed:** The required computation time of a heuristic.
3. **Simplicity:** The degree of difficulty to understand and to code the heuristic.
4. **Flexibility:** The degree to which the heuristics could accommodate the various side constraints encountered in a majority of real-life applications.

Firstly, the accuracy of a heuristic is assessed by comparing its solution value with the current best known value. However, analyzing heuristic results is fraught with difficulties. Authors often report results that are the result of the best combination of starting solutions and algorithmic parameters [5] [2]. Another misrepresentation of the the accuracy criteria is the consistency of which the heuristic can produce good quality results [5]. It is reported that users tend to prefer a heuristic that performs well all the time rather than one that may perform even better most of the time, but rather poorly other occasions [5].

Secondly, the computation time of a heuristic is often a trade-off with its accuracy. The preference of the user in terms of speed is also highly dependent on the context in which the algorithm is used. If the algorithm needs to run daily in order to determine the route planning for that day, a computation time of maximum 10 minutes might be preferred at the cost of some accuracy. When determining the fleet size and the route planning for a number of years, a high accuracy might be preferred. Generally, when making long term strategic business decisions, speed would be less important than accuracy for most users.

Thirdly, some heuristics are rarely selected because they are too difficult to understand and to code. Sufficient information about the heuristic should be provided such that a reasonably skilled programmer can apply the heuristic. Part of the simplicity criteria is the robustness of a heuristic. A robust heuristic should work properly, even if not every single detail is implemented. Furthermore, the number of parameters is reported to be a good indicator of the simplicity of a heuristic.

As a rule of thumb can be stated that fewer parameters increases the simplicity of the heuristic.

Fourthly, the flexibility of a heuristic is measured in the degree to which it can facilitate different side-constraints encountered in real-life situations. Such side-constraints in the context of the retailer are the time windows at the stores and the maximum working duration of the truck drivers. Heuristics often facilitate such side constraints by making use of two objective functions [5]. The first objective function computes the routing costs of a solution. The second objective function reflects the degree to which the solution violates the side constraints. Usually, as the search progresses, the weight of a side-constraint violation is adjusted using penalty parameters. These penalty parameters self-adjust based on whether the previously found solution was feasible or infeasible.

3.3.2 Heuristic selection

Local search methods tend to arrive at a local optimum and the quality of this local optimum can be very poor compared to the optimal solution [19]. To escape this local optimum, an escape strategy can be employed. Methods that utilize such an escape strategy are classified as *metaheuristics* [18]. Several relevant reviews have been published concerning the available metaheuristics in 2002 [5], 2005 [3] and in 2010 [11]. The latter review focuses on the performance of heuristics on large scale problems, while the first two focus on the the performance of heuristics on the medium sized Solomon benchmark instances [17]. While some heuristics perform systematically better in terms of solution quality, each run with the heuristics seems to give a different result. In addition to this, among the top performing heuristic, the improvement in solution quality is only minor. As reported in [3], the heuristics developed at that time may require significant computation time. More recent literature focuses on large problem instances including for example 400 customers. The reason that some heuristics take such a long time is due to the nature of their neighbourhoods. Neighbourhoods, referred to as 'large neighbourhoods' [11], can become exponentially large as the number of stores grows, causing the computation time to become very large per iteration. In more recent literature, many new metaheuristics have been proposed that aim to retain or even slightly improve the solution quality, while decreasing the computation time. An example of recently proposed heuristic is the one reported in [16]. In this research the granular neighbourhoods described in [19] are implemented to reduce neighbourhood size, combined with the diversification principles of the tabu search heuristic proposed in [4]. Tabu search is a widely used local search method that utilizes a 'tabu list' as a local optimum escape mechanism. This list is used as an escape strategy and decreases the chance that the heuristics gets stuck in a local optimum. Once a candidate solution has been picked from the neighbourhood, the attributes identifying the move away from the previous solution are stored in the tabu list. For example, if store 3 is removed from route 4 to form the best neighbourhood solution, the attributes (3,4) are listed in the tabu list. This means that solutions that the reinsertion of store 3 in route 4 is for a predetermined number of iterations unless some aspiration criterion is satisfied. An example of an aspiration criterion is that the reinsertion of certain attributes is allowed if the resulting solution is better than the current best solution.

For this research it is important that the difference between the current and proposed situation becomes evident. For this reason it is important that the selected heuristic should give a realistic estimate of the transportation costs that can be expected in practice. However, the heuristic will only be used to generate the results to determine the estimate for the transportation costs in this research. The final routing schemes will be generated by the retailer using their licensed routing software. For this reason, and the remain within the time scope, simplicity is also an important factor. Keeping this in mind, the heuristic that will be adapted to fit the case of the retailer will be the one reported in [4]. This heuristic has shown to perform well in terms of solution quality in literature, but requires significant computation time for large scale problems. The heuristic is easily adaptable to the practical features and constraints of the retailer VRP. In order to decrease the computation time and therefore the number of runs that can be executed per case, a parallel programming approach will be used to utilize the multi-core properties of modern CPU's.

3.4 Conclusion

Looking at the aspects of the retailer VRP, it becomes clear that it can roughly be generalized as a Vehicle Routing Problem with Time Windows (VRPTW). Exact methods will not be applied in this research due to them being able to only solve problems with a small to medium size and the limited accessibility to solvers that can solve these problems. For this reason, a decision has been made to look for heuristics. Instead of solving the problem to optimality, heuristics approximate the optimal routing scheme. After weighing different aspects concerning the quality of heuristics, the unified TS algorithm reported in [4] is selected to use for this research. In the next chapter, the unified TS algorithm[4] is adapted to fit the case of the retailer.

Chapter 4

Model Design

The goal of this model is to determine, per decentral bakery, a good approximation of the optimal routing scheme to replenish the stores on a daily basis. Per decentral bakery, the weekly routing schemes for both the current situation and proposed situation have to be designed. The model used to determine the routing schemes for the retailer is based on the tabu search metaheuristic. The tabu search heuristic is a local search metaheuristic. This means that the heuristic searches all available solutions by moving, at every iteration, from the current solution to the best solution in its neighbourhood. How the heuristic determines what the best solution is and how the neighbourhood is generated will be explained, in-depth, later in this chapter. A tabu search heuristic is characterised by its use of a tabu list during the search process. The tabu list servers as a anti-cycling mechanism. This means that it aims to prevent the search process to get stuck in only one part of the solution space.

The model that is applied in this research consists out of three parts (figure 4.1): (i) generating the initial solution, (ii) tabu search and (iii) post optimization. As mentioned in the previous chapter, the model in this research is based on the unified tabu search algorithm [4]. Parts (i) and (ii) of the model used in this research are based on the original unified tabu search algorithm. Due to the complexity of the post-optimization heuristic used in the unified tabu search heuristic [4], it was less suitable for use in this research. Instead, to optimize the route duration, the forward slack method [15] is used as post-optimization heuristic.

The first section of this chapter contains a brief description of the required input data for the model. A more in depth description of the input data is provided in Chapter 5. Section 4.2 gives a general overview of the model in terms of its constraints, neighbourhood structure and diversification mechanisms.

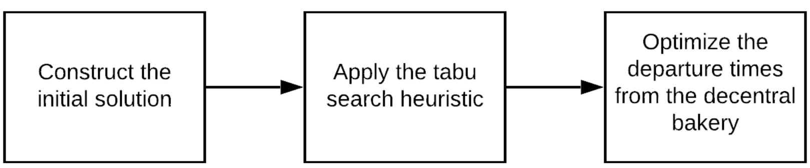

Figure 4.1. The model design

4.1 List of notations

- Indice i represents the store number with i.
- e_i represents the earliest unloading time at store number i.
- l_i represents the latest unloading time at store number i.
- $e1_i, e2_i$ represent the earliest arrival time of the first and second time window of store number i, respectively
- $l1_i, l2_i$ represent the earliest arrival time of the first and second time window of store number i, respectively
- a_i represents the arrival time at store number i.
- s represents a solution consisting out of the routes to replenish all allocated-stores.
- k represents a route with number k.
- m represents the maximum allowed number of replenishment routes.
- $B(s)$ represents the set of attributes (i,k) of a solution. (i,k): Store number i is visited in route k], with (k = 1,...,m).]
- Θ represents the tabu list length.
- c(s) represents the transportation costs of solution s, containing the fuel, driver and vehicle costs.
- f(s) represents the constraint violation penalty
- q(s) represents the violation of load constraints in solution s.
- d(s) represents the violation of duration constraints in solution s.
- v(s) represents the violation of time window constraints in solution s.
- α, β and γ represent the scaling factors for the load, duration and time window violation penalty, respectively.
- δ represents the degree of adjustment made to the scaling factors after every iteration.
- p(s) represents the penalty function that penalizes the solution based on the number of times the attributes that make up the solution were added during the search process.
- λ is used to control the severity of the penalty function p(s).
- z represents the sequence in which the stores are visited during a route.
- b represents the last visit in a route, this is always the decentral bakery.

4.2 Required data and general overview

To apply the heuristic, information about each store and the routes between these stores and their allocated decentral bakery is required. For each store the following information is required:

- A geographic location.
- The average demand per day measured in metal racks in case trucks are used and stacks of crates in case sprinters are used.
- The time it takes to unload the containers with pastries from the truck or sprinter (the unloading time).
- The time window(s) consisting out of the earliest and latest unloading time(s).

In practice, two types of stores can be differentiated based on their time windows. The first type of store can be accessed throughout the night with either a key or a security code. The second type of store cannot be accessed during the night using a key or code, and thus has to be visited within specific time windows. These time windows usually dictate that the store has to be visited in the morning, not earlier than for example 7:00. Some stores have multiple time windows, for example, one time window in the evening before closing time, and one time window in the morning, when the store opens. How the model handles these multiple time windows is explained in section 4.2.1.

Furthermore, information about the routes between every store and between every store and its allocated decentral bakery is required. The following three characteristics of the routes between all stores and bakeries are relevant:

- Travel distance
- Travel duration
- Fuel costs

The travel duration and travel distances used by the model are based on traveling via real roads, not Euclidean distances. The actual distance and duration matrix will be generated using a third party distance matrix generation software. The data collection process will be described in depth in Chapter 5. Figure 4.2 shows a summary of the information required per store and route.

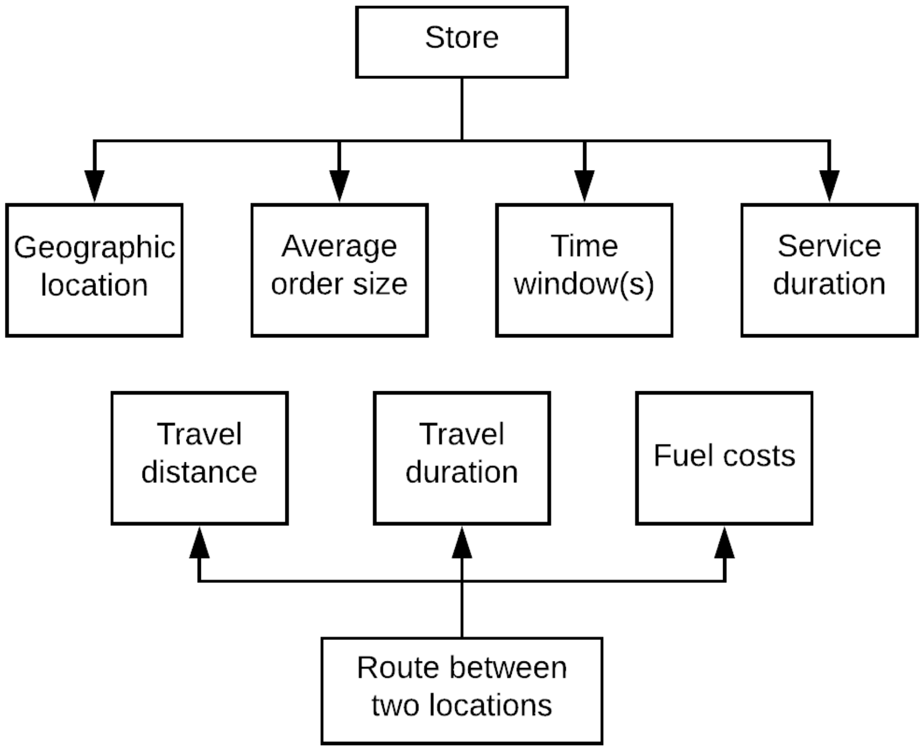

Figure 4.2. The required information per store and route

4.2.1 Constraints and time windows

The resulting routing scheme after applying the model should satisfy the following constraints:

1. Every route starts and ends at the decentral bakery.
2. Every store is allocated to exactly one route.
3. The total load of a replenishment route does not exceed the vehicle capacity.
4. The total duration of a replenishment route does not exceed the maximum duration.
5. Unloading at a store begins within one of the time windows of the respective store and every replenishment route ends at the decentral bakery within the total delivery time window

Stores are identified by a number i. A time window for store i consists out of two parts: the earliest unloading time (e_i) and the latest unloading time (l_i). If the driver arrives at a store before the earliest unloading time, the time window constraint is not violated, but the driver has to wait until the earliest unloading time. In this case, a waiting time is incurred. If the driver arrives at a store with two time windows, the following procedure is used to determine with which time window is calculated whether the vehicle is on time, too early or too late:

if $(a_i \leq (\frac{e2_i - l1_i}{2})$ **then**
  $[e_i, l_i] = [e1_i, l1_i]$;
else
  $[e_i, l_i] = [e2_i, l2_i]$;
end

where:

- $[e1_i, l1_i]$, represents the first time window of store number i.
- $[e2_i, l2_i]$, represents the second time window of store number i.
- a_i, represents the arrival time at store number i.
- Both time windows are defined such that $e1_i \leq l1_i \leq e2_i \leq l2_i$

4.2.2 Neighbourhood structure and tabu status

Tabu search is a local search metaheuristic and therefore explores the solution space by moving at each iteration from the current solution s to the best solution in its neighbourhood. The neighbourhood of a solution are all the solutions that can be generated by performing a single adjustment to the solution. To restrict the number of routes allowed by the model in a single solution, a maximum m is determined. The value of m is determined beforehand using a trial run. The details of the trail run are discussed in Chapter 6.

In the model, a solution s is defined by its set of attributes $B(s) = [(i, k)$: Store number i is visited in route $k]$, with (k = 1,...,m) [4]. The neighbourhood of a solution consists out of all solutions in which an attribute (i, k) is removed, and replaced with an attribute (i, k'), where $k \neq k'$. An example of the neighbourhood of a solution is given in Figure 4.3

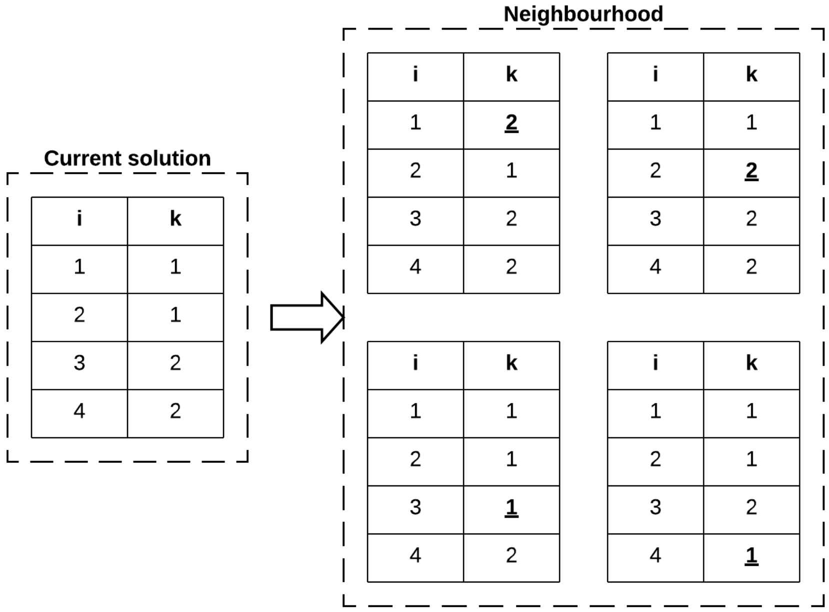

Figure 4.3. Example of the neighbourhood of a solution.

The route from which the store is removed is reconnected by simply linking the predecessor and successor of the store. Store number i is *inserted* in route number k' such that the transportation costs of that route are minimized [4]. An example of an insertion is given in Figure 4.4.

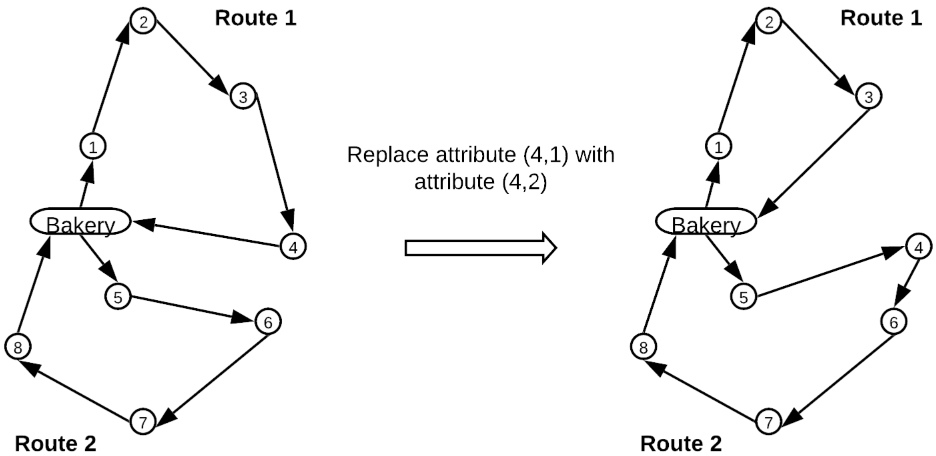

Figure 4.4. An example of the insertion of store 4 in route 2

If for example, removing store number 4 from route number 5 results in the best solution in the neighbourhood, a tabu status is applied to the attribute $(4, 5)$. This tabu status forbids its reinsertion for the next Θ iterations, where Θ represents the tabu list length. The tabu list length has to be determined beforehand.

This process is explained in Chapter 6. The tabu status can be revoked through an aspiration criterion. In this case the aspiration criterion allows the reinsertion of certain attributes if the resulting solution is better than the current best solution.

4.2.3 Cost and penalty functions

To evaluate a solution, a cost function is used. An important feature of the unified tabu search heuristic is the possibility of exploring infeasible solutions during the search process [4]. For a solution s, the travel costs are calculated using the cost function $c(s)$. Similar to the transportation costs, this cost function is comprised out of three parts:

- fuel costs: Calculated based on the sum of the fuel costs of all routes in the solution
- driver costs: Take the sum of the driving time, (un)loading and waiting time of every route in the solution and multiply it by the hourly rate of the driver.
- vehicle costs: The costs per vehicle multiplied by the maximum number between the number of evening and morning routes.

The cost function of solution s is therefore equal to: $c(s) = \text{fuel costs} + \text{driver costs} + \text{vehicle costs}$.

Concerning the daily vehicle costs, normally these are simply the number of routes driven per day multiplied by the costs per vehicle per day. In the case of the retailer the vehicles used during the evening routes can also be used during the morning. The actual number of vehicles needed is therefore the maximum number between the number of evening and morning routes. In case the route includes a store that has a time window that dictates the store can only be visited in the morning, the route is classified as a morning route. If no such store is visited, the route is classified as an evening route.

Another important feature of this heuristic is the fact that it allows exploration of infeasible solutions [4]. Infeasible solutions in this context are solutions that do not satisfy one or more of the load, duration or time window constraints. Although exploring infeasible solutions is allowed, a constraint violation penalty is applied. The cost function including the constraint violation penalty is represented by $f(s)$. $f(s)$ consists out of three parts [4]:

- $q(s)$ = the violation of load constraints in solution s.
- $d(s)$ = the violation of duration constraints in solution s.
- $v(s)$ = the violation of time window constraints in solution s.

The total violation of the load and duration constraints are calculated with respect to the maximum capacity and duration. The time window is violated only if the vehicle arrives later than the latest unloading time of the respective store. In that case the time window violation is calculated using the latest unloading time of the applicable time window at the store. In case the store has two time windows the applicable time window is determined using the procedure explained in Section 4.2.1. To facilitate the exploration of the search space, each violation penalty is scaled using a scaling factor [4]. Each violation penalty has its own scaling factor. These scaling factors are α, β and γ for the load, duration and time window violation penalty, respectively. All these parameters start at value 1. However, during the course of the algorithm, these parameters self-adjust based on whether the constraints they scale are violated. The degree to which they adjust is represented by δ. If the constraint is satisfied its scaling factor is adjusted by dividing the scaling factor by $1+\delta$. If it is violated the scaling factor is multiplied by $1+\delta$. To give an example, imagine a solution that does not violate the duration or load constraints, but violates one time window in one of its replenishment routes. Let δ be 0.2. It is the first iteration so all parameters α, β and γ equal 1. α and β are adjusted to be: $1/(1 + 0.2) = 0.833$. Since the time window constraint is violated γ is adjusted to be $(1 * (1 + 0.2) = 0.833$. By dynamically adjusting these factors, the search process is facilitated and is especially useful in tightly constrained search spaces [4]. How δ is determined beforehand is explained in Chapter 6. As mentioned before, the model used in this research utilizes anti-cycling and diversification mechanics. Next to the previously explained tabu list, the model employs penalty functions to control the degree of diversification in the search process. This penalty function penalizes the solution based on the number of times the attributes (i, k) that make up the solution were added the current solution during the search process. The penalty function is as follows [4]:

$$p(s) = \lambda c(s)\sqrt{nm} \sum_{(i,k)\in B(s)} \rho_{i,k}$$

where:

- λ is determined beforehand to control the intensity of the diversification.
- c(s) represents the transportation costs of solution s.
- n represents the number of stores allocated to the decentral bakery.
- m represents the maximum number of routes allowed in a solution.

The scaling factor λ is determined beforehand by performing a sensitivity analysis. This analysis is described in Chapter 6. The higher λ is, the higher the diversification during the search process [4]. The scaling factor $c(s)\sqrt{nm}$ is part of the equation to scale the penalty according to the transportation costs, the number of routes and the number of stores. When evaluating any solution in the neighbourhood of solution s, $p(s)$ is included in the evaluation. Therefore, any neighbourhood solution is evaluated based on its $f(s) + p(s)$ value.

4.3 The unified tabu search heuristic

The algorithm consists out of three parts: constructing the initial solution, performing the iterations and post-optimization of the departure times. During this section each part of the algorithm will be explained using examples and visualizations where possible.

4.3.1 Initial solution

The first step in the heuristic is to construct the initial solution from which the unified tabu search heuristic starts. The manner to which the initial solution is constructed is the same as reported in [4]. The first step in constructing the initial solution is to order all stores based on the angle they make with the decentral bakery. For clarification, consider the example of a decentral bakery with 8 allocated stores in figure 4.5.

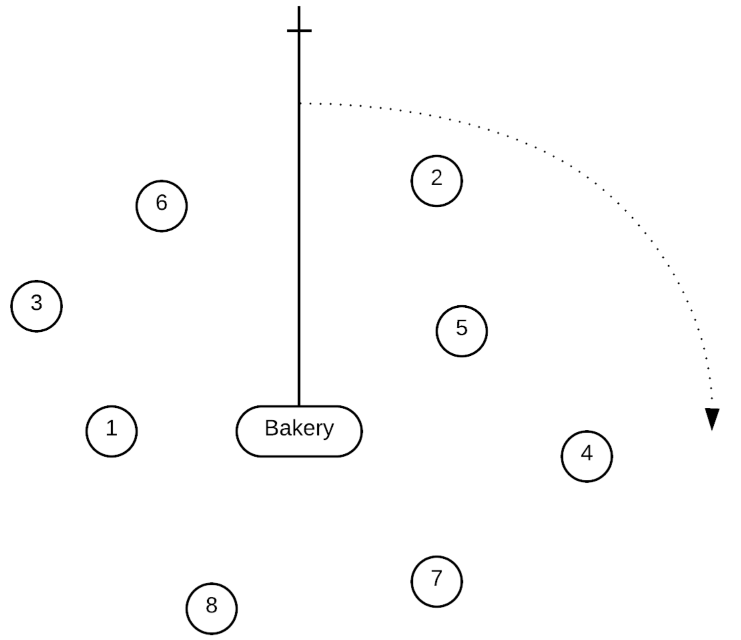

Figure 4.5. A decentral bakery with 8 allocated stores

Starting with a conceptual line in Figure 4.5 straight upwards, the stores are ordered by moving the conceptual line clockwise in a full circle. The stores are ordered based on which order they 'touch' the conceptual line as it moves around. Following this step, the order of stores in the example displayed in figure 4.5 is as reported in table 4.1

Table 4.1: *Example of store ordering using the conceptual line*

Order	**Store nr. (i)**
1	2
2	5
3	4
4	7
5	8
6	1
7	3
8	6

The next step is to determine in which sequence the stores are picked to place in the routes within the initial solution ($s_{initial}$). First a random number between 1 and the number of stores has to be drawn. This number is referred to as RN. The first store in the sequence is the store with that has order RN. From this store, one cycles through the stores based on the way they were previously ordered, starting at the store with order RN and ending at the store before the store with order RN. To give an example consider the example from Figure 4.5 and Table 4.1. If the random number drawn is 4, the sequence for the initial solution would be the stores with order [4,5,6,7,8,1,2,3]. This corresponds to a store sequence of store number [7,8,1,3,6,2,5,4]. The store sequence for the initial solution from the example with RN = 4 is displayed in 4.2.

Table 4.2: *Example of the cycled store sequence*

Sequence	**Store (i)**
1	7
2	8
3	1
4	3
5	6
6	2
7	5
8	4

The goal of drawing the random number is to start from a randomly cycled initial solution every time the algorithm is started. Since not the whole solution space is explored during this algorithm, a change in initial solution could give a different final result. To construct the initial solution using this sequence the following procedure is performed [4]:

1. Set Route number (k) = 1 ;
2. Following the sequence of stores, perform the following steps for every store in the sequence:
   - If the insertion of the store into route k results in the violation of load or duration constraints, increase k by one if $(k + 1) \leq m$. If k is already at the maximum allowed level (m), ignore the constraint violation and place it into route number k anyway.
   - Insert the store into route number k so as to minimize the increase in total travel time costs of route number k .

A slight simplification of this procedure is visualized in Figure 4.6

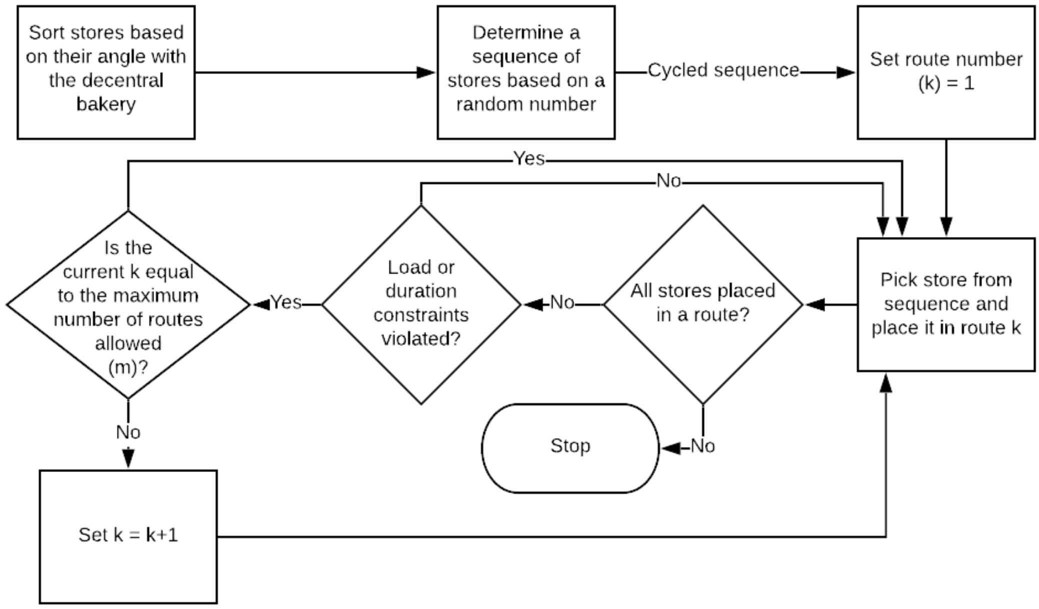

Figure 4.6. The procedure used to generate the initial solution

The insertion of a store between two successive stores can only be performed such that $e_{Predecessor} \leq e_{Storenr.} \leq e_{Successor}$. The initial solution should consist out of a maximum of m routes. Routes with number $\{1, ...m-1\}$ should satisfy the load and duration constraints, but may not satisfy the time window constraint. Route number m may violate any of these constraints. The departure time from the decentral bakery is equal to the start of the total delivery window, which is at 18:00. If departing from this time causes a waiting time at the first visit in that route, the departure time will be moved forward so that no waiting time at the first visit occurs. For example, consider a route in which the first visited store has an earliest arrival time of 7:00 and driving from the bakery to this store required 30 minutes. In this case, the departure time is set at 6:30 in the morning, so that no waiting time occurs. This method does not, by itself, result in the optimal arrival times at every store. Therefore, the final step of the algorithm is to optimize the departure times once the tabu search heuristic has been applied.

4.3.2 Tabu search iterations

The algorithm starts with the initial solution and chooses, each iteration, the best solution in the neighbourhood of the current solution. The best solution in the neighbourhood is referred to as the *candidate solution* ($s_{candidate}$). Once found, this solution is compared against the best solution (s_{best}). If $c(s_{candidate} \leq s_{best}$ and $s_{candidate}$ is feasible, the candidate solution is marked as new best solution. After the comparison the candidate solution is marked as the current solution ($s_{current}$). This process is repeated for η iterations. Each iteration, the parameters α, β and γ are updated as described in Section 4.2.3. In addition to this, every iteration the tabu status is updated as described in Section 4.2.2. The tabu search iterations [4] are summarized as follows:

1. Set $\alpha = 1, \beta = 1$ and $\gamma = 1$;
2. If the initial solution ($s_{initial}$ is feasible, set $s_{best} = s_{initial}$ and $c(s_{best}) = c(s_{initial})$, if $s_{initial}$ is not feasible set $c(s_{best}) = \infty$;
3. Set $s_{current} = s_{initial}$ 4. Repeat for η iterations:
   - Choose a neighbourhood solution in $N(s_{current})$ that minimizes $f(s \in N(s_{current}) + p(s \in N(s_{current})$ and is not tabu or satisfies the aspiration criterion.
   - Set this solution as $s_{candidate}$
   - If $s_{candidate}$ is feasible and $s_{candidate}) < c(s_{best})$, set $s_{best} = s_{candidate}$ and $c(s_{best}) = c(s_{candidate})$.
   - Update the tabu list.
   - Update α, β and γ according to the constraint violation of $s_{candidate}$
   - Set $s_{current} = s_{candidate}$

4.3.3 Optimization of departure times

Once all iterations have been performed, the resulting routing scheme is post-optimized in the final step of this algorithm. In [5] an adaptation [12] of the GENIUS [10] insertion and post-optimization procedure is applied. For reasons mentioned before, another post-optimization method is used for this research. The selected post-optimization method is the 'forward slack' method [15]. By applying this method, the route duration is minimized by shifting the departure from the bakery forward as far as possible, without violating the time window constraints.

The forward slack method is applied to every route in the solution. Let $z = \{0, 1, 2, ..., b\}$ be the sequence in which the stores are visited in the route. $z = 0$ and $z = b$ represent the first and last visit in the route, the decentral bakery. For all stores (not the bakery) visited in the route, the following needs to be calculated/determined:

- The time at which unloading at the store in sequence z starts, $unload_z$
- The time at which the driver arrives at the store in sequence z , $arrive_z$
- The waiting time at store in sequence z , $wait_z$
- The earliest and latest arrival time at the store in sequence z ,$[e_z, l_z]$

Once calculated, the slack per store visited has to be determined. The slack variables for all stores with sequence $z = \{1, 2, ..., b-1\}$ are calculated as follows:

$$Slack_z = max(l_z - unload_z, 0) + \sum_{x=1}^{z} wait_x$$

Subsequently, the improved departure time can be calculated by adding the minimum slack value to the current departure time. To illustrate this method, consider the route in table 4.3. All numerical values in this table are in minutes.

Table 4.3: *Example of a route with calculated slack variables*

z	[e;l]	Arrive	Unload	Wait	Departure time	Slack
0	[0;200]				10	
1	[20;60]	20	20	0	26	43
2	[50;70]	36	50	14	57	37
3	[80;90]	67	80	13	88	40
b	[0;200]	98				

In the example, when adding the minimum slack value (37) to the departure time from the decentral bakery, one would get the route as shown in table 4.4. All numerical values in this table are in minutes.

Table 4.4: *Example of a route with improved departure time*

z	[e;l]	Arrive	Unload	Wait	Departure time	Slack
0	[0;200]				44	
1	[20;60]	54	54	0	60	6
2	[50;70]	70	70	0	77	0
3	[80;90]	87	87	0	95	3
b	[0;200]	105				

In the example, the previous routing in Table 4.3 has a total route duration of 91 minutes, while the improved routing scheme in Table 4.4 a total route duration of 61 minutes.

Applying this method does not always result in improved route duration. For example during evening routes, all stores can usually be entered using a key or code. Therefore, the time window spans the total delivery window. Due to the nature of this method, such a route would be shifted forward in time, resulting in the maximum feasible departure time from the decentral bakery, without any route duration improvement. This would be disadvantageous in practice, since the vehicles used for the evening routes will also be used for the morning routes. If one where to shift the route forward in time as far as possible, the vehicles would not be available. To prevent this from happening, the routing scheme is only changed if any improvement in total route duration is achieved.

4.4 Conclusion

In this chapter, the unified TS heuristic reported in [4] is designed to fit the case of the retailer. First a general overview, together with the required information has been provided. Afterwards, the three parts of the algorithm have been described: the construction of the initial solution, the application of the tabu search heuristic and the optimization of departure times. Before applying the model to the case, three parameters need to be predetermined by performing a sensitivity analysis: Θ (tabu-list length), δ (constraint violation weight adjustment) and λ (degree of diversification). In the end, these variables all influence the degree to which the total solution space is explored. How the sensitivity analysis is performed is explained in Chapter 6.

The next step in this research is to collect and, if needed, transform the required data. Afterwards this data is used with this model to determine the routing schemes during the week and on Saturday. Next to describing the required input data, the next chapter describes the data collection process and the data transformation process if needed.

Chapter 5

Data Collection

With the model designed, the next step is to gather the required input data. The goal of this chapter is to describe the data gathering and preparation process for the input data of the model. First, an overview of the required data will be given. After the overview, the data collection process for all the required data is described. The data required for the model is displayed in this chapter when possible. Due to their size, some of the data tables are placed in the Appendix.

5.1 Overview of required data

Following the model design in the previous chapter, the following data is required:

- For each store:
  - The geographic location.
  - The decentral bakery to which it is allocated.
  - Demand represented by the average order volume in both metal racks and stacks of crates.
  - The time windows.
  - The service duration/unloading time.
- For the sprinter and the truck:
  - The vehicle costs including maintenance, insurance and lease/depreciation costs.
  - The average fuel costs per kilometer.
  - The driver's hourly salary.
- A cost matrix containing:
  - The fuel costs between every relevant location.
  - The travel duration between every relevant location.

5.2 The stores

The geographic location and decentral bakery allocation of each store could directly be extracted from the retailer's internal database and are both reported in Appendix B. Concerning the demand per store, two averages are used to represent the daily order size. Per store, one average for the weekdays and one average for Saturday is used. Naturally, since demand in the model is deterministic, it is desirable to take the average of as much demand data as possible. Unfortunately, for technical reasons, it is currently impossible to extract more than a week worth of demand data from the retailer's database. During the year, the weekly demand seems to be uniform, with a small number of exceptions caused by holidays. As a result, in conversation with the retailer a decision is made to use a representative week as the basis for the average order size used in the model. Two routing schemes per decentral bakery and per situation are generated. For this purpose, the following demand data is required:

- The average number of metal racks ordered per store on a weekday.
- The average number of stacks of crates ordered per store on a weekday.
- The average number of metal racks ordered per store on Saturday.
- The average number of stacks of crates ordered per store on Saturday.

The number of containers ordered per store is calculated using the information and equation described in Chapter 2. The data concerning the average order size per store is reported in Appendix C.

As mentioned in the previous chapters, most stores have one or multiple time windows. Some stores do not have time windows because they can be accessed outside opening times using a key or a security code. The time windows for each store have been requested and collected from all six decentral bakeries and reported in a single document. In case a store has two time windows, the time window in the evening and morning are represented by time window 1 and time window 2, respectively. The time window(s) for every store are shown in Appendix B.

The unloading time per store can be estimated using an equation commonly used by the retailer. This equation combines a fixed component and a variable component that is dependent on the amount of metal racks or stacks of crates that have to be unloaded. The equation is as follows:

$$\text{Service duration} = 6 + 1.4 * \text{Number of containers}$$

Using the average amount of metal racks ordered per store during the weekdays and on Saturday, the service duration at every store during both the weekdays and the weekend can be calculated for the current situation. Similarly, using the average amount of stacks of crates ordered per store, the service duration at every store during both the weekdays and the weekend can be calculated for the proposed situation. Loading and unloading at the start and end of every replenishment route takes on average 30 minutes in both the current and proposed situation. Therefore, the duration of every replenishment route includes a loading/unloading time at the decentral bakery of $2 * 30 = 60$ minutes.

5.3 Vehicle, fuel and driver costs

The vehicle costs consist out of three cost factors: maintenance costs, insurance costs and lease/depreciation costs. The maintenance and insurance costs factors for the truck are based on the averages of the year 2018, which as declared by the retailer is representative for an average year. Table D.5 shows the insurance and maintenance costs incurred for the fleet of trucks in 2018.

One sees in Table D.5 that decentral bakery in Helmond is missing. The decentral bakery in Helmond outsources their replenishment routes and therefore has no trucks of their own and does not contribute to the total maintenance and insurance costs incurred. Using the total costs in 2018 and the total fleet size, the average maintenance and insurance costs per truck can be calculated. For the sprinter, no historic data is available. In conversation with the monitoring and control department at the bakeries, is determined that the maintenance and insurance costs of the sprinter will approximately be equal to 80% of the maintenance and insurance costs of the truck. The third and final component of the vehicle costs are the lease/depreciation costs. Based on the information provided by monitoring and control department at the bakeries, the yearly costs for lease/depreciation for the truck and for the sprinter are calculated. The trucks are more expensive per vehicle, but because trucks have a longer average lifespan than the sprinters, the yearly costs are actually lower for the truck compared to the sprinters. To summarize, the vehicle costs for the truck and sprinter are shown in Table D.6.

Secondly, the fuel cost per kilometer for both type of vehicles has to be determined. Fortunately, accurate data on the amount of kilometers driven per vehicle in 2018, and the fuel costs per decentral bakery in 2018 is available. Using this information, the fuel costs are calculated. No historic data for the sprinter is know, but it is known that the trucks consume 1 liter of diesel per 3 kilometers on average and the sprinters consume 1 liter per 6 kilometers on average. Using this information, the fuel cost per kilometer for the sprinter can be calculated.

Finally, the driver costs can also be derived from current employee salary data. The advantage in terms of costs of the sprinter over the truck is that the sprinter can be driven by anyone with a BE drivers license. The reasoning is that while the truck is driven by more expensive truck drivers, the sprinters can be driven by decentral bakery production employees.

The total driver costs are different for each decentral bakery and consist out of salaries of both own and externally employed drivers. Since the retailer's own drivers have a different salary then the external drivers, a weighed average hourly salary has to be calculated. The weighed average is based on the amount of hours driven by both type of drivers. Similarly, the weighed average hourly salary of a truck driver at the other decentral bakeries except the decentral bakery in Helmond are calculated. Since Helmond outsources all their transportation, no historic data on the hourly salary of their drivers is available. Instead, the average hourly salary for the truck drivers in Helmond is the average of the hourly salaries of the truck drivers of the other bakeries. For the drivers of the sprinter, the average hourly salary is based on the average salary of a production employee in a decentral bakery. Therefore, the hourly salary of a sprinter driver is calculated in the same manner as the truck driver, but with the amount of hours worked and hourly salaries of the own and external production employees in each decentral bakery. Table D.3 shows the hourly salaries of the truck and sprinter drivers per decentral bakery.

5.4 Cost matrix

The cost matrix is the basis of the cost calculation for a routing scheme. In this matrix, the fuel costs and the driving duration between every relevant location are reported. To get an as accurate result as possible, the input data should represent reality as accurately as possible. For this reason, it is important that the actual distances and driving duration over the known roads are used between every relevant duration. Since the store to decentral bakery allocation is already known, only the information about the routes between every store assigned to the same decentral bakery is required. Similarly, only the information about the routes between every store and its assigned decentral bakery, not every decentral bakery, is required. Unfortunately, such an up-to-date matrix is currently not available at the retailer. Therefore, the data has to be generated using third party distance matrix utility software. The duration that is given as output, when using the selected software, is based on the travel duration when driving with a regular sized car. The sprinter is not expected to be significantly faster than a truck on average. However, both are on average slower than a regular sized car. Therefore, based on a number of data samples concerning the driving duration in practice with a truck, and in conversation with the retailer, the average driving speed when driving with a truck and sprinter was determined to be slower. Based on this driving speed and the distances between every relevant location, the driving duration when driving with a truck and sprinter is determined. Additionally, using these distances and the fuel costs per kilometer for the truck and for the sprinter, the fuel costs when driving between every relevant location is calculated. Due to their size, the cost matrices are too large to be included in this report and can for this reason be accessed on request.

Chapter 6

Results

The goal of this chapter is to provide an in-depth description of the results of the algorithm selected for this research. The first part of this chapter provides an in-depth description of the test design for every decentral bakery. In the second part of this chapter, the results are calculated and presented in a manner that allows a thorough analysis to be performed.

6.1 Test design, verification and sensitivity analysis

A routing scheme has to be generated for every bakery, such that the transportation costs are minimized and all constraints are satisfied. Due to the significant difference in order size per store, every bakery requires two routing schemes, for the weekdays (Sunday till Friday) and for Saturdays. As described in Chapter 2, the bakeries in Dordrecht and Almere also have a different routing scheme for the Sundays. However, the retailer expressed that the tests could be simplified by just assuming that the routing on Sunday would be the same as on the weekdays. For this reason, Sunday is in this model assumed to be the same as Monday till Friday. To recap, with the routing for a certain day is meant, the replenishment routes driven during the evening/night the day before and in the morning of the day itself. As a result, four different routing schemes have to be determined.

1. A routing scheme for the replenishment of stores during the weekdays in the current situation.
2. A routing scheme for the replenishment of stores during the weekdays in the proposed situation.
3. A routing scheme for the replenishment of stores on Saturday in the current situation.
4. A routing scheme for the replenishment of stores on Saturday in the proposed situation.

Since the retailer has six decentral bakeries, the total number of cases for which the model has to generate a routing scheme is equal to $6 * 4 = 24$.

As was mentioned before, the model requires three parameters to be determined beforehand, λ, δ and the tabu-list length Θ. This implies that, to generate a high quality routing scheme, a sensitivity analysis has to be performed. Since each decentral bakery has a different number of allocated stores, these parameters have to be determined for every decentral bakery. One of the disadvantages of the model is that performing an exhaustive sensitivity analysis on three variables is impossible, time-wise. This is a result of the significant time it takes to perform a single experiment (up to two days for Dordrecht). Instead of performing an exhaustive analysis, per decentral bakery a shortened analysis was performed. This analysis concerned case 1 and starts with all three parameters at a relatively high level. By decreasing each of the values one at a time, an estimate of the best parameter configuration is made. While some configurations perform consistently better, the quality of the solution seems to depend heavily on the quality of the initial solution. For this reason, the best parameter configuration changes based on the quality of the initial solution. After determining a good parameter configuration for a certain decentral bakery, multiple runs are performed for all cases to increase the quality of the routing scheme. This increases the likelihood of finding a good solution. Based on the number of routes generated in the solution found during the sensitivity analysis, the value of the maximum allowed number of routes (m) is determined. This is done in this manner since one knows that a feasible solution is possible with that number of routes.

The computation time required for the generation of the routing schemes for the four cases per decentral bakeries varies significantly based on the number of stores allocated to a decentral bakery. The model seems to provide good solutions when ran for 10,000 iterations. The computation time required to perform these iterations ranges from slightly less than two hours (Sneek) to almost two days (Dordrecht). To increase the likelihood of finding a good solution, each case is run 8 times. Doing so results in $8*4 = 32$ cases per decentral bakery. To greatly decrease the total computation time required, the algorithm is programmed in such a manner that allows for each case of a decentral bakery to be executed in parallel. Using the High Performance Computing Cluster of the University of Twente, all 32 cases for each decentral bakery can be run in parallel, making the total computation time significantly more manageable.

To confirm whether the model provides solutions that represent reality, the actual routing schemes for Zwolle is compared to the routing schemes generated for Zwolle in the current situation by the model. By comparing the costs and distances driven by both routing schemes is verified whether the model produces results that represent reality in a manner that is acceptable to the retailer.

6.2 Results of the VRPTW

This section contains the results of the VRPTW. The first and second subsection describe the results for the routing schemes on the weekdays and Saturdays, respectively. In these sections the fuel costs, driver costs and the required fleet size will be reported and discussed. The third section describes the resulting vehicles costs following the required fleet size per decentral bakery for both the weekdays and Saturdays. In the fourth section, an overall cost comparison is made to determine the difference in terms of transportation costs in both situations.

6.2.1 Replenishment routing for the Weekdays

The first and second case of each decentral bakery concerns the routing for the weekdays for the truck and sprinter, respectively. Figure E.2 and E.3 show the routing in the current and proposed situation for the weekdays. Note that while the routing in these pictures is displayed by straight lines, the actual replenishment routes follow the available roads. Therefore the displayed routing may be misleading to some extend. The fuel and driver costs of the corresponding routing schemes are reported in Table E.1 and E.2, respectively.

In Table E.1 and Figure E.4, one sees that the fuel costs in the proposed situation are significantly lower for all decentral bakeries compared to the current situation. Interesting to note is that the total distance driven only differs slightly, but the fuel costs are significantly different in both situations. Although the distance driven is not significantly different, due to the much lower fuel consumption of the sprinters, the fuel costs in the proposed situation are much lower compared to the current situation. Another observation is that the fuel costs costs and distance driven for the decentral bakery in Almere are low compared to the amount of stores it serves. A likely cause of this result is the fact that the stores served by this decentral bakery are located relatively close to each other, significantly decreasing the distance driven.

One sees in Table E.2 and in Figure E.5 that the total duration is slightly longer for almost every decentral bakery in the proposed situation compared to the current situation. Nevertheless, the sprinters are driven by production employees, whom have a lower average hourly rate than truck drivers. The lower hourly rate of the drivers in the proposed situation compensates for the slight increase in duration. As a result, the overall the drivers costs are lower in the proposed situation compared to the current situation. This difference is relatively less than the one that can be observed for the fuel costs.

Finally, the required fleet size is calculated based on the number of routes driven in the evening/night and in the morning. The number of morning and evening routes and the resulting fleet size per decentral bakery in both situations is reported Table E.3 and E.4. Nothing can yet be said about the vehicle costs per decentral bakery, since this is also dependent on how many vehicles are required during for the replenishment on Saturday. The vehicle costs are discussed in section 6.2.3.

The results concerning the required fleet size show that the required fleet size is equal in both situations. Interesting to note is that the decentral bakeries in Helmond and Dordrecht both have a significant imbalance between the number of morning and evening/night routes. This seems to be caused by the narrow morning time windows that many of their allocated stores have.

6.2.2 Replenishment routing for Saturday

The third and fourth case concern the routing on Saturdays in the current and proposed situation. The routing for these cases is displayed in Figure E.6 and E.7. Similar to the routing schemes of the first and second case, these figures depict the routing with straight lines between the locations. Since in practice the routing will follow the available routes, the displayed routing in these figures may be misleading to some extend. The fuel costs and distance driven of the corresponding routing schemes are reported in Table E.5. Additionally, the driver costs and duration are reported in Table E.6.

The results reported in Table E.5 and Figure E.8 show that, similar to the routing during the weekdays, the fuel costs in the proposed situation seem to be lower than in the current situation. However, this time the total distance driven per day seems to be significantly higher in the proposed situation. Nevertheless, the much lower fuel consumption of the sprinter causes the fuel costs to be lower.

The results concerning the driver costs and duration in Table E.6 and Figure E.9, show that the total duration is significantly longer in the proposed situation. Regardless of the lower hourly rates of the drivers of the sprinters, the total driver costs still come out to be more expensive in the proposed situation, with the decentral bakery in Helmond as exception.

The results concerning the required fleet size in Table E.7 and E.8 show that, again, Helmond and Dordrecht appear to have a significant imbalance between the number of morning routes and the number of evening/night routes in both situations.

6.2.3 Vehicle costs

The only remaining cost factor to be calculated are the vehicle costs in both situations. To recap, the fleet size of every decentral bakery is equal to the maximum between the required number vehicles on Saturday and the required number of vehicles during the rest of the week. Based on the results reported in the previous sections, the fleet size is determined and is reported in Table E.9 for the current situation and in Table E.10 for the proposed situation. The costs to utilize each vehicle for a single day is calculated using the yearly vehicle costs reported in the previous chapter. Using these costs and the fleet size, the daily vehicle costs is calculated for all decentral bakeries. The results of this calculation are reported in the last column of Tables E.9 and E.10 are visualized in Figure E.10.

In the previous chapter is described that the yearly costs for a sprinter are higher, due to its decreased technical lifespan. This together with the fact that in the proposed situation the fleet size is larger, causes the vehicle costs in the proposed situation to be higher than in the current situation.

In the current situation there is no difference between the required fleet size on Saturday and the rest of the week. On the contrary, in the proposed situation the required fleet size on Saturday compared to the weekdays is different for almost every decentral bakery. For every decentral bakery except Dordrecht, the total number of routes driven in the current situation on Saturday also does not differ from the weekdays. The only difference between Saturday and the rest of the week, in terms of input data, is the increase in average order size per store. As a result, this result shows that by increasing the demand in the current situation not much changes in terms of required vehicles or the total number of routes. In turn, this shows that the load constraint is not the major bottleneck in the current situation, but the time windows or the duration constraints are. The average capacity utilization of a truck during the week is therefore lower than the utilization of a truck during the weekend.

6.2.4 Comparison based on yearly costs

Using the results of all four cases, the yearly transportation costs are calculated for every decentral bakery. During a regular week, all decentral bakeries except Dordrecht and Almere replenish their stores 6 times per week. The decentral bakeries in Dordrecht and Almere also have to replenish their stores on Sunday, causing them to drive their replenishment routes all days of the week. As mentioned before, the Sunday routing scheme will be considered the same as the routing scheme for the weekdays. Therefore the the decentral bakeries in Sneek, Helmond, Zwolle and Doetinchem the weekday routing scheme is driven 5 times per week and the Saturday routing scheme once per week. The decentral bakeries in Almere and Dordrecht drive the weekday routing scheme 6 times per week and the Saturday routing scheme once. Multiplying the costs found for these routing schemes accordingly results in the yearly transportation costs, and the subsequent results are reported in Table E.11 and E.12. The relative differences between the two situations are reported in table E.13.

The comparison between both situations concerning the cost factors yields the same conclusion for all decentral bakeries. The vehicle costs are generally higher in the proposed situation, while the fuel and driver costs are lower in the proposed situation. Especially the fuel costs are much lower due to the 50% reduction in fuel consumption per km of the sprinters. In general, the proposed situation seems to perform better in terms of total transportation costs for all decentral bakeries. However, the degree of difference differs per decentral bakery. Table E.13 shows that for for example Zwolle, switching to the proposed situation only decreases the transportation costs by 2.81%. On the contrary, Dordrecht shows a saving of 22.3% in total transportation costs by switching to the proposed situation. The degree of savings therefore seems to vary from bakery to bakery.

In addition to this, Table 6.1 shows that the profitability of the proposed situation is also heavily influenced by an increase in demand. These results show that while the proposed situation is cheaper during a weekday, on Saturdays the relative savings achieved are much lower. Some bakeries even have a slight increase in transportation costs on Saturday in the proposed situation compared to the current situation.

Table 6.1: *Savings achieved in the proposed situation compared to the current situation*

Decentral bakery	Saving achieved on weekday	Saving achieved on Saturday
Sneek	20%	3%
Helmond	15%	9%
Doetinchem	15%	-4%
Zwolle	13%	-6%
Dordrecht	29%	12%
Almere	16%	-1%
Average	18%	2%

A visual comparison of the composition of the total transportation costs in both situations is shown in Figure E.11.

Nevertheless, overall the yearly transportation costs in the proposed situation are lower than in the current situation. The estimated monetary savings when changing to utilizing sprinters and stackable crates is equal to approximately 13.15% of the estimated total yearly transportation costs in the current situation.

Chapter 7

Conclusion and recommendations

The goal of this investigation is to calculate the difference in transportation costs between the current and proposed situation. In the current situation, the replenishment routes are driven with trucks and metal racks containing the pastries. The retailer wants to change this process by changing to a type of stackable crate instead of the metal rack. The subsequent reduction in weight and volume, in turn, allows the retailer to change to a sprinter instead of the larger truck. The results of this investigation are described in the previous chapter. In this chapter, the conclusions of the research are discussed together with some recommendations for implementation.

7.1 Conclusions

Following the results of the routing model, the first comparison made is the fuel and driver costs incurred during a weekday in the current and proposed situation. Although the total distance driven is more or less the same in both situations, the results show that in terms of fuel costs the proposed situation performs significantly better during the weekdays. The cost savings are in this case therefore achieved because the fuel consumption of the sprinter is half that of the truck. Compared to the current situation, the proposed situation saves 50.03% in terms of fuel costs during the weekdays. For the routing on Saturdays, the results are different. In the proposed situation, the total distance driven is on average 16.52% more than the estimated total distance driven in the current situation. Nevertheless, the fuel costs in the proposed situation are lower due to the much lower fuel consumption of the sprinters. On Saturdays, the proposed situation saves 41.74% in terms of fuel costs compared to the current situation.

The results concerning the driver costs during the weekdays (Figure E.5) show that the proposed situation is performing better. During the weekdays, the total duration is one average 2.53% more in the proposed situation. Dordrecht is the only bakery that has a lower total duration in the proposed situation compared to the current situation. The difference is however quite small, only 2.8%. Dordrecht seems to be the exception here due to the exceptionally large difference between the hourly rate of the truck drivers and the production employees. Nevertheless due to the much lower average salary of a production employee compared to that of a truck driver, the proposed situation still saves costs on average. In the proposed situation, the driver costs are on average 14% lower during the weekdays compared to the current situation. On Saturdays, the results are again quite different. In the proposed situation the duration is on average 21.69% higher compared to the current situation. Due to the decreased hourly rates of the sprinter drivers, this only results in a slight cost increase when compared to the current situation. The driver costs on Saturday are on average 2.36% higher in the proposed situation compared to the current situation.

The vehicle costs are higher in the proposed situation for every decentral bakery when compared to the current situation. This is not only because generally more vehicles are required in the proposed situation, but also because of the increased costs per sprinter. The difference between the weekdays and Saturday is only an increase in average order size per store. In Chapter 6, the results show that the required fleet size in the current situation is the same during the week as on Saturday. In the proposed situation, the increased order size almost always causes the required fleet size on Saturday to be higher. This shows that the required fleet size is generally less sensitive for increased demand in the current situation than in the proposed situation. This could be an indication that the load or duration constraints are the bottleneck in the current situation. Nevertheless, the vehicle costs still are 10.96% higher in the proposed situation compared to the the current situation.

In the previous chapter, Table 6.1 showed that the savings achieved by switching to the proposed situation varies between a weekday and on Saturday. The proposed situation is always cheaper during the week, but is only slightly cheaper on a Saturday Nevertheless, overall the proposed situation yields less transportation costs than the current situation. The total savings achieved by switching the sprinters and stackable crates are estimated to be 13.5% of the transportation costs in the current situation. As can be seen in Table 7.1, not every decentral bakery shows the same relative savings. The relative savings range from 2.81% (Zwolle) to 22.30% (Dordrecht). Since the driver costs are the largest portion of the transportation, the total achieved savings also are heavily influenced by the degree of driver cost savings achieved.

Table 7.1: *Relative cost differences when compared to the current situation*

Decentral bakery	Fuel cost decrease	Driver cost decrease	Vehicle cost increase	Total cost decrease
Sneek	48.80%	12.25%	37.21%	10.53%
Helmond	48.85%	10.38%	2.91%	13.23%
Doetinchem	48.18%	5.58%	28.63%	5.97%
Zwolle	46.42%	1.34%	37.21%	2.81%
Dordrecht	52.49%	27.96%	28.63%	22.30%
Almere	47.38%	8.25%	28.63%	8.65%
Average	48.69%	10.96%	27.20%	13.15%

To summarize:

- The fuel costs are always lower in the proposed situation compared to the current situation.

- The driver costs are lower in the proposed situation on a weekday, but slightly higher on Saturday.
- The required fleet size is more sensitive to an increase in average order size in the current situation, but switching to the proposed situation increases the overall vehicle costs.
- The difference between the total costs during a weekday and on Saturday shows that an increase in average order size heavily influences the relative savings achieved in the proposed situation.
- Switching to the proposed situation achieves savings for every decentral bakery. The degree of savings achieved varies, but on average an estimated cost saving of 13.15% can be achieved.

7.2 Recommendation for implementation

The results of the investigation shows that there is a saving potential when changing to using stackable crates and sprinters in the retailer's supply chain. However, saving transportation costs was not the only reason to propose changes to the replenishment process. Several other advantages exist such as less noise during unloading, easier access to cities and the fact that sprinter drivers are less scarce. Therefore, first the qualitative aspects of the change to crates instead of metal racks should be researched. The crates bring advantages in terms of reduced noise and weight, but can also bring disadvantages such as reduced visibility of their contents and reduced ease of handling when removing the pastries from their crates. Therefore the first step after this investigation should be an investigation into the effects this has on all parties that are affected by the change to crates and sprinters. Once the qualitative aspects are known, the retailer can make a more informed decision.

The next phase should be a pilot at one decentral bakery, if the retailer decides to implement the change. A change to the distribution network as significant as this is likely to come with unforeseen problems and costs. Factors such that have previously not been taken into account such as the stochastic nature of demand and reliability of delivery could have unforeseen costs. Since the stores are allocated to a single decentral bakery and the decentral bakeries operate independently from each other, it is to implement the changes to one decentral bakery first. If the results of the pilot are satisfactory, the retailer can move on to the final phase.

Currently, there are still several trucks that are driven under the terms of a lease contract and are therefore relatively new. Therefore, it is very likely that the sprinters will be incrementally introduced into the replenishment routes. The change to stacks of crates has to happen before the change to sprinters, since due to their weight limit, the sprinters cannot replenish the stores using the metal racks as containers for the pastries. However, since the trucks have a weight limit that is more than sufficient in any case, they are allowed to transport the stacks of crates. Therefore the final phase should start with replacing the metal racks with stacks of crates. Afterwards, depending on the lease contracts and value of the existing fleet, the sprinters should be introduced.

Chapter 8

Discussion, limitations and opportunities for further research

The results of this research should show that changing to sprinters and crates has a potential for a decrease in transportation costs. However, there are some points of discussion about the research conducted, as well as several topics that require further research. The goal of this chapter is to discuss any limitations of this research as well as to provide some interesting topics for the retailer to investigate further.

There are several limitations to the model used for this research. The first limitation is that the order size for every store was handled as if it is deterministic when determining the routing schemes. However, in reality the order size is stochastic due to the natural variation in demand. Using a stochastic order size as input for the model would allow the retailer to take the reliability of delivery into account. In case the retailer wants to determine the actual routing scheme, a simulation model could for example be used to include variation in demand and attune the routing scheme based on their desired customer service level. Furthermore, due to technical limitations the demand information used in the model is based on one representative week during 2018. Once the retailer determines their actual routing using their licensed routing software, the input data could be based on an average based on a larger data set, once this is available.

Secondly, because the routing is higher during on the Saturdays, the fleet size is generally based on the required vehicles on Saturday. No direct cost minimization has been applied concerning the routing and the required vehicles as a trade-off between the routing on Saturday and during the weekdays. Future research could be done on the resulting weekly transportation costs when changing the available fleet size at the bakeries. Finally, because a heuristic instead of an exact method is used in this research, there is no guarantee that the results presented are optimal. They should therefore only be considered an estimate of what the actual transportation costs will be.

There are several other interesting research topics related to the minimization of transportation costs at the retailer. As already mentioned in Chapter 7, the qualitative aspects of changing to sprinters and stackable crates have not been researched yet. Before changing to the new containers and vehicles an extensive analysis concerning these aspects has to be performed. Aspects such as the contractual agreements with the franchise owners or and expected future changes in urban environmental laws could have a significant effect on the feasibility of the proposed situation.

Another interesting topic is an investigation into the added costs of the time windows. As is clear to the management, the time windows of the stores have a large effect on the transportation costs. A brief and simple relaxation of the time windows at the decentral bakery in Helmond gives an impression of the added transportation costs caused by the time windows. The relaxation was performed with low computational effort and calculated the vehicle costs by simply dividing to total number of routes in half and rounding upwards. The results of the relaxation, together with the results for Helmond found in this research are reported in Table E.14. These results show that, relaxing the time windows yields a significant decrease in transportation costs. Of course this is not a realistic estimation since not all time windows are caused by the preferences of franchise holders. In practice some time windows are also caused by local law. In the future, the retailer could investigate which time windows are caused by local law, and which by the preference of a franchise owner. A careful analysis of the additional cost caused by the time windows set by franchise owners could be used to potentially further reduce the transportation costs. This information could for example be used to introduce incentives to any franchise owner that widens their time window. Another approach could be to use this information during the price negotiations once the current contract expires.

In addition to the time windows, the production scheme can also be adapted to potentially facilitate different delivery times. For example, by adjusting the production scheme so that delivery in the afternoon during opening times becomes possible. Stores that have narrow morning time windows could for example be prioritised, and delivered in the afternoon before closing time. This change would require the production schedule to allow for finishing a store order per batch, instead of finishing one product type per batch (which is currently the case). Naturally, further research is required to investigate the feasibility of such a scheduling approach.

Bibliography

[1] Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. “Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints”. In: *European Journal of Operational Research* 218.1 (2012), pp. 1–6. ISSN: 03772217. DOI: 10.1016/j.ejor.2011.07.037. URL: http://dx.doi.org/10.1016/j.ejor.2011.07.037.

[2] Richard S. Barr et al. “Designing and reporting on computational experiments with heuristic methods”. In: *Journal of Heuristics* 1.1 (1995), pp. 9–32. ISSN: 13811231. DOI: 10.1007/BF02430363.

[3] Olli Bräysy and Michel Gendreau. “Vehicle routing problem with time windows, Part II: Metaheuristics”. In: *Transportation Science* 39.1 (2005), pp. 119–139. ISSN: 00411655. DOI: 10.1287/trsc.1030.0057.

[4] J. F. Cordeau, G. Laporte, and A. Mercier. “A unified tabu search heuristic for vehicle routing problems with time windows”. In: *Journal of the Operational Research Society* 61.7 (2001), pp. 1072–1077. ISSN: 01605682. DOI: 10.1057/jors.2009.51.

[5] J. F. Cordeau et al. “A guide to vehicle routing heuristics”. In: *Journal of the Operational Research Society* 53.5 (2002), pp. 512–522. ISSN: 14769360. DOI: 10.1057/palgrave.jors.2601319.

[6] J Cordeau et al. “The VRP with Time Windows”. In: *The vehicle routing problem*. 2000.

[7] G. B. Dantzig and J. H. Ramser. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91. ISSN: 0025-1909. DOI: 10.1287/mnsc.6.1.80.

[8] M. Desrochers, J. K. Lenstra, and M. W P Savelsbergh. “A classification scheme for vehicle routing and scheduling problems”. In: *European Journal of Operational Research* 46.3 (1990), pp. 322–332. ISSN: 03772217. DOI: 10.1016/0377-2217(90)90007-X.

[9] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. “A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows”. In: *Operations Research* 40.2 (1992), pp. 342–354. ISSN: 0030-364X. DOI: 10.1287/opre.40.2.342. URL: http://pubsonline.informs.org/doi/abs/10.1287/opre.40.2.342.

[10] Michel Gendreau, Alain Hertz, and Gilbert Laporte. “New Insertion and Postoptimization Procedures for the Traveling Salesman Problem”. In: *Operation Research* 40.6 (1992), pp. 1086–1094.

[11] Michel Gendreau and CD Tarantilis. “Solving large-scale vehicle routing problems with time windows: The state-of-the-art”. In: *Cirrelt* 4 (2010), pp. 1–45. URL: https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2010-04.pdf.

[12] Michel Gendreau et al. “A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows”. In: *Operations Research* 46.3 (1998), pp. 330–335. ISSN: 0030-364X. DOI: 10.1287/opre.46.3.330.

[13] Brian Kallehauge. “Formulations and exact algorithms for the vehicle routing problem with time windows”. In: *Computers and Operations Research* 35.7 (2008), pp. 2307–2330. ISSN: 03050548. DOI: 10.1016/j.cor.2006.11.006.

[14] Francesco Maffioli. “The vehicle routing problem: A book review”. In: *4or* 1.2 (2003), pp. 149–153. ISSN: 16142411. DOI: 10.1007/s10288-003-0013-7.

[15] Martin W. P. Savelsbergh. “The Vehicle Routing Problem with Time Windows: Minimizing Route Duration”. In: *ORSA Journal on Computing* 4.2 (1992), pp. 146–154. ISSN: 0899-1499. DOI: 10.1287/ijoc.4.2.146.

[16] Michael Schneider, Fabian Schwahn, and Daniele Vigo. “Designing granular solution methods for routing problems with time windows”. In: *European Journal of Operational Research* 263.2 (2017), pp. 493–509. ISSN: 03772217. DOI: 10.1016/j.ejor.2017.04.059.

[17] Marius M. Solomon. “Algorithms for the Vehicle Routing and Scheduling Problems With Time Window Constraints.” In: *Operations Research* 35.2 (1987), pp. 254–265. ISSN: 0030364X. DOI: 10.1287/opre.35.2.254.

[18] Paolo Toth and Daniele Vigo. “Models, relaxations and exact approaches for the capacitated vehicle routing problem”. In: *Discrete Applied Mathematics* 123 (2002), pp. 487–512.

[19] Paolo Toth and Daniele Vigo. “The Granular Tabu Search and Its Application to the Vehicle-Routing Problem”. In: *INFORMS Journal on Computing* 15.4 (2004), pp. 333–346. ISSN: 1091-9856. DOI: 10.1287/ijoc.15.4.333.24890.