Integrating and optimizing the milk-run of Thales Hengelo by creating a Vehicle Routing Problem model

University of Twente
Master Thesis IEM

Thomas Krullaars

April 2024

Master Thesis Industrial Engineering and Management

Production and logistics management track

Faculty of Behavioural, Management and Social sciences

University of Twente

Integrating and optimizing the milk-run of Thales Hengelo by creating a Vehicle Routing Problem model

Author
T. Krullaars (Thomas)
MSc Industrial Engineering and Management
April, 2024

Thales Hengelo
Zuidelijke Havenweg 40
7554 RR Hengelo
The Netherlands

University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

Supervisor Thales Hengelo
L. Dubruel Sabathier (Louis)

Supervisors University of Twente
Dr. A. Trivella (Alessio)
Dr. ir. E.A. Lalla (Eduardo)

Preface

The completion of this thesis means the completion of my Master Industrial Engineering and Management. This also means the completion of my student time and my student life. I can look back at a fantastic time of almost seven years as a student in Enschede, in which I set up a completely new life and gained some of the best experiences in my life. I am proud and grateful for the opportunities I have received and taken to develop myself as a professional and also as a person.

First of all, I would like to thank Thales for trusting me with this project and allowing me to perform my master thesis under their supervision. I would like to thank my supervisor Louis Dubruel Sabathier for supporting me when I needed it and the warm welcome I received within Thales. This welcome could also not have been possible without the other colleagues of the Improvement Team and the other people I have collaborated with, so thank you guys as well. At Thales, I learned a lot about the real working life, since this was my first real experience in a company next to my Bachelor Thesis in Covid times. It has helped me to understand what I like and what I want in a future job and career, which I am really grateful for.

Furthermore, I would like to thank my first supervisor from the university, Alessio Trivella, for guiding me through my master thesis. The first time we spoke, we started with a fairly vague problem and had to adjust and redefine in order for it to be a real master thesis. I am glad to say that we accomplished that and I could not have done that without your help and ideas. I also want to thank my second supervisor from the university, Eduardo Lalla, for providing intermediate feedback.

Finally, I want to thank all the people that have supported me, not only throughout my master thesis, but throughout my whole student life. In special, I want to thank Rozan Hopman for being my buddy during the writing of my thesis and providing valuable support and feedback from start to end. I want to thank my family, roommates, fellow board members, football and tennis friends and fraternity for the constant interest and support and for all the great memories we have made in Enschede and beyond. You made my student life the way it is and I have enjoyed it enormously.

I hope you enjoy reading this thesis and slapen doen we ‘s nachts.

Thomas Krullaars
Enschede, May 2024

Management Summary

This research was conducted at Thales Hengelo, which has a focus on the naval defence branch of Thales. Within Thales Hengelo, there are multiple workcells and departments that work on different type of products. The pick-up and delivery of products for these workcells within Thales is done with a basic milk-run. The milk-run uses multiple vehicles to visit every station. The biggest part of the milk-run is visited by a vehicle called the tugger train. Another part of the route is covered by the bakkerskar, and loose transport is done by a forklift truck. Currently, the milk-run drives twice a day where it delivers and picks up several kinds of demand: jobs within one hour, 24 hours or three working days. There are also some other implications that the milk-run is (partly) responsible for which are all in the range of pick-up and delivery. The milk-run is responsible for 20-40% of the daily hours that are put in the transport of products within Thales which is around 30-40% coverage of the total parts movements.

At the moment, the implementation of the milk-run is not sufficient to tackle all requests for internal transport, which leads to production lines staying still while waiting on new products. Thales would therefore, like to see a milk-run in place where one vehicle can serve all the requesting workcells all their desired products, as now there is no integration of the milk-run throughout Thales. Furthermore, another building will open soon, which also has to be supplied. These elements all lead to a strong need for an improved milk-run that is connected to all parts of Thales. Combining this current situation with the research goal, the research aims at answering the research question:

How should the plan, which integrates and optimises the milk-run throughout Thales, be built up?

The literature review revealed that the milk-run can be regarded and modelled as a Vehicle Routing Problem (VRP) with some extensions to replicate the situation at Thales. The VRP that is created combines multiple extensions which are introduced in the literature review. The first extension is using time windows for demand to be fulfilled in. A variant on this is introduced in the form of soft time windows. This allows products to be delivered outside of their time window, but will be penalised if this is the case. The next extension is the split delivery possibility to be able to model the 24-hour job demand over a whole day. Lastly, the compatibility constraint extension is introduced to allow and disallow certain vehicles to drive to certain nodes on certain times. All in all, this leads to the Split Delivery Vehicle Routing Problem with (partial) Soft Time Windows and Compatibility Constraints (SDVRPSTWCC). This is modelled and solved in Python with the help of the solver Gurobi. Before solving, the input data needs preparation.

Data regarding the milk-run, locations of drop-off stations, product demand and its division during the day is gathered. Based on this, three pre-processing steps were taken. The first being the clustering of drop-off locations for the milk-run. Because of a lack of freedom in the route that is possible at Thales, stations are clustered, as travelling one station means also (practically) passing the other stations within the cluster. The second and third pre-processing techniques make sure that the demand is spread over the day and allocated to certain time windows respectively to represent a typical day concerning demand for the milk-run in Thales. This acts as input for the VRP model. The output of the VRP model is a combination of the objective function and the routing. The objective function adds up the travel time of each route, the stopping time that belongs to each customer and the penalties that are awarded for overtime of products delivered outside their time window. This objective is expressed in seconds.

We analyse four main scenarios for the base case analyses, these are shown in Table 1. The current situation entails the current way of working and has an objective function of 6,254. Optimizing the current situation in the model, provides an improved routing which corresponds to an objective function of 5,614,

which is a decrease of 10.2%. Comparing this to the benchmark routing of 6,523, there is an improvement of 14%. This benchmark routing is created by using the nearest neighbour algorithm to form standard routing. Looking at future scenarios, we differ between the possibility of a new vehicle (which can combine indoor and outdoor driving), and in both cases the new building along with the new spread of demand is in place. The rise of 1-hour jobs and the new building cause a bigger need for a more regular milk-run. Providing this regularity as an optimal solution and using one vehicle, decreases the solution for the future situation to 5,803. The routings for both future situations perform better than the benchmark routing. Using two vehicles for the same route causes a great expansion in time as the switch in vehicles requires a lot more driving distance and thus more time.

Situation	Frequency * vehicles	Objective	Benchmark routing
Current situation	2*2	6,254	6,523
Optimal current situation	4*2	5,614	
Optimal future situation 1 vehicle	4*1	5,803	6,138
Optimal future situation 2 vehicles	4*2	7,115	7,450

Table 1 - Findings base case and benchmark analysis

The routing of the overall optimal solution that is created, is always a combination of driving the complete route for visiting all customers and one or two routes that are shorter, because of the lack of demand for certain nodes. For the current situation two sets of routes are created: Routes 1 and 4 cover a part of all clusters (LC – W1.1 – W0 – W1.2 – LC) and are done with one vehicle. Routes 2 & 3 are the optimal complete route (LC – W1.1 – W1.2 – W0 – Z – GCC – LC) and swap vehicles after W0. The optimal complete and partial routes in the future are fairly similar, but include the new building STC after leaving W0. The routing that is computed is shown in Figure 1. Optimal complete route means the route that visits all clusters in the optimal order.

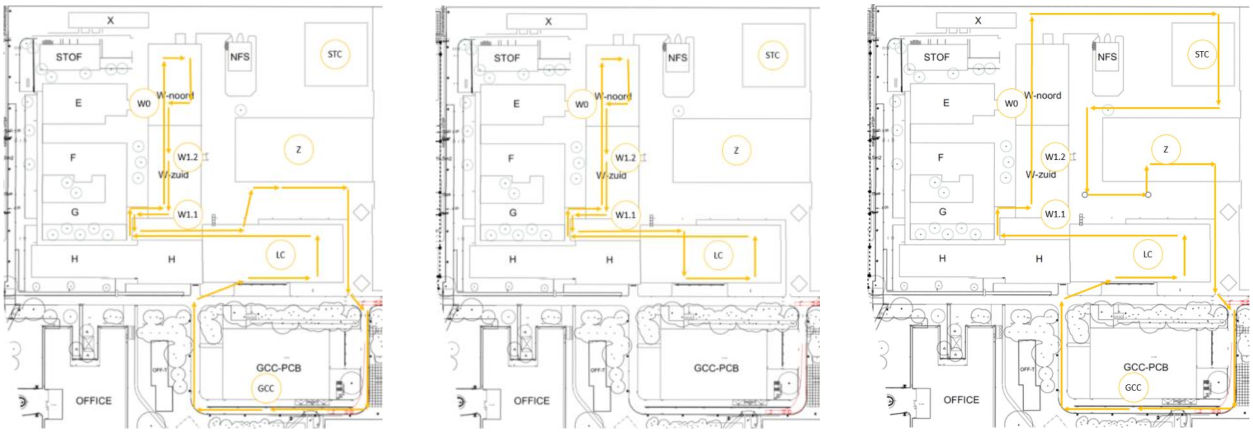


Figure 1 – Left: Current complete route; Middle: Current partial route; Right: Future complete optimal route

To assess the solution, a sensitivity analysis was conducted across various parameters. Adjusting the 24-hour job demand showed no impact on the optimal solution. Introducing additional 1-hour job demand marginally affected the objective function, with increases of only 10% and 16% for increasing demand factors of 2 and 3 respectively. Combining both 24-hour and 1-hour demands yielded consistent solutions for a factor of 2, but led to a 27% objective function increase for a factor of 3 due to an extra vehicle requirement. This underscores the non-sensitiveness of the solution. Regarding vehicle capacity, it was evident that as long as capacity remains above half of the total daily demand, 19 products a day, the objective function remains stable. This indicates that the current capacity of 50 boxes is sufficient, with only slight adjustments needed for future demand increases. Finally, the penalty for delivering products in overtime was altered. When the ratio of one internal transport movement being equal to a certain number of minutes of overtime for a product drops below 0.11, the model occasionally decides to deliver products in overtime, depending on distance and quantity. This means there is a critical point at equalling one internal transport movement to 40 minutes of overtime, which is half of the current ratio.

We provide clear steps to integrate and optimise the milk-run throughout Thales. Experiments with the model confirm that driving the milk-run four times daily boosts performance, both currently and in the future. Allowing Thales to make this shift, saves time compared to driving twice daily. While more resources are needed upfront for the increased frequency, overall resource usage drops because 1-hour jobs are incorporated into the milk-run. While the current route is optimal for two vehicles, it could improve with a vehicle capable of indoor and outdoor clusters in one route. Continuing to search for such a vehicle is crucial. If not feasible, upgrading the outdoor ground surface could help. Consolidating routes into larger ones with inside and outside combined will greatly reduce milk-run time, especially with the new building farther from the logistics centre. The time saved translates to increased productivity and saving at least 2 hours and 40 minutes. Investing in a new vehicle, estimated to cost as much as the current one, would allow disposal of the old vehicles, resulting in further savings. As the main goal for Thales is to be able to handle the upcoming demand, this saving is a welcome extra benefit. Based on these conclusions, we formed the main recommendations for Thales in the current situation:

- Increase the number of milk-run shifts per day to four, to increase productivity and be able to handle future demand.
- Implement the optimal routing scheme based on the VRP model, to save time and not unnecessarily visit empty stations.
- Take a new vehicle into use to facilitate the optimal routing created by the VRP model. This way indoor and outdoor routes can be combined in one route.
- Replace 1-hour job requests with 'next milk-run jobs' to decrease the separate product movements as much as possible. This enhances health and safety by reducing vehicle traffic on the shop floor and standardizing employee tasks.

Next to the current recommendations, there are some additional recommendations to follow-up in the future to add onto the milk-run and its process:

- Include buildings/stations that are currently not in the regular loop of the milk-run by implementing a button system. When a button is placed at the station, the internal transport employees know to include this building in their route if the button is turned on.
- Work on visual management to raise awareness of the milk-run. This will create more trust in the milk-run and therefore decrease the requests of 1-hour jobs.
- Digitalise the milk-run information and create a dynamic version to calculate the routing. This includes a device present on the vehicle of the milk-run which mentions what stations to visit and what the best routing is for those stations combined. This will guarantee an optimal route for each tour and enhance traceability of products.

Contents

List of abbreviations

CSHW	Combat Systems Hardware
CW	Central Warehouse
KPI	Key Performance Indicator
LC	Logistics Centre
OTD	On-Time Delivery
SDVRP	Split Delivery Vehicle Routing Problem
SDVRPSTWCC	Split Delivery Vehicle Routing Problem with (partial) Soft Time Windows and Compatibility Constraints
SLA	Service Level Agreement
VRP	Vehicle Routing Problem
VRPCC	Vehicle Routing Problem with Compatibility Constraints
VRPSTW	Vehicle Routing Problem with Soft Time Windows
VRPTW	Vehicle Routing Problem with Time Windows

List of tables

List of figures

1. Introduction

This research aims at improving and optimising the process behind the milk-run of Thales Hengelo. A milk-run is a manually operated, cyclic transport system delivering raw materials and finished goods, using a fixed route and time schedule (Droste & Deuse, 2012). In this chapter an introduction of the encountered problem is provided. This is done by elaborating on multiple elements that are relevant for the main topic of the research. In Section 1.1, the company Thales is introduced. Then, in Section 1.2 the motivation for the research is explained, where the initial assignment is provided along with the problem identification. Finally, the research approach is worked out in Section 1.3, where the research goal is posed together with the research questions and the scope of the research.

1.1 About Thales

Thales Group is a global leader in technology with more than 80,000 employees all over the world. Thales is investing in digital and “deep tech” innovations to ‘build a future we can all trust’. Thales's high-tech solutions, services and products help companies, organisations and governments to achieve their goals and ambitions within five different markets — security, defence, aerospace, space, and transportation.

This research takes places at Thales Hengelo, which is mainly focused on the naval defence branch of Thales. The team in which this research has been conducted, is the Supply Chain Improvement Team. This is divided into Lean and Industry 4.0. This research is conducted through the Lean segment. Close collaboration with the logistics department of the ISCM (Industrial Supply Chain Management) segment is also in place. Thales is built up out of many layers. The layers that are mentioned all are part of the Naval Industry & Supply Chain organisation. There are thirteen workcells that work on different type of products in Hengelo. These are spread out over five departments. Thales desires a situation in which each workcell can make use of the same milk-run with the same vehicle, which is currently only installed in one building. This covers around 30-40% of the products/workcells. Thales would like to see a milk-run in place where one vehicle can serve all the requesting workcells all their desired products.

Thales is at a critical point considering their supply chain. This is because of the fact that most of their new projects are very late compared to their commitments. The development of several projects takes longer than expected. That is why every improvement in the current supply chain of Thales is important for them to win back time. When tackling this problem, the supply chain for the workcells that currently do not have a milk-run, or a milk-run that is isolated from the main one will improve. Also, the already installed workcells that make use of the milk-run will profit from this new situation.

1.2 Research motivation

Section 1.2 explains the problem identification and the research problem that comes along with this. For the problem identification, a problem cluster is created to visualise where problems connect and which problem causes what. Also, the action problem and the core problem are identified to lay a basis for the rest of the research. And the research problem is stated and linked to the KPIs that are present throughout the research.

1.2.1 Problem identification

Thales currently finds itself in a situation where the process around its milk-run is inefficient, because of a lack of integration with the rest of the company. This ineffective milk-run causes multiple problems. Some of these problems in their turn lead to other problems. To visualise this situation and identify the core problem, the action problem and every problem in between them, we set up a problem cluster. The cluster shows the core problem on the left and works towards the action problem on the right. Each arrow in the cluster means that the problem on the left of the arrow causes the problem on the right of it. Therefore, it is clear there is only one action problem, which is the following:

Action problem: Production lines stay still while waiting on new products.

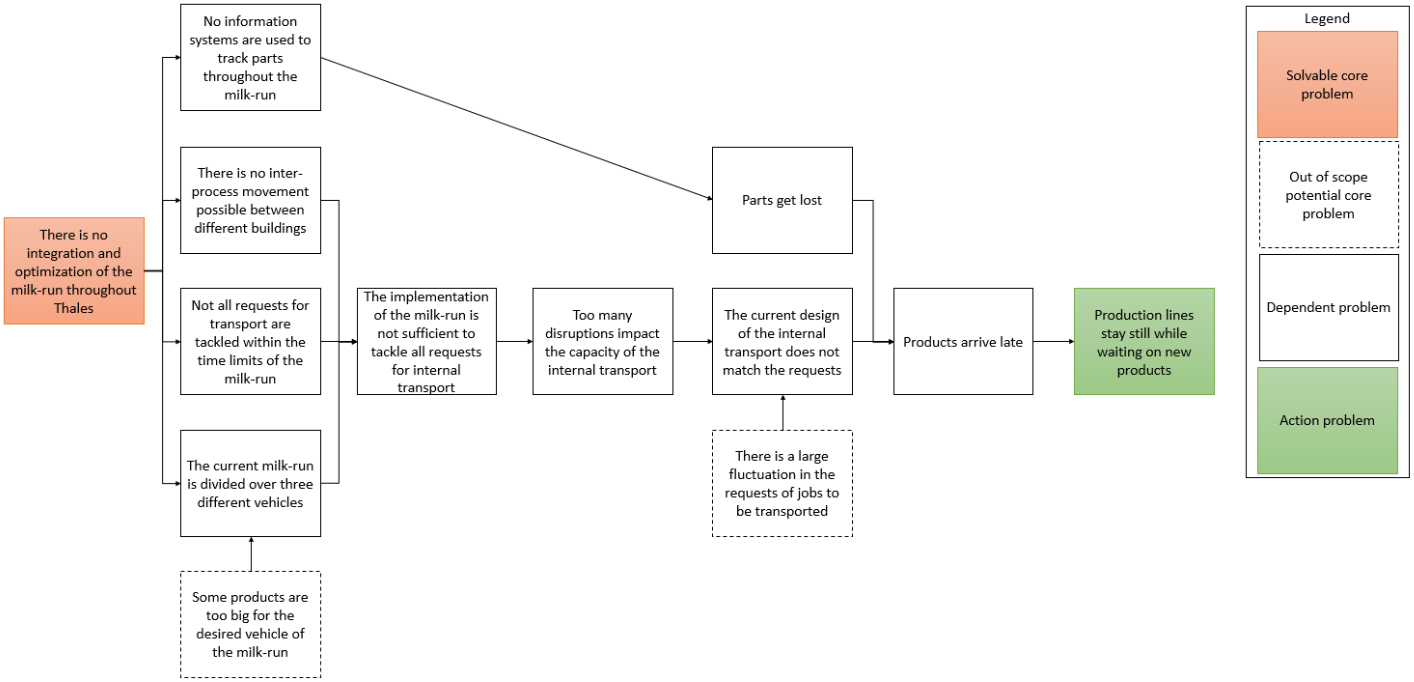


Figure 2 – Problem cluster milk-run process

This action problem is caused by the fact that sometimes products arrive late when they are requested at a department. There are two reasons for the late arrival of products: parts get lost, and the current design of the internal transport does not match the requests. Parts get lost, because there are no information systems that track parts throughout the process of the milk-run, which means there is no overview of where products are within the facilities in between part usages. The design problem is caused by the occurrence of too many disruptions (irregular job requests that need priority over the regular jobs) that impact the capacity of the internal transport and the fact that there is a large fluctuation in the requests of jobs to be transported. This fluctuation in demand can be a potential (partial) core problem but is left out of scope for this research as this will not solve the whole problem.

The many disruptions are due to the reason that the implementation of the milk-run is not sufficient to tackle all requests for internal transport. This has three causes; There is no inter-process movement possible between different buildings, which causes products to have to be dropped at the logistics centre before it reaches another building. Not all requests for transport are tackled within the time limits of the milk-run. And the current milk-run is divided over three different vehicles. This also means that products have to be transferred from vehicle to vehicle before being able to reach a different building. At the moment, there is a desired vehicle for a milk-run that delivers only to one production facility building. This system is designed in such a manner that other buildings cannot be visited by the same vehicle and thus do not connect with this milk-run. This is due to the fact that the vehicle that drives the milk-run is unable to go outside, because the vehicle is not suited for outdoor driving, which means some workcells cannot be reached with this vehicle.

The division over vehicles is partly caused by the fact that some products are too big for the desired vehicle of the milk-run. Therefore, those products are excluded from the milk-run as they are too heavy or shaped too inconvenient to be carried by the tugger train. For these products, a forklift truck is used by the regular internal transport. This truck is directly sent from one location to the other. This potential core problem is unsolvable as the weight and shape of these products will not change and therefore always require a forklift truck to carry them from A to B. This problem is therefore out of scope. Therefore, one problem remains, which is the core problem of this research.

Core problem: There is no integration and optimization of the milk-run throughout Thales.

In the current situation, there is no plan for the integration and optimization of the milk-run throughout Thales. A plan to tackle this situation will eventually tackle the action problem as all problems are related. This is therefore the solvable core problem and to solve this problem is the aim of this research.

1.2.2 Research problem

The problem identification of Section 1.2.1, provided the core problem. Combining the action problem with the core problem, provides the straight-forward research problem. There are some KPIs directly related to this integration and optimization and the current performance of the milk-run. These concern topics like the on-time delivery & lead time of the jobs, the productivity and the costs that come along with the milk-run. These KPIs are monitored in order to create a clear picture of the status of the integration of the milk-run and are closely linked to the action problem, since this problem mentions production lines staying still. This is due to effects that are measured by the KPIs. For example, lead time and on-time delivery play a major role in analyzing the action problem and to see if a decrease is observed in the relevant KPI. The productivity is a result of the research problem, which should therefore increase when this problem is being tackled. Lastly, the costs are a relevant aspect to keep in mind but are mostly incorporated within the productivity KPI. Data gathering concerning the KPIs is a struggle within Thales, so sometimes it might be difficult to gain insight when desired.

1.3 Research design

This section describes the research goal and the corresponding research questions. The research questions are divided throughout the rest of the chapters and are divided over several sub-questions that will be addressed one-by-one.

1.3.1 Research goal

According to the rules of Heerkens (2017), the research goal comes forward from the core problem and is to answer the main research question:

Research question: How should the plan, which integrates and optimises the milk-run throughout Thales, be built up?

Next to this research goal, it is important to keep the limitations and challenges in mind that come along with the implementation of this plan. To elaborate further on all of these aspects, multiple directions on the subject must be investigated with the help of other research questions.

1.3.2 Research questions

The main research question results in several sub-questions, which need to be answered. These are structured throughout the chapters of this research: current situation, literature review, solution design and the analysis of the results. After these chapters, the conclusions and recommendations will be provided. Each chapter has its own research question which are followed up by more in-depth questions to explore different aspects of the research question.

Chapter 2 – Current situation

Question 1. What is the current situation of the milk-run of Thales?

- 1.1 How is the process around the current milk-run organised?
- 1.2 What is the current route of the milk-run?
- 1.3 How are the workcells organised and what is their demand?
- 1.4 What does the process of keeping track of the products look like?
- 1.5 How is the other transport within Thales organised?
- 1.6 What are the KPIs that are relevant for the milk-run?

The first research question aims to analyse the current state of the milk-run and the relevant factors that influence the milk-run. Sub-questions have been formulated to evaluate all of these factors where necessary. Chapter 2 therefore, discusses the main process of the milk-run, its route, the visited workcells, the information flow around it, the remaining transport in Thales and the KPIs of the milk-run.

Chapter 3 – Literature review

Question 2. What methods to create and optimise a model for the milk-run are available in the literature?

- 2.1 What is a milk-run?What solutions exist in the literature to model similar problems as the milk-run?How can a milk-run be optimised as a VRP?What extensions for the VRP are needed to represent the milk-run of Thales?

To link the current state of the problem to the appropriate academic field, a literature review will be carried out. This literature review aims to provide a thorough basis of knowledge to create and optimise a model for the milk-run. First of all, the principle of a milk-run will be explained. This is followed by the possibilities to model a milk-run. Also, questions concerning the setting up of the model of the milk-run and the data needed will be handled. Lastly, optimisation techniques and extensions on the milk-run are discussed.

Chapter 4 – Solution design

Question 3. How can we design a model which integrates and optimises the milk-run throughout Thales?

- 3.1 How should the data be prepared before the implementation of the VRP model of the milk-run?
- 3.2 What sets, parameters and variables are considered for the VRP model of the milk-run?
- 3.3 What objective function and constraints are considered for the VRP model of the milk-run?
- 3.4 How can the optimal solution be determined?

To put the literature into practice and apply it to the core problem, we must look at the solution design. Chapter 4 aims at defining the model for the milk-run, where the preparation, sets, parameters, variables, constraints, input data and solution determination are discussed.

Chapter 5 – Experiment design

Question 4. What outcomes does the model provide and how is its overall performance?

4.1 What model do we use to model the milk-run of Thales?
4.2 How is the data prepared to provide a stable basis for the experiments?
4.3 How does the solution improve with the help of the model currently and in the future situation?
4.4 How sensitive are the solutions?
4.5 How does the model perform in terms of scalability?

Chapter 5 will dive into the experiments that will be carried out with the help of the model. It starts off by applying the pre-processing techniques to prepare the data and create a firm base for the experiments. After this base is set, the performance of the model is looked into and the main solutions are provided. These will be compared to the current situation and a benchmark instance. Also, the sensitivity and scalability of the model and the solutions will be tested.

1.3.3 Scope

The crucial aspect of the research and thus the major focus is to create new and improved design for the milk-run, where optimising the routing and its frequency of the milk-run are the most important parts.

Furthermore, the problem cluster shows the problem ‘No information systems are used to track parts throughout the milk-run’. This implies that there are options to improve the traceability of the products on the milk-run as well, but this part will be considered as a secondary problem. This means the cause and the consequences of this problem will be analysed, but an in-depth solution for this problem on its own will not be provided as it is of less influence to answering the research question. There will however be some recommendations concerning this topic, but it remains a secondary goal of the research.

Lastly, there are multiple aspects that are also relevant for the milk-run and can improve the process around it, like a steadier demand flow. However, this research cannot discuss all of these topics, so they are out of scope.

2. Current situation

Chapter 2 describes the current situation at Thales. An answer on the first research question *'What is the current situation of the milk-run of Thales?'* is provided throughout the sub-chapters. Section 2.1 describes the current organisation of the milk-run. In Section 2.2, the route of the milk-run is examined. Afterwards, the workcells are described in Section 2.3. Then, Section 2.4 describes the information flow around the milk-run. Section 2.5 provides information concerning the other transport that is carried out. And Section 2.6 describes the objective of the milk-run. Lastly, Section 2.7 provides a conclusion of this chapter.

2.1 Milk-run process organisation

The process of the milk-run has many elements and characteristics that are linked to it. It is important to adequately identify each aspect of the milk-run that influences the milk-run or is influenced by it. An elaboration on the following elements is provided to gain insights in the process and its organisation: the features of the milk-run, the vehicles that are used for the milk-run, the level-pull follow system that is in place and the Kanban system that is involved in combination with the levelling box.

2.1.1 Features

First of all, it is important to establish the meaning of a milk-run and the way Thales utilises it. In a basic milk-run system, material handlers drive standard routes through a facility at determined time intervals. During this interval the material handlers pick up containers at a central storage area, follow predetermined routes, deliver the materials to the requested areas and return to the supermarket (inventory buffers between different stages of production). A milk-run can also work the other way around, where a vehicle picks up items along the route and brings them to the storage area (Klenk, Galka, & Günthner, 2015). For Thales, the milk-run does both. It starts off loaded to deliver products along the route, but it also picks up products that have finished a certain step in a department to deliver back to the central warehouse (or other departments).

Thales uses the milk-run for delivering products at locations in their production facilities and picking up products from locations and bringing them back to the demanded location, which is the logistics centre most of the time. The general Service Level Agreement (SLA) is to deliver each requested job (a list of different products for one 'end-product') within three working days. When the milk-run does not perform well, the on-time delivery and the lead time might get impacted, triggering the action problem of the research; production lines stay still while waiting on products.

The milk-run drives twice a day; at 10 AM and 2 PM. Next to these regular milk-run operations, where products are picked from the warehouse and sent on the milk-run and reverse, there are also some other implications that the milk-run is (partly) responsible for (when size of the products and the time-constraints allow):

- 24-hour pick jobs – Jobs that must be delivered within 24 hours. These jobs are loaded onto the milk-run after the job is picked if time allows.
- 1-hour pick jobs – Jobs that must be delivered within 1 hour. These jobs are occasionally loaded onto the milk-run after the job is picked if time allows.
- Direct delivery receipts – Products that skip the inventory phase and are directly delivered to departments after receiving them in the warehouse. These are placed on the milk-run when possible.
- Kanban-cards – These cards are dropped off and picked up by the milk-run operator. An explanation on these cards can be found in Section 2.1.3.

- Other types of distribution; spreading maps for picking instructions, shipments between departments & pickup of empty packaging.

Currently, the milk-run delivers only at stations and substations of them. This means that the milk-run operator drops something off at a rack and someone from the department has to walk to the rack and take it up to their own workplace. For some workstations it is beneficial to deliver directly to the workplace instead of the rack. For others this is not possible/desirable because of a shortage of storage room or inaccessibility of an area.

The milk-run is responsible for 20-40% of the daily hours that are put in the transport of products within Thales. This differs severely, because of the fluctuations in the number of requests for 1-hour jobs and other transport that cannot be brought by the milk-run. The milk-run is responsible for a 30-40% coverage of the total parts movements. Here the definition of a part movement is a movement where an internal transporter performs an action. So, moving a box of 500 screws and moving a full radar are both one movement. Currently, internal transport takes around ten hours a day in total. Of these ten hours, the milk-run is used for around 2 to 2.5 hours and is responsible for 35% coverage of the total part movements. Next to the milk-run, there is also some regular internal transport, which moves products from A to B without a certain route. These are discussed in Section 2.5.

2.1.2 Vehicles

The milk-run that is installed at Thales covers most of the production facilities. It uses three vehicles to visit every station (the point of usage). The biggest section of the milk-run is visited by a vehicle called the tugger train. Another part of the route is covered by the bakkerskar and loose transport is done by a forklift truck. Together, these three vehicles complete the whole milk-run to supply regular demand from the production facilities. This regular demand covers products that are brought from the logistics centre to the workcell, products from the workcell to the logistics centre and products from one workcell to the other.

2.1.2.1 Tugger train

Tugger trains are used to transport materials efficiently within a warehouse. A tugger train consists of carts that get towed by an operator, who drives the 'locomotive'. This tugger train brings relatively 'small' products around. For Thales, the tugger train consists of the LTX 70 vehicle, which is equipped with tires for inside the production halls. Thus, this vehicle is dedicated to indoor movements of products. This type of tugger train can be adjusted, so that it can go outside, but then it would need other wheels and frames (wagons) behind it compared to the current frames. The frames that are in use behind the LTX 70, the B-frames, are fully designed for indoor use as well. This disallows the tugger train to go outside as the platform of the wagons is too low and would bump into the road. The frames are suitable to transport pallets with load on it, but this is not done. Every pallet that needs transport is moved by a forklift truck. There are other frames available that are suitable for outside use with or without a hood over it to cover for weather circumstances, these are the C-frames. All the vehicle elements are shown in Figure 3.

Currently, the tugger train is leased from the company Still. Leasing the LTX 70 in combination with the four B-frames costs Thales €1,245.- (ex. BTW) per month. This lease contract can be terminated and other options in this company, like the C-frames can be chosen or it can be decided to choose a completely different company, which can provide similar vehicles. The tugger train is not able to drive outside as the ground surface of the road is too bumpy and the products should not get wet, so when it rains, there is no possibility to go outside with this vehicle anyway.

Figure 3 - Tugger train vehicles – Full, LTX70, B-Frame & C-Frame (Still, 2023)

2.1.2.2 Bakkerskar

As the current tugger train is unable to go outside, it means it cannot swap buildings. Therefore, another vehicle, called the bakkerskar takes a different part of the route to visit other buildings. This vehicle has a lower capacity than the tugger train, but is able to drive on the road outside. Therefore, currently after the tugger train has visited the indoor locations, it is parked at the logistics centre and the operator switches to the bakkerskar. This bakkerskar then immediately leaves the logistics centre and visits the necessary outdoor locations. It brings the same type of products as the tugger train. Sometimes a product that is picked up in the inside part of the milk-run is immediately placed in the bakkerskar to be brought to another location. This vehicle is purchased and does not need extra costs except for some charging.

2.1.2.3 Forklift truck

For large or unhandy shaped products, a forklift truck is used. The same goes for the products that are placed on pallets. There are multiple forklift trucks in different sizes. They are equipped with two sets of lift spoons, so that two pallets can be transported around at the same time if the size (height) of the pallet allows. The forklift trucks are used for inside and outside use and are taken up in the current milk-run route for two stations. These stations are not always visited by the milk-run, this depends on supply and demand.

2.1.3 Level pull flow, Kanban (cards) & Levelling box

There are some lean techniques in place at Thales that support the milk-run or are supported by the milk-run. These are level pull flow, Kanban cards and a levelling box. A pull system (level pull flow) is a lean technique for reducing waste in production processes. Applying a pull system allows you to start new work only when there is a demand for it. This allows a minimization of overhead and optimization of storage costs. The purpose of having a pull system at Thales is to build products based on actual demand and not on forecasts. By doing so, Thales can focus on eliminating other waste activities in their production processes. As a result of this, optimization of resources and a reduction in the possibility of overstocking are established.

Furthermore, applying a pull system will allow you to deliver work just-in-time. Just-in-time is a production model where deliverables are produced to meet actual demands and avoid overstocking and push strategies. This is a crucial aspect of the production process for Thales as a lot of workcells and thus production facilities are struggling with the space they are given. Currently, another building is being built to compensate for the extra demand that has arisen lately. The space for storage of products on the shop floor is thus very limited. By utilising a pull system, this space will be as empty as possible, so there is more room for other products to be produced (Toneva). The current pull idea within Thales, therefore, is to deliver everything at the LC (Logistics Centre), because the storage there is enough. Then during the next milk-run it can be brought to the corresponding department if necessary. For a future scenario it can be helpful to drop off products that are collected during a milk-run to its destination (other than the LC) within the same milk-run. The milk-run now picks up the products from the shop floor when it has the chance. By this, the pull system is created. This pull system is partly being controlled with the help of Kanban cards.

The milk-run uses this Kanban card system in combination with a levelling box to control the pull of the products within the production lines of Thales. 'Kanban – literally, a visible record or plate used as a means of communication, of conveying ideas and information. … the card or marker system used to control work-in-process inventory, production, and parts suppliers' (Esparrago Jr, 1988). This system manages the flow of products and authorizes production and movement of materials between work centres. In the production facilities of Thales, so-called 'launchers and small bins are installed on stations to provide overview of the Kanban cards and regulate the flow. The milk-run will visit these stations and carry out the corresponding task of the Kanban card.

The overview of these cards is located at the levelling box. The levelling box is for storage of the Kanban cards and an overview regarding the status of the Kanban process accompanied with the relevant KPIs, like On-Time Delivery. This levelling box is also called a Heijunka board. Such a Heijunka Board is a Lean Production technique that helps to communicate a timed production plan to the factory floor (Hersyah & Derisma, 2018). Before each milk-run, the overview of the levels of volume and mix of the products is reviewed and taken into consideration to adjust each milk-run accordingly.

2.2 Milk-run route

Currently the milk-run route is divided over the three vehicles that are described in Section 2.1.2. The tugger train takes up the biggest part of the route, the bakkerskar takes most of the rest and the forklift truck is used once in a while to visit the remaining locations. The route of the forklift truck is hardly part of the actual milk-run. There are a lot of constraints and logical elements that impact the current route of the milk-run and will impact any future routes that might be proposed. In this sub-chapter an elaboration on the route of the milk-run is provided. This is done visually by illustrating the path of the vehicles on an overview map and zoomed in maps. Added on this is a written explanation of the route and why it is the current way. The maps also provide locations of the stations and substations that are in place for delivering and picking up products.

Stations and substations

Stations are on-site locations where racks and/or cupboards or other storage room is created for the milk-run to drop off the products that are brought along with it and to pick up products from departments that may leave these departments. Drop-off is done in so-called supermarkets, pickup is done out of shop stocks. Each workcell has one or multiple stations that they can use. Products that are requested by the milk-run are assigned a station that belongs to the request. Sometimes a station has some substations beneath it. Meaning that there are a few locations very close to each other, but are separated administratively. The reason for this can be found in the purpose of each substation. For example, one substation is used for picking up regular products from a department, another is used for picking up waste products and a third one is used for delivering products. It can also be that each substation is a consecutive step in a pull-flow system, so each substation follows up on the previous one and each substation therefore has different needs (and different in- and outputs).

2.2.1 Full route

Figure 4 shows the combined route of the milk-run. The following arrows are present on the map, which indicate a route:

●	Tugger train	Yellow
●	Bakkerskar	Light blue
●	Forklift truck	Dark blue
●	Walking	Red

Furthermore, the yellow circles indicate a location of a station.

The vehicles of the milk-run are parked in the logistics centre at station 1. This is where all three vehicles start their part of the route and end up again. The route of the milk-run starts with the tugger train route. Then the bakkerskar takes over. And lastly the forklift truck occasionally drives its route. The milk-run visits different workcells and thus different buildings. Maps that are provided zoom in on different elements where necessary.

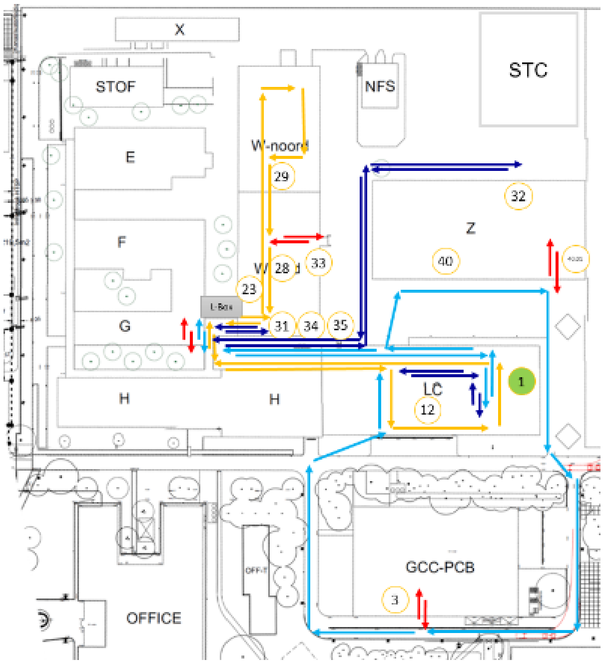


Figure 4 - Full route

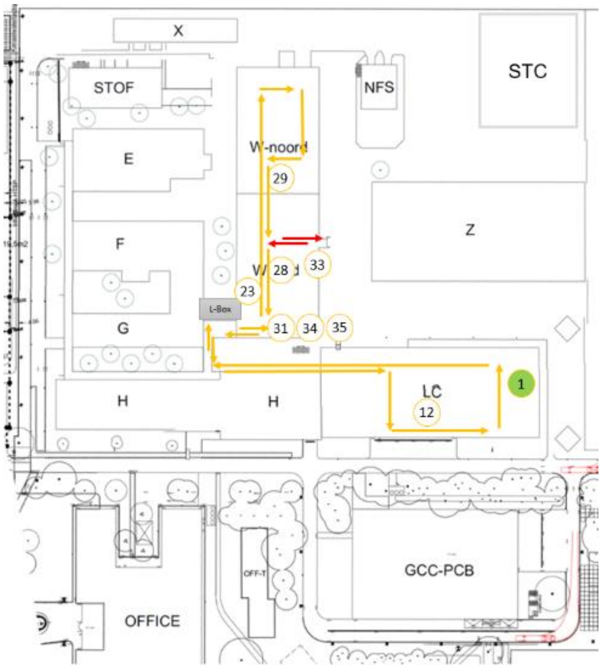


Figure 5 - Tugger train route

2.2.2 Tugger train route

The first route that is taken is the tugger train route, which is shown in Figure 5. The tugger train passes through the building by driving with all its wagons and carts and parking on certain areas. When it is parked, the suspension of the wagons can be lowered and the cart on top of it can drive off. This route starts at station 1 and drives through the LC towards the levelling box. Here, the train is parked and the operator picks up the necessary Kanban cards and places them in the corresponding boxes attached to the tugger train.

Then, the first cart is driven off the wagon and taken to station 31 on the ground floor, which is the first station of W0 (Figure 6). This station is equipped with multiple substations for pickup, regular drop-off and small volume drop-off. This is followed by taking the same cart into the elevator and then visiting stations 34 and 35. These stations are located in the same hallway, so are easily combinable. After these stations, the cart is taken back down in the elevator and is put back on its wagon.

The route continues along station 23 and then two alternatives are possible. The first option is parking at station 28, to visit this station and then taking a cart up the elevator to station 33. Then returning the cart and driving along station 29 and the big hall of W0, where substations of 29 are located to deliver and gather Kanban cards. The second option is turning these two around by first visiting the big hall of W0 and then parking at 28 and visiting station 33. Station 33 has a lot of substations as it covers most locations of the workstations of W1 (Figure 7). To enter the station, a jacket has to be put on and a quick daily test of electricity charge has to be carried out. Then the doors can be opened and the cart can enter the shop floor. Substations that are present at this station are: 33.20, 33.21 & 33.22, which are directly behind the door to the left. 33.23 is slightly further in the workcell and 33.96 & 33.97 are located on the right of the door.

Station 29 (also called 29 and 30) is located at the entrance of the big hall in W0. This station has a lot of space for small items, but is also designed to gather big amounts of pallets and larger products. Substations of 29 are located further in the hall. 29.07, 29.09 & 29.11. At these stations Kanban cards have to be gathered, but it is also possible that material has to be dropped here.

After either of the alternatives, the train drives back along station 23 and 31 (without stopping) and parks at the levelling box to put in the gathered Kanban cards. Lastly, the train returns to the LC where it deposits the picked-up products at station 12 and then ends its ride at station 1.

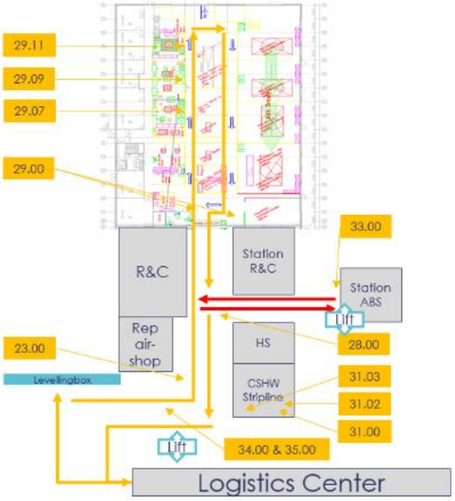


Figure 6 - Building W0

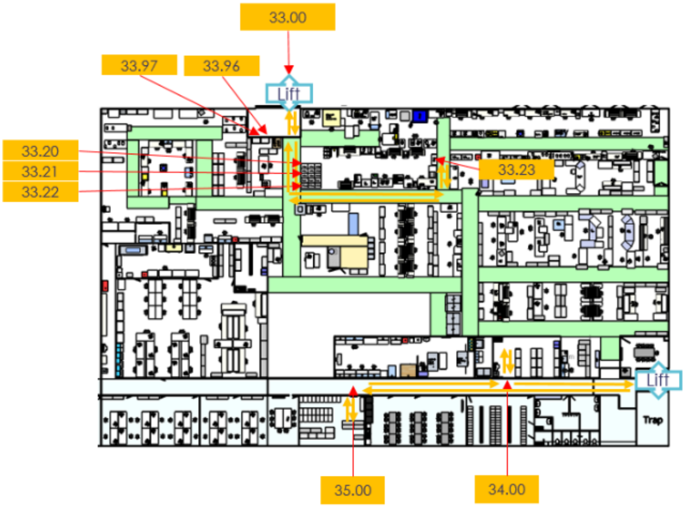


Figure 7 - Building W1

2.2.3 Bakkerskar route

After the tugger train has parked at station 1, the operator will switch vehicles and enter the bakkerskar and start the route shown in Figure 9. Before entering he checks whether he picked up products for locations of the bakkerskar during the run with the tugger train. If this is the case, these products are loaded into the bakkerskar. Now the bakkerskar is ready to go.

The bakkerskar visits the locations on-site that also require relatively small products next to the stations inside building W. These buildings currently are Z and GCC-PCB and the new building STC will also require those products in the future. The different operators of the milk-run have different ways of driving along the buildings, but do visit all of them in each drive. The most common way of driving is by first visiting station 40 and 40.01, which are located in building Z (Figure 8). For station 40, the location of the shop stock/supermarket is directly behind the door. For station 40.01, the location is slightly further inside the building. Building Z, however, will be fully redesigned to house different products. This way the aim is to get rid of the current stations and create a layout where products can be dropped off at the workstations. This would mean that the vehicle will have to enter the building and drive a route through it. For now, it only stays outside.

Figure 8 - Building Z

After visiting building Z, the bakkerskar drives towards GCC-PCB, where station 3 is located. For this station, the bakkerskar is parked and the operator goes inside to gather products and drop off products if needed. Station 36 is also located in building GCC-PCB, but this is not an actual part of the milk-run. It can happen that products have been requested by this station. In this case, the operator loads it onto the bakkerskar anyway. Kanban cards are also collected and delivered at both buildings. After building GCC-PCB, the bakkerskar drives back to the LC. It first must drop off the Kanban cards at the levelling box, so it visits this location. And afterwards, it returns to the parking location inside the LC and the milk-run is generally finished. There is however also a chance that the forklift truck must be used for the milk-run.

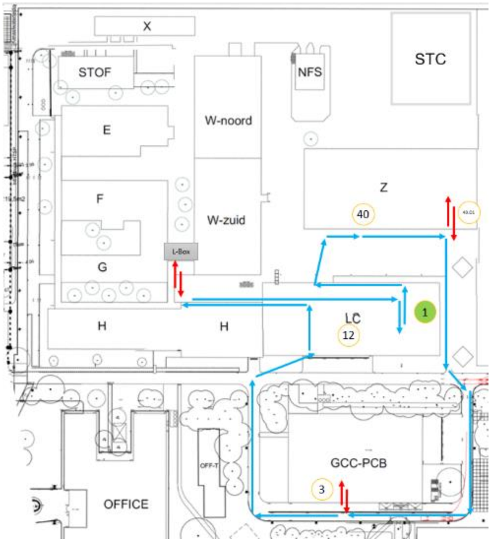


Figure 9 - Bakkerskar route

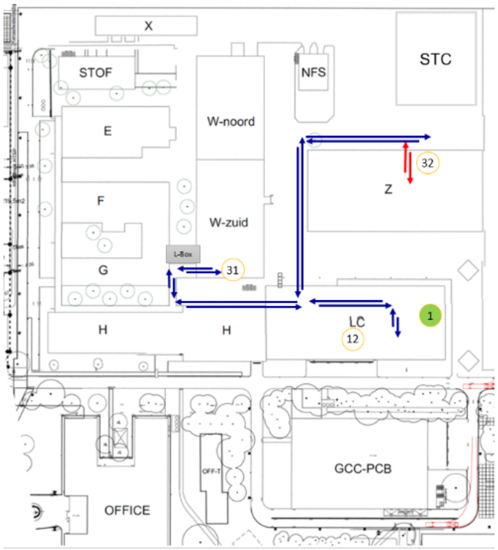


Figure 10 - Forklift truck route

2.2.4 Forklift truck route

The forklift truck is the last element of the milk-run route (Figure 10). This forklift only visits two dedicated stations. The first one is station 31 in building W0, where a lot of products are stacked on a pallet and placed outside of the station to be picked up by the forklift truck. The other station is station 32 in building Z. This station is not visited by the bakkerskar, so only large products are brought and picked up here by the forklift truck. The forklift truck is thus not always part of the milk-run. A lot of intern transport is done by the forklift, because products are too big or too heavy. This transport is not a part of the milk-run and will also not become part of the milk-run, because of the same reasons.

2.3 Workcells and demand

There are thirteen workcells within Thales Hengelo that together create all the products and carry out the processes that are available at Thales to support this creation. Some workcells are situated at one location and some are spread out over multiple buildings. This sub-chapter elaborates on the different workcells and locations that are present within Thales and their respective usage of the milk-run.

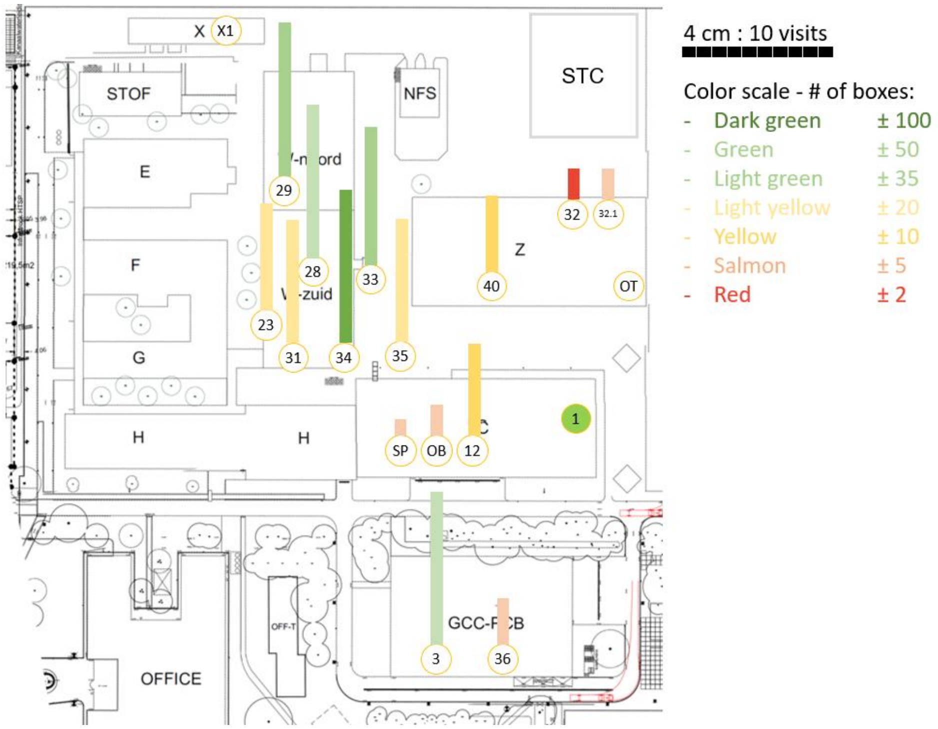


Figure 11 - Weekly data of milk-run pickup and delivery

Data has been gathered about the pickup and delivery on all the stations. The overview of this data is added to Appendix B. For this data, operators of the milk-run have kept track of all the boxes they have delivered and picked up at each station. During one week, the milk-run drove twice a day, which means that ten runs have been recorded. These are visualised in Figure 11. The data has been reviewed by the operational manager and an employee of the logistics centre to verify whether the week embodied a representative week or that demand was higher or lower than usual. They both indicated the week was very representative concerning the demand they had. There was a slight issue with incoming goods, which meant slightly less products were taken on the milk-run. This was not very significant, but relevant to keep in mind for further use and elaboration on the data. It is apparent that pick-up is always less than delivery. Figure 11 shows the two most important factors of the data:

- The frequency that the milk-run has visited a station. Which means either a product was delivered or picked up or both. The length of the bars indicates the number of visits out of ten runs. So, a bar of four cm means the milk-run has visited this station in each run, e.g. station 29.
- The number of boxes that have been picked up or delivered. This concerns the total amount of boxes that have been picked up or delivered during the week. The colour shows the number that belongs to the number of boxes, ranging from 0 to 100 and from red to green respectively.

The figure clearly shows that there is a lot of diversity between stations regarding the two factors above. Some stations get visited (almost) all of the runs and some are only visited twice a week. A visit means that at least one box is picked up or delivered. The same goes for the number of boxes that are delivered and picked up; There are stations that have a high demand, like 88 boxes for station 28 (building W0). And there are stations where the demand is very low, like station 32 (building Z1), which only received one product and sent out one product during the week. The main division of visits and number of boxes picked up and delivered per building are shown in Table 2.

Building	Visits	Boxes
LC	10	21
W0	10	141
W1	10	157
Z0	5	27
Z1	2	5
GCC-PCB	10	40
X1	0	0

Table 2 - Visits and number of boxes of a weekly milk-run per building

There is no data available which represents the demand per station throughout the day. That is why the data that is presented in Figure 11 will be the basis for the representation of the demand. This data however is clustered after a week of measuring. No time stamps concerning the requests were added to this. To include this time element of the demand, a dataset has to be considered to create a somewhat representative situation concerning the time element of the demand distribution. We looked into an existing dataset of the requests that the internal transport received for 1-hour and 24-hour jobs over 250 working days. These requests are shown in Figure 12. The left graph shows the demand for 1-hour jobs, the middle graph shows the demand for the 24-hour jobs and the right graph shows the combined demand. Each timestamp on the x-axis represents a period of an hour starting from the mentioned time. Combining these two data sets will provide a solid basis to analyse the situation.

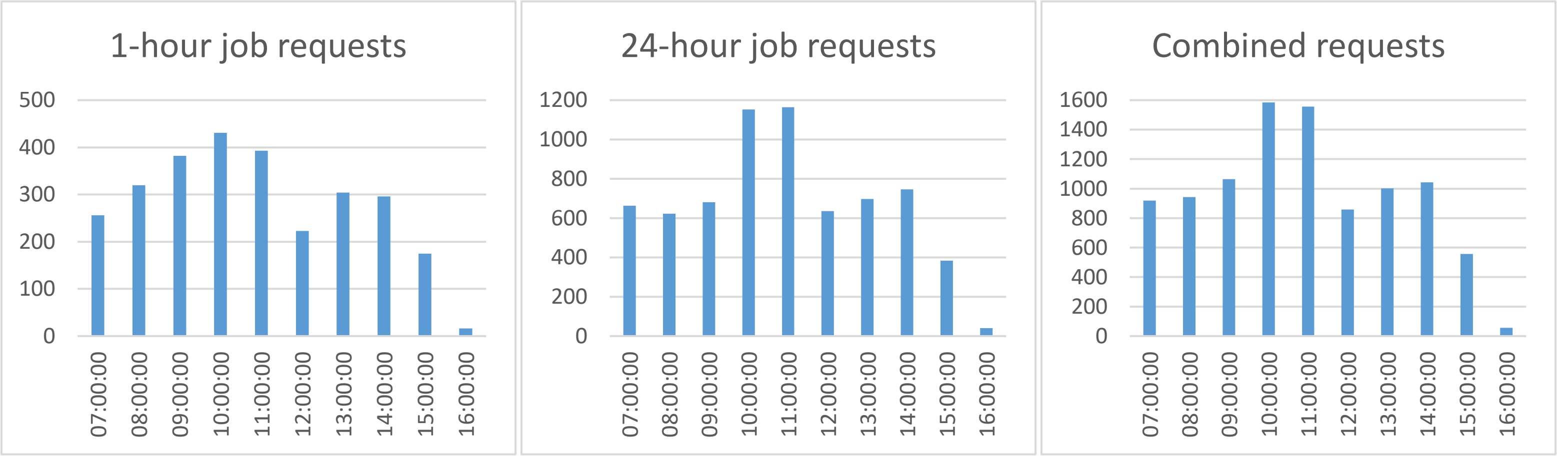


Figure 12 - Requests demand distribution over 250 working days

To create a full picture of how each workcell deals with the milk-run, interviews have been carried out with their respective workcell-leader. The key take-aways of these interviews are provided in Appendix A. together with an overview which shows what workcell corresponds with what station(s). The most relevant findings concerning the route through/along the (work)station(s) are provided:

- **Sensors –** The shop floor of this workcell will increase as some other work lines are transferred to another location. More big radars will be built here, which might have implications on the route of the milk-run through the hall. The lay-out is still unclear, but it is not desirable to drive through the building very often.
- **Combat Systems Hardware (CSHW) –** A lot is sent by pallet, which should also be possible to take with the milk-run instead.
- **Aerospace & Building Blocks –** Currently, it sometimes takes too long for products to arrive. Having more frequent milk-runs would tackle this problem.
- **Mechanics & Inspection –** Sometimes delivery to other departments is done by themselves instead of the milk-run, because this can take too long and the (internal) customer might be waiting for it.
- **Track Sensor Systems –** Is transferred to the new building, where driving along the workstations is a possibility and is desired.
- **Search Sensor Systems –** The workcell search sensor systems is divided over two buildings. In building W0, inside the high hall, the STIR (one radar) is made. This will be relocated to the new building. The other part of the workcell is located in building Z. This building layout will be redesigned to create more capacity for these radars. Currently, the milk-run stays outside for this workcell, but the new layout will allow the train to go inside and deliver directly at the workstation.
- **Optronics *(Micro-Electronics, Optronics & Subs)* –** This building is quite remote from the rest of the buildings. That is acceptable, because this building barely has to be visited by the milk-run as their demand and supply is low.
- **Installation** – The new layout of building Z desires a situation where the milk-run delivers products to the supermarkets on site.

2.4 Information flow

To get more insights in the logistics flow of the milk-run, we look at the information flow around it. Part of the problem cluster is the problem of products getting lost and the lack of traceability of products. This lack of traceability is present, because no information systems are used to track products. This is one of the direct main causes of the action problem. It is therefore useful to dive deeper into this aspect of the problem. To provide a clear insight in the flow of the products before the start of the milk-run and the corresponding information around the distribution of products, a swim lane flowchart is created in Figure 13. This flowchart consists of five layers, which represent actors in the process. The three layers above represent employees who carry out tasks in the picking process. The fourth layer represents Oracle, the system in which product flow and inventory is kept track off. This layer contains an icon of a database when an update in the data has happened because of the actions of one of the three layers above. Lastly, the fifth layer represents the operator of the milk-run.

The process starts off when a job order is received. A job consists of multiple products that should be delivered to the same location. Then, the two tasks of the planner follow; First, he starts (and creates) the job. This means that the job is registered in Oracle. The job contains a number of products which should be present in the warehouse when starting a job. A location tag is added to the corresponding products of the job. After the job is started, the planner releases the job to be checked by office. Office will do a final control on the job list and will check for inconsistencies or other elements that are not correct concerning the job in Oracle. Once these inconsistencies are removed, office will set the job to 'pending', which means that it is free to start by a picker of the Central Warehouse (CW). After these steps, the rest of the picking process is fully carried out by the picker. The first activity of the picker is to select one (or more) job(s) to pick and then link and send this to the scanner. The scanner is a handheld device which scans products and can create stickers that should be attached to each (bundle of) product(s). This is also the next step in the process. After each product of the job has been scanned and a sticker has been put onto it, the physical picking process of the job is completed. Then, the LPN (License Plate Number) is linked to the job, which means that the job can be 'drop loaded'. This implies that the inventory of the CW is updated, so all the products are removed from this Oracle environment and are sent to the workstation, where they will be delivered by the milk-run. Subsequently, the last step for the picker is to load the picked job onto the assigned vehicle of the milk-run and this picking process is fully completed. Finally, the milk-run operator drops off the products at the corresponding station. Here, no data is registered concerning this drop-off, but the process simply ends when the products have been delivered.

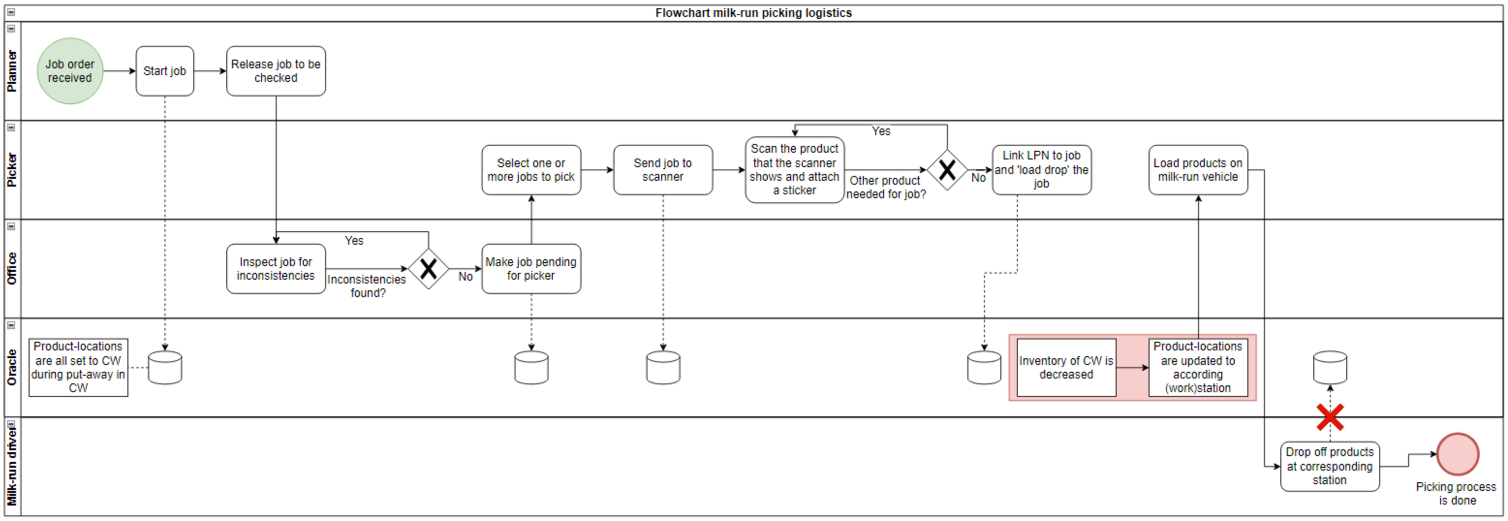


Figure 13 - Information flow diagram picking process

Lost products

The last two steps that are present in the layer of Oracle have been marked red, because there is a discrepancy between the needs of the pickers, planners & others and the reality. As described, currently the products in Oracle will be transferred from the CW to their assigned workstation. This is already done before the milk-run has even taken off. This leads to wrongful assumptions of the locations of the products that are brought by the milk-run. There are multiple elements in the timeframe of just before loading the products on the milk-run vehicle until delivery on a workstation that can go wrong:

- The product might be delivered at the wrong workstation by the milk-run operator.
- The product might not be delivered at all.
- The product might not be loaded onto the milk-run at all.

This leads to discrepancies between the actual location and the location that is shown in Oracle. These discrepancies cause products to get lost in the process, while nobody knows where they are. This is also caused because there is no data log in the step of the milk-run operator. Sometimes these products are found days or weeks later at other workstations or in the CW.

> *Planner: "We* (read: the planners) *receive the complaint 'Not enough delivered' by employees on the shop floor around 5 – 10 times a week. Most of these mistakes are made in the picking process* (read: picking too few or wrong parts). *Around once a week this code is due to a part that has gone missing in the process. The parts that are not delivered (enough) are often parts that are already low on stock, which is inconvenient."*

The warehouse operational manager confirms the quote of the planner above by stating that "there is a weekly search for a lost part or product". These parts/products differ from expensive to cheap and unimportant to crucial elements for the continuation of a certain task/product line.

Lost products sometimes cause major delays for departments as they might be waiting for certain products to continue their production processes. Such pauses in the production process severely impact the way of working for some departments. As for some products, there are quite extensive set-up processes that are required before working on a type of product. When a product that is needed to continue working is not delivered, because it is lost, this means the current product can not be finished and the employee must start to work on another product. Then, when the product is found, the same set-up process has to be done all over again. It can also happen that the product the employee was working on, was the only required product at that time. This means that the employee cannot continue working at all. Both of these scenarios are very costly for Thales as the employee wastes its working time. It can also happen that a part is lost for so long that a new order of the part is placed, when the old part is then not necessary anymore, this is a massive waste of products, time and money. All of these lost parts are due to a lack of traceability throughout the picking to delivering process. A recommendation on this topic should be provided to tackle this problem.

2.5 Other transport

Not only the milk-run influences all the KPIs of the process that are linked to the action problem. There is also other transport that affects elements like on-time delivery and lead time of products and the productivity of the whole transport team in Thales. Therefore, this section describes all the remaining relevant aspects of the transport within Thales that are not specifically/directly linked to the milk-run. This provides context for the milk-run in order to establish what the impact of an improved milk-run can be. For these types of transport, we only look at relevant transport that can be affected by changes in the milk-run. E.g., the transportation of a radar of ten meters is out of scope. Relevant transport aspects that impact the milk-run are: 1-hour jobs, 24- hour jobs, direct delivery receipts and the products that are delivered by the forklift truck. All of these aspects impact the usage of the milk-run in some way.

1-Hour jobs and 24-hour jobs are jobs that are requested by employees when they desire a product sooner than the agreed SLA of three working days. As the name says, 24-hour jobs must be delivered within 24 hours after the request has been submitted. Therefore, when such a job is picked, it now has two possibilities to join a milk-run before the 24 hours have passed. That is why most of these jobs are loaded onto the milk-run. For the 1-hour jobs, this becomes less likely as they can only be taken with the milk-run if they are requested shortly before the milk-run departs, but not too short as they still have to be picked. When the frequency of daily rides from the milk-run increases, the number of 1-hour jobs that are brought by the milk-run will also increase. This would relieve a lot of work for the logistic workers as they would not have to travel for separate products as much as they do currently. The warehouse operational manager even proposed to get rid of this service when the milk-run would travel four times a day. But even without getting rid of this service, a relieve on this would already be very helpful. The logistics department has set a limit for both services as otherwise it would disrupt their daily operations too much. For 24-hour jobs this limit is 50 jobs per day. For 1-hour jobs a limit of ten is put in place. The warehouse operational manager says the following regarding these limits:

> *"There are days that we have more than 50 requests for 1-hour jobs. This is too disruptive and keeps us from our other tasks. … For both services, 50% of the time we surpass the limit of number of requested jobs per day."*

These numbers are once again verified by the numbers of the week this manager is talking about: there have been 58 1-hour jobs and 255 24-hour jobs. This clearly stresses the need for a relieve on this service and the milk-run can assist in this. Currently, delivering a 1-hour job takes four minutes on average (between one and five minutes overall).

Lastly, there are products that are transported by the forklift truck. These can be divided over three segments. The first segment consists of products that are transported by the forklift truck and belong to one of the forklift stations of the milk-run (31 & 32.1). These are part of the milk-run. The second segment consists of products that are simply too big and/or too heavy to be transported to the other stations of the milk-run with the use of the tugger train or the bakkerskar. They get transported by the forklift truck since there simply is no other way of transporting them. The last segment are the products that are now brought by the forklift truck but can be transported by the milk-run when looking at their size and weight. These products are often stacked on pallets and then become too heavy to be transported by the milk-run. This segment can be incorporated into the milk-run when more attention is paid to this, and products do not get stacked but get picked up by the milk-run separately.

2.6 Objective Key Performance Indicators

The model that will be created is focused on improving the current situation as much as possible. There are several KPIs that can be monitored to fulfil this goal. Some KPIs will work against each other, so weights have to be given to the KPIs or a balance has to be created in another way. First of all, it is important to provide a clear elaboration on each of the KPIs, to highlight the objective of this research.

On-Time Delivery & Lead time 1-hour jobs, 24-hour jobs & standard jobs

We want the lead time for products to become as short as possible, so production gets delayed less. So, visiting a station more often would be beneficial for some of the workcells. The way this KPI can be organised, is seeing whether a 1-hour job, a 24-hour job and a standard job has been delivered on time or not. For some jobs it might be beneficial to see how quick a job has been delivered. But for most of the jobs it is more important to have a yes/no on the On-Time Delivery (OTD). Measuring the ratio of how many jobs have been delivered on time compared to the total products in the 1-hour, 24-hour and standard jobs will provide a clear percentage of the status of this KPI.

To create the full image of this aspect, we can measure the time between a product has been ordered and the product has been delivered. We classify this as lead time. Delivering 1-hour jobs on time is more important than delivering the 24-hour jobs on time, which is in its turn more important than the standard jobs.

Productivity (Output/Resources)

Another KPI is the productivity. The productivity is the ratio between the used resources and the created output of the full internal transport. On one hand we have the workload that goes into the internal transport. This contains the workload for the operators of the milk-run. Driving more and longer routes will result in a higher workload for these operators, so more resources are used in this case. In contrary to this, visiting the shop floor more often will result in a lower workload for other employees as they do not transport products themselves anymore, so lower resources on this side. The same goes for the transportation of 1-hour jobs. We want to increase the workload for the operators from the milk-run as little as possible, but it is worth it to increase this workload if that means that the workload of transporting products for other employees and the delivery of 1-hour jobs decrease even more. This creates a difference in output and improves the overall productivity. Operators of the milk-run are responsible to transport products and when they do this more often, other employees can carry out their own tasks more often.

The productivity KPI can be measured in a ratio of output divided by resources. Output can be expressed in the movement of parts. Resources can be measured in a time dimension where the costs for the time of operators are taken. Other resources are the transportation costs of the vehicles. The KPIs do not necessarily support each other, so a trade-off might have to be made between them. In order to create a right balance between these KPIs, weights or costs can be assigned to come to an objectively optimal solution. So, to create a balance and therefore an optimal point, we must provide a common factor to each KPI to scale them and measure them relative to each other. Attaching a cost element onto each KPI is the most doable for all of them.

Main goal

In order to connect the research goal/objective to the KPIs, a main goal is set up. The research goal of developing a plan for the integration and optimization of the milk-run throughout Thales, has most of the elements of the KPIs in it. By designing this plan, the action problem of production lines standing still while waiting on products will be tackled as well. Therefore, the main goal in terms of KPIs can be established in the following summarized manner:

Maximize Productivity while sticking to the established **On-Time delivery level** and **lead time targets**.

2.7 Conclusion

In this chapter, we investigated the current situation at Thales by answering the research question *'What is the current situation of the milk-run of Thales?'* and its sub-questions. This section discusses the answers to the sub-questions.

How is the process around the current milk-run organised?

The current situation of the milk-run of Thales is quite diverse and lacks a plan for integration throughout the production facilities. This plan should concern multiple aspects of the milk-run. The milk-run now, drives twice a day. Next to these regular milk-run operations, where products are picked from the warehouse and sent on the milk-run and reverse, there are also some other implications that the milk-run is partly responsible for, like 1-hour and 24-hour jobs. The rest should be completed within the SLA of three working days. Thales makes use of a level pull flow system with the support of kanban cards and a levelling (Heijunka) box. This system is utilised for some products that are produced in flow. Kanban cards help to keep an overview of the demand, which is visualised at the levelling box.

What is the current route of the milk-run?

The milk-run is divided over three different vehicles. Most of the products on the milk-run are transported by the tugger train, which is a locomotive with four wagons behind it. This vehicle drives through the main building and picks up and delivers the most products for the most stations. The bakkerskar is the second vehicle of the milk-run and delivers and picks up at the stations that are located outside the main building. Lastly, the forklift truck is incorporated into the milk-run and delivers and picks up products that are too big or heavy for the other two vehicles. Because of the division over three vehicles, there are also three routes that are taken for each milk-run. The fork-lift truck route is not always used for every milk-run, but the other two are.

How are the workcells organised and what is their demand?
Every workcell within the production facilities has its own needs and its station(s) where products are picked up and delivered. Some stations require a lot of products and some stations barely require any. Taking into account the differences between them is very important for the setup of the milk-run and its route.

What does the process of keeping track of the products look like?
There are some traceability issues concerning the delivery of products in the milk-run. This is due to a discrepancy between the picking process and the delivery process at the shopfloor. This discrepancy is relevant to investigate, but is separate from the routing of the milk-run.

How is the other transport within Thales organised?
The logistics department, which is responsible for the internal transport, has set a limit for both 1-hour and 24-hour delivery services as otherwise it would disrupt their daily operations too severely. For 24-hour jobs this limit is 50 jobs per day. For 1-hour jobs a limit of ten is put in place. This limit is crossed around 50% of the time. Relieving stress at these jobs by creating a better organised milk-run, will also relieve other parts of the internal transport and will help to tackle the action problem of production lines standing still.

What are the KPIs that are relevant for the milk-run?
The standing still of the production lines is reflected in the KPIs that are relevant for the milk-run. These KPIs are the on-time delivery and lead times of the three types of jobs: 1-hour jobs, 24-hour jobs and regular SLA jobs. The productivity is another important KPI of the milk-run. This is reflected in the output that the milk-run/internal transport creates divided by the resources it takes to deliver and transport all the products for the internal transport department.

3. Literature review

In this chapter, the literature is reviewed by answering the second research question: *"What methods to create and optimise a model for the milk-run are available in the literature?".* The chapter starts by addressing the basics of a milk-run in Section 3.1. Subsequently in Section 3.2, the Vehicle Routing Problem is introduced, where the mathematical formulation is presented. Lastly, Section 3.4 elaborates on the VRP by proposing extensions to fit the milk-run situation. Section 3.5 provides a conclusion of the findings of the literature review.

3.1 Milk-run logistics

Reducing logistics costs has become a vital area for manufacturing companies in creating profits. Balancing the workload on the shop floor while ensuring a continuous and smooth running of production are aspects to keep in mind when aiming for reductions. Adequate vehicle routing planning can be very helpful for the logistics operation management in this area. As a logistics model to transport products, the milk-run has attracted wide attention (Mei, Jingshuai, Teng, Xiuli, & Ting, 2017).

The concept of milk-run logistics originates from the dairy industry. The idea concerns a transportation network where all materials for the input and output required by several stations are covered by one vehicle that visits all these stations and circulates according to a pre-defined schedule. This transportation concept is cost-effective when the volume of the input or output of each single station is essentially smaller than a truckload. The milk-run concept is commonly applied in internal plant logistics to transport raw materials/parts, finished products, and other waste between manufacturing and assembly stations and the warehouses of the plant (Baudin, 2005).

A milk-run is a manually operated, (cyclic) transport system delivering raw materials and finished goods, using a fixed route and time schedule. According to Droste and Deuse (2012) a milk-run cycle consists of the following steps:

1. Loading material on means of transport
2. Transporting material to the point of usage
3. Unloading material at the point of usage
4. Loading empties on means of transport
5. Transporting empties
6. Unloading empties at return location

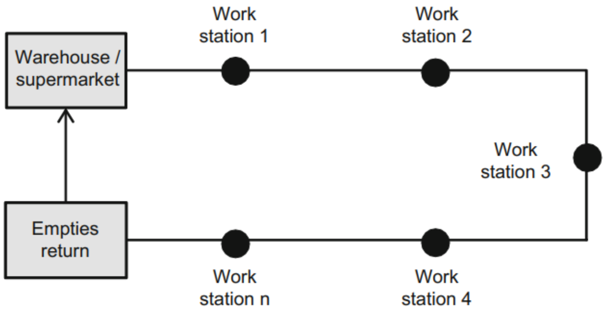


Figure 14 - Basic structure of material provision by milk-run (Droste & Deuse, 2012)

The return of empties is a key element of milk-runs letting material and boxes circulate synchronized with production's consumption. Figure 14 shows the basic structure of material provision by the milk-run. For other scenarios of the milk-run, the 'empties' can be replaced by regular products or parts that must be redirected to the warehouse. This situation is very comparable to bringing empties, only the weight differs. In this case, the 'empties return' is replaced by a drop-off point in the warehouse.

Milk-run systems are all about logistics support for the supply chain. Implementing a milk-run system results in reduction in cost of transportation, travelling path and fuel consumption and there are multiple other advantages of using a milk-run. In general, the reasons why milk-run logistics have been broadly taken into usage are reduction in transportation costs, improvement of the assembly manufacturer's production line and greater accuracy of JIT goods delivery due to synchronization, improvement of the vehicle loading rate and shorten the total distance travelled, it reduces the risk of product quality problems and it changes logistics strategies and thus reduces investment risks (Brar & Saini, 2011).

There are situations where a milk-run might not be a very effective solution. Milk-runs can be very powerful but are not applicable to every scenario. The usage of a milk-run is effective for products that are frequently used in moderate quantities. It is also effective for locations that are relatively close to each other. Milk-runs are not effective when a product is needed in multiple (truck)loads, a product is only required sporadically , in small quantities, from a supplier who does not provide any regular used items or the location of the supplier is far from the others (Baudin, 2005).

3.2 Milk-run modelling

There are multiple ways to model a milk-run. The first way is by using a Traveling Salesman Problem (TSP). This is a concept for organizing transportation routes and analyzing which route should have the first pick up and which should have the next pick up from a range of suppliers. The Travelling Salesman Problem is a problem for a salesman journeying between multiple cities to sell his goods. The objective is to traverse each city efficiently, aiming to cover all destinations in the shortest time possible, visiting each city precisely once along the route. The solution entails determining the optimal path that achieves this goal, minimizing both distance and time spent during the salesman's journey (Horowitz, Sahani, & Rajasekaran, 2007). The TSP can be formulated as an integer linear programming to solve the routing problem of the Milk Run. The Traveling salesman problem is being used in many fields nowadays. Some of its applications are manufacturing of microchips, vehicle & packet routing in GSM, drilling printed circuit boards etc. In other words, say we have a set of n number of cities, and then we can obtain $(n-1)!$ alternative routes for covering all the n cities. Traveling Salesman Problem is to procure the route which has the least distance.

There are various methods to solve a TSP. Three methods will be explained. The first is the method of total enumeration. This method evaluates all potential routes in the whole solution set. This is a very ineffective way of working. The second method is branch and bound, which is one of the oldest and most used algorithms for the TSP. It uses an upper bound and a maximum lower bound of the objective function and calculates solutions in between and disregards the rest, while updating these bounds. The third method is the algorithm of Clarke and Wright. The initial solution assumes that each location is supplied individually and always has a return to the starting base. The fundamental concept revolves around calculating cost savings by incorporating additional locations into a circular route. A notable advantage of this algorithm is its ability to accommodate additional constraints commonly encountered in practice, such as the optimization of multiple circular routes, the utilization of various vehicles while respecting their capacity limits and other logistical requirements (Dahiya & Sangwan, 2018).

The other and primary way of modelling a milk-run is by using a Vehicle Routing Problem (VRP) model. The majority of the studies on milk-runs in the literature uses a VRP to solve the posed challenge. According to the characterization that is provided by Gyulai, Pfeiffer, Sobottka, and Váncza (2013), a milk-run schedule can be considered as a special instance of the VRP (with time windows and a limited number of vehicles). A VRP is a process where the aim is to design a set of some minimum cost vehicle routes through some customer locations, so that each route starts and ends at a common location and some side constraints are satisfied (Laporte, 2007). Many practical transport logistics and distribution problems can be formulated as a vehicle routing problem whose objective is to obtain a minimum-cost route plan serving a set of customers with known demands. In general, each customer is assigned to exactly one vehicle route and the total demand of any route must not exceed the vehicle capacity (Lau, Sim, & Teo, 2003). An example of a common application mentioned is thus the milk-run, but also the newspaper round and food delivery can be modelled with the help of a VRP.

In 1959, the first research was done regarding the VRP. Since then, the VRP is one of the most studied problems in Operations Research. Braekers, Ramaekers, and Van Nieuwenhuyse (2016) mention in their paper that the current VRP models are very different compared to the early 60s. Nowadays, the goal is to incorporate real-life difficulties and complications into the problem. Examples of these complexities are traffic congestions, time windows for pickup and delivery and input information that changes over time. These are only some of the variants on the VRP. There are a lot more with each having their own purpose and way of modelling. Combining such variants is the key to represent a certain problem as close as possible (Kumar & Panneerselvam, 2012). In Mei et al. (2017), a close example to the situation of Thales is provided, where the VRP is set up to solve a milk-run problem. They use time windows and vehicle load constraints to represent their problem. Solving a VRP can be done with the help of algorithms. Mei et al. (2017) combine the VRP with the C-W Algorithm. This algorithm is fit to solve a TSP, but does not consider constraints, so cannot solve a VRP. This way it can be used to solve single type of vehicle routing problems in the whole model. Algorithms can play a crucial role in solving Vehicle Routing Problems (VRPs) by providing efficient methods to find optimal or near-optimal solutions. Algorithms can help in certain forms. The most common forms are optimization, heuristics and metaheuristics:

Optimization in Vehicle Routing Problems (VRPs) aims to find the best routes for vehicles while minimizing costs or maximizing efficiency. Techniques include exact algorithms, which aim for the globally optimal solution but are computationally expensive and mathematical programming formulations, such as linear programming, which model VRPs mathematically to find optimal or near-optimal solutions.

Lenstra and Kan (1981) studied the complexity of the VRP and proved that the problem is NP-hard. Because of this, exact algorithms can be used to solve smaller problem instances. If the problem instances become too large, (meta)heuristics are more suitable to solve the VRP. Most of the real-life problems are too large. Therefore, these heuristics are used.

Heuristics are rule-of-thumb techniques used to quickly find good solutions to complex problems like VRPs. These heuristics efficiently explore the solution space to find feasible solutions without exhaustively searching all possible routes. Common examples of heuristics used for VRPs and applied within cases of a milk-run are:

- Nearest Neighbor (NN), which is a method that builds a route solution by finding the closest location to the previous point. It selects the next location based on the shortest distance from the last location (Wicaksono, Puspitasari, Ariyandanu, & Hidayanti, 2020).
- Insertion Heuristics, where tour-building strategies begins by setting up each route based on various criteria. During each iteration to insert a new location into the current incomplete route, the best choice based on the criteria is made and inserted between two consecutive locations along the route.
- Saving Heuristics, where the heuristic starts with n separate routes, each assigned to a specific vehicle for serving customers. The parallel implementation of this heuristic for constructing tours involves adding a unique link between two end customers in partially formed routes during each iteration. This addition is guided by a metric indicating potential cost savings (Solomon, 1987).

Metaheuristics are higher-level strategies used to efficiently explore large solution spaces and find good solutions. In VRPs, metaheuristic algorithms provide frameworks for iteratively improving solutions through exploration and exploitation of the solution space. These algorithms efficiently explore diverse regions of the solution space, escaping local optima to find high-quality solutions. Many of the most successful meta-heuristics for the large VRPTW instances are based on some form of parallel computation (Kumar & Panneerselvam, 2012). Some major examples of metaheuristics used to optimize milk-runs are:

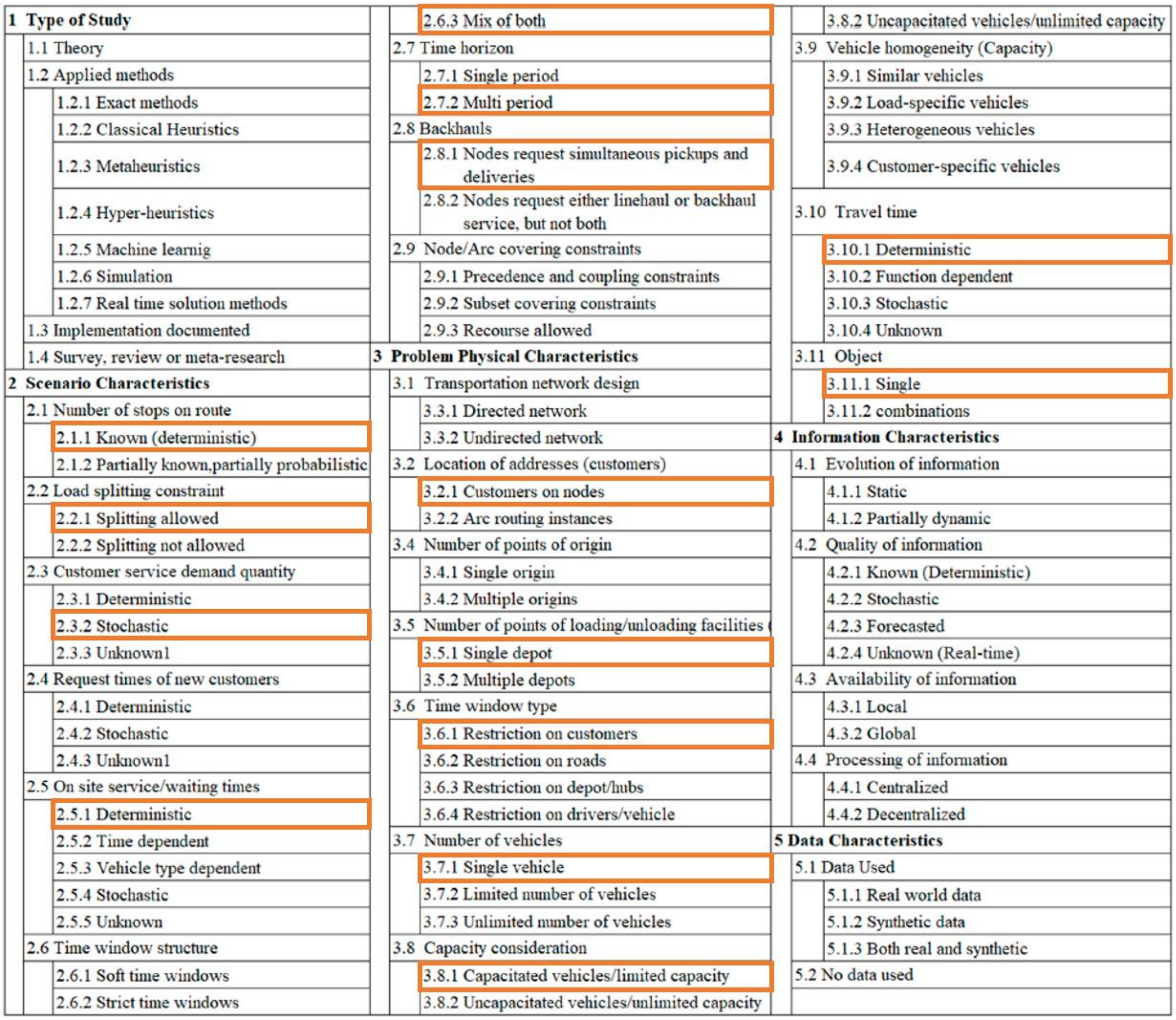
- Simulated Annealing (SA) uses a stochastic method to explore and transition to new solutions, known as neighborhood solutions. If a superior neighborhood solution is found from the current solution, it replaces this current solution. SA also accepts moving to lower neighborhood solutions with a certain probability to avoid local optima, where the temperature gradually decreases during the search. Higher acceptance probabilities are ensured at higher temperatures and vice versa towards the end of the search (Kuo, 2010).
- Tabu Search (TS) is a method where during each iteration, the algorithm explores the neighborhood of the current solution and selects the best available solution as the new current solution. To avoid getting stuck in local optima, even if the new solution is worse, it replaces the current solution. Cycling is prevented by prohibiting revisiting recently selected solutions, managed through a tabu list. Generally it is seen as one of the best heuristics for the VRPTW (El-Sherbeny, 2010).
- Genetic Algorithms (GA) is a technique where a population of solutions is maintained and a reproductive process selects parent solutions from the total population. Offspring solutions are produced which show some of the characteristics of each parent. The fitness of each solution is related to the objective function value, in the case of a milk-run, this can be the total distance travelled, combined with the degree of any constraint violation (Baker & Ayechew, 2003).

Reviewing this section, it can be concluded that using a VRP is very relevant to solve the milk-run problem of Thales. In order to resemble the actual situation Thales is in, some variants on the VRP must be studied and elaborated on to facilitate a well-designed scenario. The combination with an optimization algorithm, a heuristic or a metaheuristic can also be considered when modelling. As metaheuristics are used to efficiently explore large solution spaces and find good solutions, this would be the best addition to a VRP model which has a large solution space. To better understand the description of the VRP, we first dive deeper into the VRP modelling and review variants/extensions afterwards to see how we can exactly match the milk-run of Thales. The adjustability of a VRP model fits perfect for this need.

Throughout the evolution of the original Vehicle Routing Problem, its variants have continuously developed, leading to multiple models proposed in the literature. We can classify the situation with the milk-run in Thales as a certain instance of the VRP. To classify the problem, we use the same method as (Karakash, 2024). He looked at the three-level classification of the VRP models of (Ni & Tang, 2023). The overview of the VRP taxonomy is shown in Figure 15. We have marked the characteristics of a VRP that match with the milk-run of Thales with orange boxes.

For the scenario characteristics, the milk-run has a known number of possible stops on the route, where splitting the demand is allowed. This option in the VRP is known as the Split Delivery VRP and explained in Section 3.4.3. This demand is stochastic in reality, but will be considered deterministic throughout this research as there is a lack of data to be able to model this adequately. The service times are deterministic as they always take up the same amount of time. Due to the different type of jobs, there are both hard and soft time windows. These variants are explained in 3.4.1 and 3.4.2 respectively. There are multiple slots during the day in which demand is present, so the time horizon has multiple periods. Lastly, the nodes (customers) request both pick-up and delivery. Pick-up will be disregarded in the modelling as we can make the assumption (Section 2.3) that delivery is always higher than pick-up, so no bottlenecks are created there.

There are also physical characteristics for the problem. Firstly, for the location, the customers are on the nodes instead of on a routing. There is one single depot, the logistics centre, from which the products are distributed. For the time windows, their only restriction is on the customers as they require the products in certain periods. One vehicle drives at the time, so one vehicle is used, which has limited capacity. Lastly, the travel time is deterministic and the demand is in boxes, so there is one single sort of object. A characteristic that is unmentioned in this overview is the compatibility constraints certain nodes have with certain vehicles. This is explained in 3.4.4, as this is a relevant characteristic for the milk-run of Thales. The information characteristics do not impact the modelling of the VRP so are disregarded in this research.



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

Figure 15 - Taxonomy of the VRP literature by (Ni & Tang, 2023). The orange boxes mark the characteristics of the milk-run of Thales

3.3 VRP Formulation

The VRP has been researched extensively throughout the years. Therefore, a lot of different formulations of the problem exist. We must adapt it in a manner to fit the situation of the milk-run of Thales. First, the basic model of the VRP and its formulation will be discussed. In this paper of Kallehauge, Larsen, Madsen, and Solomon (2005), the formulation is widely accepted and used for VRPs. Brink (2023) has adapted that formulation for the classical VRP formulation, which will be taken for this paper. Section 3.4 shows the extra formulations for the VRPPD. For now, the notation we use is the following:

V	set of vehicles; $V = \{1, \ldots, V\},\ k \in V$
N	set of demand nodes; $N = \{1, \ldots, n\}$
A	set of network arcs; $A = \{(i,j): i,j \in N \cup \{0\}\}$
Cap	vehicle capacity;
d_i	demand for customer i; $i \in N$
t_{ij}	travel time on arc $(i,j) \in A$;
c_{ij}	travel cost on arc $(i,j) \in A$;

Kallehauge et al. (2005) and Brink (2023) define a VRP as follows. The VRP has a fleet of vehicles V, a set of customers N and a directed graph G. The fleet is considered homogeneous, which means that all vehicles are identical. The constructed graph consists of $|N| + 1$ vertices, where the customers are denoted 1,2,…,n and the depot is vertex 0. The set of arcs A, represents direct connections between the depot and customers and among the customers. With each arc (i,j), where i≠j, there is a cost c_{ij} and a time t_{ij}, which may include service time at customer i. Each vehicle has a capacity q and each customer i a demand d_i. It is assumed that q, d_i, c_{ij} are nonnegative integers. The model then contains two decision variables x and s. For each vehicle k and arc (i,j), where i≠j, i≠n, j≠0, x_{ijk} is defined as

$$x_{ijk} = \begin{cases} 1, if\ vehicle\ k\ drives\ directly\ from\ i\ to\ j \\ \qquad\qquad 0, otherwise \end{cases}$$

The other decision variable s_{ik} is defined for each customer i and each vehicle k and it gives the time vehicle k starts to service customer i. In the case that vehicle k does not service customer i, the decision variable does not attain a value.
s_{ik} start of service time of vehicle k for customer i, $i \in N, k \in V$

Kallehauge et al. (2005) state that the goal of the program is to minimize total cost in such a way that each customer is visited once and every route begins and ends at the depot. This goal is shown in the objective function in (1.1). Furthermore, the constraints for this basic model are provided in (1.2) until (1.9). This description is also created by Kallehauge et al. (2005).

$$(1.1) \quad min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$$

subject to

$$(1.2) \quad \sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in N$$

$$(1.3) \quad \sum_{i \in N} d_i \sum_{j \in N} x_{ijk} \leq Cap, \quad \forall k \in V$$

$$(1.4) \quad \sum_{j \in N} x_{0jk} = 1, \quad \forall k \in V$$

$$(1.5) \quad \sum_{i \in N} x_{i0k} = 1, \quad \forall k \in V$$

$$(1.6) \quad \sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in N, \forall k \in V$$

$$(1.7) \quad s_{ik} + t_{ij} - M(1 - x_{ijk}) \leq s_{jk,} \quad \forall i, j \in N, \forall k \in V$$

$$(1.8) \quad \sum_{k \in V} \sum_{j \in N} x_{0jk} \leq |V|, \quad \forall k \in V, \forall j \in N$$

$$(1.9) \quad x_{ijk} \in \{0,1\}, \quad \forall i, j \in N, k \in V$$

Constraint (1.2) ensures every customer is visited once. The truck capacity cannot be exceeded because of constraint (1.3). With constraint (1.4), the model makes sure that every vehicle will start at the depot and with constraint (1.5), each vehicle ends in the depot as well. Constraint (1.6) ensures that when a vehicle enters a location it also leaves the location. With constraint (1.7) a relation is created between the departure time of a vehicle at a customer and the next customer on the route, thus eliminating subtours. Constraint (1.8) is a constraint that incorporates a maximum number of routes compared to the number of vehicles. This can be used to make sure not too many vehicles are used for the whole milk-run. Lastly, constraint (1.9) is the integrality constraint for decision variable x_{ijk}.

3.4 Extensions on the VRP

As mentioned in Section 3.2, the VRP has a lot of characteristics and thus extensions for different scenarios that can be implemented to fit each scenario. For a VRP to represent the milk-run of Thales, certain extensions are required. The first one is adding time windows to the VRP in Section 3.4.1. The extension of using soft time windows is discussed in Section 3.4.2. After this, the Split Delivery VRP is explained in Section 3.4.3 and the VRP with Compatibility Constraints is discussed in Section 3.4.4. Lastly, a conclusion is provided in Section 3.5.

3.4.1 VRP with Time Windows

The first extension that needs to be added to the VRP is the time window extension (VRPTW). This element adds a time dimension to the problem, which is required for the situation of the milk-run in Thales. The time dimension is needed, because the milk-run drives multiple times a day and not every point of the day has the same demand and the same needs as other points of the day. Time windows serve as an interval in which certain demand should be picked up. This can vary from the whole day to a certain time limit, just like the situation in Thales. To formulate the time windows, some new notation and constraints are implemented by Kallehauge et al. (2005); Each customer i has a time window $[a_i, b_i]$. This means the demand from the customer is available from a_i and must be picked up before b_i. Constraints that are added because of the time windows are the following:

$$(1.10) \quad x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0, \quad \forall i, j \in N, \forall k \in V$$

$$(1.11) \quad a_i \leq s_{ik} \leq b_i, \quad \forall i \in N, \forall k \in V$$

Constraint (1.10) replaces constraint (1.7) as it now has time windows, which changes the relationship between the vehicle departure time from a customer and its immediate successor. Using constraint (1.7) leaves too much freedom for the variable s_{ik}, which leads to infeasible solutions when using this constraint. A disadvantage of this constraint is the non-linearity of it. This implies a longer computation time. The extension, however, is needed to create the situation Thales is in and the way it handles its demand. Constraint (1.11) makes sure that the time windows are observed.

3.4.2 VRP with Soft Time Windows

Using time windows in a general way does not cover the actual situation at Thales. This is because the hard time windows either allow or disallow products to be delivered/picked up within a certain time. At Thales, it can be possible that this time window is heavily desired, but not necessary. The usage of soft time windows can help here (VRPSTW). (Salani, Battarra, & Gambardella, 2016) state that the time windows usually are considered as a kind of demand restriction in the VRP. The time windows can be divided into two types: hard time windows and soft time windows. Hard time windows have been discussed in the previous section. For this type of time windows, the service of the demand has a definite time limit, and the customer has to be served in this limited interval. Soft time windows are a slightly different type of time windows, where the service of the demand has flexible time limits, so the customers can still be served outside of the limited time interval. Therefore, if the staff that delivers/picks up is early or late to serve the customers, it will lead to the penalties, but the demand can still be served (Chiang & Cheng, 2017).

In past research, the soft time windows can be divided into many types in a certain way called the penalties calculation method. In general, the penalties are calculated for the outside both early and late of the limited time interval (Chiang & Cheng, 2017). But in some models, the limited time intervals are considered for only one side of the exceeding time, that is referred to as semi soft time windows. For this kind of soft time windows, usually the upper bound is considered as being late is undesirable (Setak, Azizi, Karimi, & Jalili, 2017). Furthermore, there is a kind of soft time window that is in between the two types of time windows. It has a certain degree of allowance, and also has to calculate the penalties. And after a certain upper and lower bound, the penalties are set to infinity. This is shown in Figure 16, where P_i is the penalty of node i, $C_e$is the unit penalty cost for being too early and C_l is the unit penalty cost for being too late.

$$P_i \begin{cases} \infty & , if\ s_{ik} < \min a_i \\ C_e(a_i - s_{ik}) & , if\ \min a_i \leq s_{ik} < a_i \\ 0 & , if\ a_i \leq s_{ik} \leq b_i \\ C_l(s_{ik} - b_i) & , if\ b_i < s_{ik} \leq \max b_i \\ \infty & , if\ \max b_i < s_{ik} \end{cases}$$

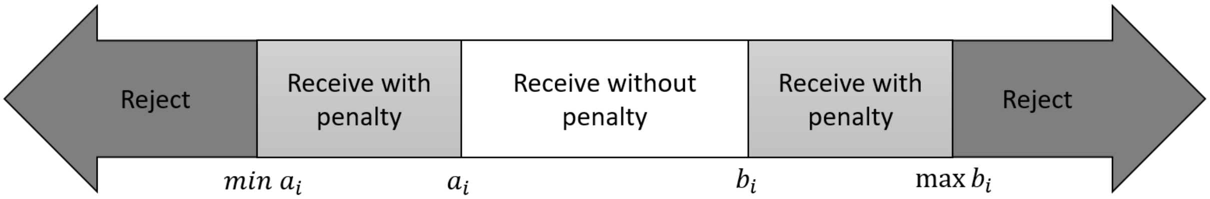


Figure 16 - The soft time windows (Chiang & Cheng, 2017)

This figure shows that there is one interval in which there is no penalty for the delivery/pick up of products. Besides this interval, there are intervals that are allowed, but will be penalized if the products are delivered/picked up in this interval. Lastly, the outer intervals are too far off from the initial time window that they are unacceptable and therefore are rejected. The VRP model will not allow products to be delivered in this interval. For Thales, the penalties should only be after the time window, as there is a possibility that products get delivered/picked up outside of the agreed time window. The products can not be delivered/picked up before their time window as this demand does not exist before this window. This means that the semi-soft time windows are enough for the situation at Thales. To incorporate the soft time windows, we can modify some of this notation. This modification simplifies the formula as we will only use the option of 'Receive with penalty'. Rejecting is not an option as every demand needs to be fulfilled. The necessary new notation is the following:

p unit penalty cost;
o_{ik} overtime of vehicle k at node i; $\forall i \in N, \forall k \in V$

The corresponding constraint to this is constraint (1.12). This constraint calculates the difference between the arrival time and the upper bound of the time window for each node. It then takes the maximum of this number and 0, so that it cannot be negative. Including the overtime variable and multiplying it with the unit penalty cost in the objective function, like $p * \sum_{i \in N} \sum_{k \in V} (o_{ik})$ is enough to complete this extension to the VRP model.

$$(1.12) \quad o_{ik} = \max\{s_{ik} - b_i, 0\}$$

3.4.3 Split delivery VRP

To simulate the situation Thales is in, another extension on the VRP is needed. The regular VRP model does not allow for demand to be split over multiple vehicles or routes. The demand per customer is either fully picked up/delivered by one vehicle or not. This means that if there is more demand at a certain customer than the capacity of the vehicle, the solution of the VRP will be infeasible. For Thales, some demand may be modelled over the whole day (24-hour jobs) and can be picked up/delivered by multiple vehicles in different routes. All demand, so also 1-hour job demand for the same location, can be split if necessary. For the milk-run, it can also be the case that it is not always necessary to visit each node in the network as often as the other nodes. Some nodes might have a very high demand for pickup and delivery of their products, while others might only need a stop of the milk-run once in a while. So, combining different demands from multiple time windows is an option. Split Delivery is a suitable extension to the VRP to tackle both of these requirements for the situation at Thales. In the Split Delivery Vehicle Routing Problem (SDVRP) a fleet of capacitated homogeneous vehicles is available to serve a set of customers. Each customer can be visited more than once, contrary to what is usually assumed in the classical Vehicle Routing Problem (VRP), and the demand of each customer may be greater than the vehicle capacity (Archetti & Speranza, 2008).

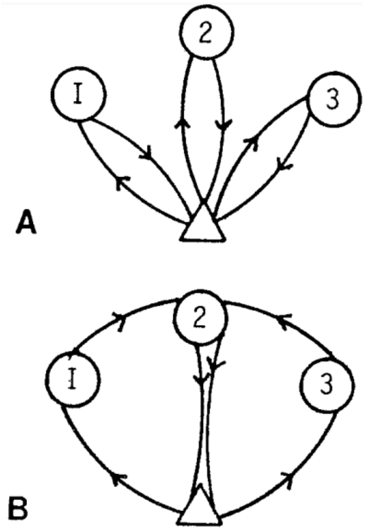


Figure 17 - (A) VRP Solution vs (B) SDVRP Solution (Dror & Trudeau, 1990)

Dror and Trudeau (1990) describe the SDVRP as a relaxation of the VRP, because the hard constraint of picking up all the demand in one time is disregarded. They demonstrate that utilising the SDVRP can lead to savings in total distance travelled, but also in total usage of number of vehicles (or thus routes). Figure 17 shows a solution of a VRP model in (A) and a solution of a SDVRP model in (B). For this instance, the capacity of the vehicles is too small to pick up demand at two nodes combined, so it makes three trips in the VRP. But the capacity of the vehicles does allow to split the demand of customer 2, which leads to a trip (and thus a vehicle) less, which causes a shorter travel distance.

To incorporate this extension in the VRP model, some new notation is needed together with some constraints that are added or replace existing regular VRP notation. A new decision variable is introduced to split demand:

q_{ik} The quantity of demand i picked up by vehicle k; $i \in N, k \in V$

The objective function remains the same for the SDVRP. The constraints however are altered. The constraints which belong to the SDVRP and are different from the VRP are the following:

$$(1.13) \quad \sum_{k \in V} \sum_{j \in N} x_{ijk} \geq 1, \quad \forall i \in N$$

$$(1.14) \quad \sum_{k \in V} q_{ik} = d_i, \quad \forall i \in N$$

$$(1.15) \quad q_{ik} \leq d_i * \sum_{j \in N} x_{ijk}, \quad \forall i \in N, k \in V$$

$$(1.16) \quad \sum_{i \in N} q_{ik} \leq Cap, \quad \forall k \in V$$

Constraint (1.13) makes sure that nodes now are allowed to be visited more than once, but still have to be visited at least once. In constraint (1.14) the total demand of node i is set to be equal to the sum of total of the split deliveries. Constraint (1.15) makes sure that q_{ik} can only take on values if the route that belongs to it is traversed by the corresponding vehicle. The new capacity constraint is constraint (1.16), which adds up all the picked-up demand by vehicle k and ensures this is lower than the vehicles capacity.

3.4.4 VRP with Compatibility Constraints

Some VRPs require a differentiation between vehicles, because one vehicle has other capabilities than the other concerning compatibility with locations (nodes). This VRP is known as the VRP with Compatibility Constraints (VRPCC) and is described in the paper of Yu, Nagarajan, and Shen (2017). In this extension of the VRP it is assumed that multiple types of services are demanded at various locations of a given network and each type of service can only be served by certain vehicles. In this formulation, each vehicle $k \in V$ can only visit a subset $V_k \subset N$, based on matches of vehicles and service types (The rest of the formulation is similar to the standard VRP formulation of Desaulniers, Desrosiers, Erdmann, Solomon, and Soumis (2002)). In this problem, a routing decision assigns each vehicle a route such that the nodes visited by vehicle $k \in V$ are in the set V_k. Yu et al. (2017) also add a binary parameter $u = (u_i^k, k \in K, i \in N)^T$ where u_i^k takes value 1 if $i \in N$ for vehicle $k \in V$, and 0 otherwise. The constraint that comes along with this is (1.17):

$$(1.17) \quad \sum_{i:(i,j) \in A} x_{ij}^k \leq u_j^k, \quad \forall j \in N, k \in V$$

3.5 Conclusion

In this chapter, the literature containing relevant topics concerning the milk-run is analysed by answering the question: *'What methods to create and optimise a model for the milk-run are available in the literature?'* and its sub-questions. This section discusses the answers to these sub-questions.

What is a milk-run?

A milk-run is described as a manually operated, transport system delivering and picking up materials and products, using a fixed route and time schedule. It concerns a transportation network where all materials for the input and output required by several stations are covered by one vehicle that visits all these stations and circulates according to a pre-defined schedule. The milk-run concept is commonly applied in internal plant logistics for all types of transportation between stations and the warehouses of the plant. A milk-run schedule can be considered as a special instance of the Vehicle Routing Problem (VRP).

What solutions exist in the literature to model similar problems as the milk-run?
The Milk-run problem can be modeled using either the Traveling Salesman Problem (TSP) or the Vehicle Routing Problem (VRP). TSP optimizes the route for efficient pickup, while VRP designs routes with constraints like time windows and vehicle capacity. Methods like total enumeration, branch and bound, and the Clarke and Wright algorithm solve TSP, while VRP can be addressed using optimization, heuristics, or metaheuristics like Simulated Annealing, Tabu Search, or Genetic Algorithms. Considering Thales' needs, a VRP approach with appropriate variants and possibly an algorithm seems most suitable for modeling their milk-run logistics efficiently. We use the three-level classification of the VRP models of Ni and Tang (2023) to classify the situation with the milk-run in Thales as a certain instance of the VRP.

How can a milk-run be optimised as a VRP?
A VRP is a process where the aim is to design a set of some minimum cost vehicle routes through some customer locations, so that each route starts and ends at a common location and some side constraints are satisfied. Many practical transport logistics and distribution problems can be formulated as a VRP whose objective is to obtain a minimum-cost route plan serving a set of customers with known demands and where the total demand of any route must not exceed the vehicle capacity. The VRP must be tailored in such a way to adequately represent a milk-run, especially the milk-run that is being investigated in this research. For this, extensions are necessary to create a good representation of the real-world scenario the milk-run is in.

*What extensions for the VRP are needed to represent the milk-run of Thales?*The initial extension for the VRP is adding time windows. This ensures that certain demand is picked up in certain periods of time. However, it may be the case that the demand is picked up later. Therefore, the usage of soft time windows is introduced. At Thales, it can be possible that a time window is heavily desired, but not necessary. The usage of soft time windows can help here. (Salani et al., 2016) state that the time windows usually are considered as a kind of demand restriction in the VRP. If the staff that delivers/picks up is early or late to serve the customers, it will lead to the penalties, but the demand can still be served (Chiang & Cheng, 2017). For a milk-run, it can also be the case that it is not always necessary to visit each node in the network as often as the other nodes. Some nodes might have a very high demand for pickup and delivery of their products, while others might only need a stop of the milk-run once in a while. It is therefore necessary to add another extension to the VRP to facilitate the creation of multiple routes that can complement each other and enable the option to visit certain nodes more often than others. In the Split Delivery Vehicle Routing Problem (SDVRP) a fleet of capacitated homogeneous vehicles is available to serve a set of customers. Each customer can be visited more than once, and the total demand of customers can be higher than the vehicle capacity. The last necessary extension to model a milk-run is for a VRP to be able to differentiate between vehicles, because one vehicle might have other capabilities than the other concerning compatibility with locations (nodes). This VRP is known as the VRP with Compatibility Constraints (VRPCC) and is described in the paper of Yu et al. (2017).

4. Solution design

In this chapter, the model is set up by answering the third main research question: *"How can we design a model which integrates and optimises the milk-run throughout Thales?"*. Section 4.1 explains the choice of the model that is used to generate solutions. Section 4.2 covers the techniques that are necessary to use and fulfil to cover the preparation concerning the input data of the milk-run. Clustering of the demand nodes, setting up the demand distribution and clustering the demand per time window are discussed techniques in this section. Section 4.3 focuses on the notation of the VRP model. In this section, the sets, parameters and variables will be introduced. There is also a brief description of the introduced notation that is not present in the earlier literature, but necessary for the model. The next section, 4.4, provides the objective and constraints of the VRP model. Finally, Section 4.5 explains the solution approach for the usage of the VRP model that is created.

4.1 Choice of model

The literature review showed that there are multiple ways of modelling a milk-run. The first choice to make is to use a TSP or a VRP as a basis to model the milk-run. We opt for a VRP as the milk-run poses some constraints that we have to adhere to. A TSP is not fit to incorporate these specific constraints as it can only solve single type of vehicle routing problems when modelled correctly. After this initial choice, there is still a lot of freedom for the direction of the model as the VRP has a lot of variants and can be assisted by algorithms, like optimization algorithms, heuristics and metaheuristics. The second half of Chapter 3 introduced variants/extensions on the VRP model that exactly fit the situation of the milk-run of Thales. We therefore opt to use this way of modelling and create a VRP model based on these extensions. The extensions are separately introduced, but will be combined in Sections 4.3 and 4.4 to build up a new VRP model. The last choice to make is whether to incorporate any of the optimization algorithms, heuristics and metaheuristics. A reason for this would be to aid with the solving of big instances for the model. However, as the regular problem instance of the milk-run of Thales does not seem very extensive, we do not find it necessary to incorporate such an addition to the model. We expect to solve the problem instances to optimality within a reasonable time without the additions, so this would seem unnecessary to implement.

4.2 Data preparation

So, we choose to use a VRP model to model the milk-run of Thales. By using this model, we can generate scenarios to test the milk-run and optimize it where necessary. The first step is to have a clear description of the exact problem to solve with the model. We know Thales is in a situation where the usage of the milk-run should be improved and optimized. We want to see what way of working brings forward the best results for the KPIs. So, productivity must be improved while also sticking to the desired on time delivery and lead time of the demand in Thales. A VRP model will support the modelling of the milk-run, so this VRP will be set up. Before this model is set up, it is important to resemble the situation Thales is in, so we need to adjust the input data in such a way that the VRP can match with this situation.

To be able to start using a model for the milk-run, we have to apply some techniques to pre-process the data that will be used as input for the milk-run at Thales. This means the real life has to be imitated as closely as possible. It is also important to keep the feasibility and the size of the model in mind while designing as this might have a big impact on factors like computation time of solving the model. To facilitate these elements, the data must be pre-processed in certain ways. In Section 4.2.1, an explanation is given on the aggregation of certain stations within production facilities. In Section 4.2.2, an elaboration on the demand distributions within the model is provided. And Section 4.2.3 will introduce the modelling of duplicated nodes to support different time windows.

4.2.1 Clustering demand nodes

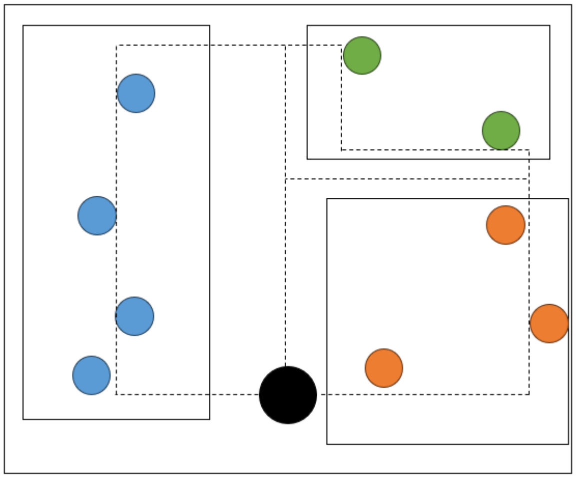


Figure 18 - Example of clustering demand nodes

Section 2.3 introduces the concept of workcells and stations that belong to it. As mentioned, stations are spread out over the production facilities within Thales. Every building has multiple stations inside of it. Certain buildings are larger than the others and each building is different. A similarity for each building is that there is only one possible route to take through the building itself. This is due to the fact that there is only one pathway for the vehicle that drives the milk-run, which does not allow for any freedom. Combine this with a large turning radius in relatively small empty areas, there is simply no possibility for other pathways than the current paths. This has a big effect on modelling the milk-run as this implies there are certain clusters of stations rather than separate stations that must be modelled. The clusters are based on routes that are taken inside a building and are created accordingly. This is implemented, because when the milk-run drives to a certain station, this means that other stations are visited anyway due to the routing and the positions of the stations on this route. A cluster of multiple stations will function as a node in the VRP. This means demand will be aggregated over the whole cluster instead of separate stations. To be able to cluster these demands and thus the stations, we identify which station belongs to which cluster. The average service times per station in a cluster are measured and added up to resemble the true situation as closely as possible. Figure 18 shows an example of the clustering of nodes. There are three buildings which the milk run supplies. The possible routes are shown with striped lines. These lines show there is no freedom within the buildings, so entering the left building to visit the lowest station inherently means, also visiting the next three stations. Therefore, these four stations can be clustered into one cluster to simplify the model and save computation time for the VRP.

4.2.2 Setting the demand distribution

The demand of the stations/clusters will be considered with a deterministic optimization approach. Fixed routes are computed using an approach based on average demands. These demands have a certain fixed distribution per day. We consider every day of the week to be the same, but not every time on the day has the same demand. For the demand input of the model, we will use the combined number of requests. This is because the 24-hour jobs are very constant and thus are representative for their part. The 1-hour jobs, however, can not always be picked up by the current milk-run, because of time constraints. But the expectation is that a large part of them will join the regular milk-run as a smaller lead time will be installed when the milk-run drives more often, since operators explain that such time intervals will be more convenient. Combine this with an expected increase in overall demand, we can assume the combined requests to be representative for the proportions of the demand. Figure 12 in Section 2.3 shows the division of this demand distribution. These proportions of the demand can be modelled by using different time windows with their own demand.

4.2.3 Cluster Demand per Time Window

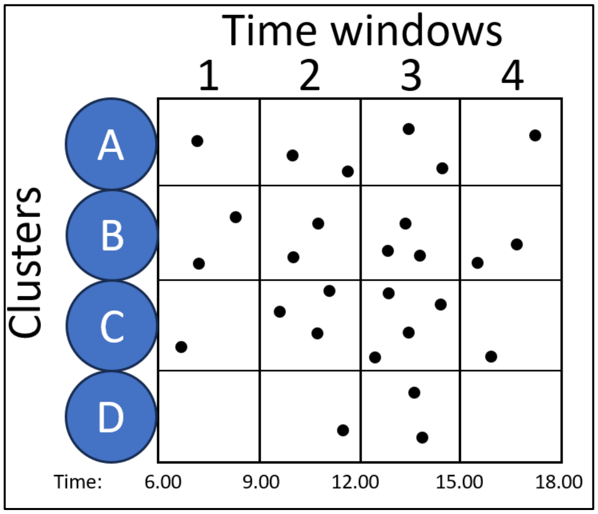


Figure 19 - Clustered demand per time window

To assign the different demands to certain time periods, time windows are necessary. Normally, when time windows are used, there is one time window per demand node. Now, multiple time windows are required per demand node. To achieve this, each demand node (cluster) will be duplicated within the model and have a time window assigned to it. This means that the VRP model will include multiple of the same nodes, which will all have different time windows and can therefore be considered as artificial nodes. These nodes will have no travel time between them. Figure 19 shows an example of how demand can be clustered over duplicates of nodes. Each dot is a demand instance. And in this case, each cluster has four time windows. When a certain instance of demand happens at 10.00 o 'clock at cluster C, this means it is taken up in a node of cluster C and time window 2, so square C2. This is done for every demand instance and each duplicate of each demand node (cluster) now has its own demand.

4.3 Sets, parameters and variables of the VRP model

After the pre-processing of the model has been completed, the actual model can be set up. To see what elements are in the model, the sets, parameters and variables of the model must be established. This chapter will introduce these elements of the model. The biggest part has been explained in Chapter 3, which is therefore based on literature. But a small part is based on practical additions. The two sub-chapters of this section will reflect on both.

4.3.1 Model based on literature

The literature has offered a good basis to create the model. In here, the sets define the entities involved, like customers and vehicles. The parameters provide the numerical values associated with these entities, like distances and the capacity. And decision variables are the variables that the VRP model solves for, determining the optimal routes and allocations of resources, linked to the objectives of the VRP.

Sets

$\mathcal{V}$	set of vehicles;	$\mathcal{V} = \{1, \dots, V\},\ k \in V;$
$\mathcal{N}$	set of demand nodes (customers);	$\mathcal{N} = \{1, \dots, N\}, i \in N;$
A	set of network arcs;	$A = \{(i,j): i,j \in N \cup 0\};$

Parameters

Cap	vehicle capacity;
d_i	demand for customer i; $\quad i \in N;$
t_{ij}	travel time on arc $(i,j) \in A$;
h_i	service time at node $i \in N$;
u_{ik}	$\begin{cases} 1, if\ i \in N \text{ for vehicle } k \in V \\ \quad 0, otherwise \end{cases};$
a_i	start of time window for customer i; $\quad i \in N;$

b_i	end of time window for customer i; $\quad i \in N$;
p	penalty for time over upper time window
M	large constant

Decision Variables

x_{ijk}	$\begin{cases} 1, if\ vehicle\ k\ drives\ directly\ from\ i\ to\ j \\ 0, otherwise \end{cases}; \quad i,j \in N \cup 0;$
y_k	$\begin{cases} 1, if\ vehicle\ k \in V\ is\ used \\ 0, otherwise \end{cases};$
s_{ik}	start of service time (arrival time) for node i; $\quad i \in N \cup 0$;
w_{jk}	waiting time for customer j served by vehicle k; $\quad j \in N; k \in V$;
q_{ik}	quantity of demand i picked up by vehicle k; $\quad \forall i \in N, \forall k \in V$
o_{ik}	overtime of vehicle k at node i, $\quad \forall i \in N, \forall k \in V$

4.3.2 Modelling additions

There are some additions and there is extra notation that are needed to make it suitable to be modelled in a practical manner instead of only theoretical and to make sure the model exactly fits the situation of a milk-run and not just a VRP model. These are not mentioned in Chapter 3 and therefore require some extra explanation. Some of these are already mentioned in the previous section, but will be explained in this section along with other relevant additions.

As mentioned in Section 4.2.3, nodes are duplicated to create a full image of the different customers. This way each customer consists of multiple nodes in the model. The total amount of nodes is calculated by multiplying the number of actual customers with the amount of time windows that are incorporated. Therefore, d_i shows the demand for customer i, where these customers resemble the duplicates of each station. So, for five customers and five duplicates, each first customer is a duplicate of the first cluster, e.g. customers 1,6,11,16 and 21. They all represent a different time window as illustrated in Figure 19. Each respective time window is represented by a_i and b_i.

For modelling purposes, the variable y_k is added to indicate if a vehicle is used or not, so multiple scenarios can be tested. This means that as much vehicles as desired can be added in the model, without them actually being used as this variable allocates a binary variable to this characteristic. This variable is incorporated in constraints (2.3), (2.4) and (2.13).

The other variable that is added is w_{jk}. This variable is used to calculate the waiting time between two nodes. Most of the time, this is simply the service duration at the node. But, it may be the case that a vehicle is too early for the time window and has to stop before it can start the drop-off and/or pick-up. This variable takes on the waiting time to facilitate this.

A parameter that is added is the service time h_i. The service time represent the time it takes to drop off and pick up products at a certain cluster. This parameter is used to discriminate between regular waiting times for a normal stop or a shorter waiting time if demand over different time windows is combined. This means that when a vehicle travels from one node to another it checks whether this node is a duplicate of the previous node, which means it is the same location. If this is the case, the stoppage time at the second (duplicated) node in the route is only a smaller fixed number of seconds instead of the larger amount for a stop, because the only extra time corresponding to it is taking off more products at once.

4.4 Objective and constraints

To formulate the model, we must define the objective of it and the constraints that belong to the problem. The objective is to find a set of routes that minimizes the total objective function, while keeping the other factors into consideration. Other factors that are kept into consideration are limited by constraints. The constraints in the model ensure that the solution that comes from the objective function adheres to practical and logistical constraints that are analyzed before.

Objective:

$$(2.1) \quad Z' = \min\left(\sum_{i\in N\cup 0}\sum_{j\in N\cup 0}\sum_{k\in V}((t_{ij}+w_{jk})*x_{ijk}) + p*\sum_{i\in N}\sum_{k\in V} o_{ik}*q_{ik}\right)$$

The aim of the goal function is to minimise the total costs that are spent within the whole transportation process. The left element shows the costs that are made by driving for a certain time and the waiting time that belongs to each customer. The right element adds up the penalties that are given for overtime of products delivered outside their time window. Therefore, the elements together should be minimised to optimise the model. Every element of the objective function is given in time, so no costs have to be considered.

Subject to

$$(2.2) \quad \sum_{k\in V}\sum_{j\in N\cup 0} x_{ijk} \geq 1, \quad \forall i \in N\cup 0 \; if \; d_i > 0$$

Constraint (2.2) makes sure that all nodes that have demand are visited at least once. It also allows the node to be visited more than once by setting the number of trips to a node equal or higher than 1.

$$(2.3) \quad \sum_{j\in N} x_{0jk} = y_k, \quad \forall k \in V$$

$$(2.4) \quad \sum_{i\in N} x_{i0k} = y_k, \quad \forall k \in V$$

Constraints (2.3) and (2.4) make sure that each vehicle leaves the depot once and enters the depot once respectively if the vehicle is used in the solution. This is done by equating the trips from or to the depot to the binary value of y_k which indicates the usage of a vehicle with a 1.

$$(2.5) \quad \sum_{i\in N\cup 0} x_{ihk} - \sum_{j\in N\cup 0} x_{hjk} = 0, \quad \forall h \in N, \forall k \in V$$

This constraint enforces flow conservation at each node, which ensures that when a vehicle enters a location it also leaves the location by setting the binary variable of leaving a node on a certain period equal to the value of entering the same node in the same period.

$$(2.6) \quad s_{ik} + t_{ij} + h_i - M(1 - x_{ijk}) \leq s_{jk,} \quad \forall i \in N\cup 0, j \in N, \forall k \in V$$

By this constraint, a relation is created between the departure time of a vehicle at a customer and the next customer on the route, thus eliminating subtours. It adds up the start of the service time of node i, the travel time from node i to j and the service time at node i and uses a big M to be equal or less than the start of the service time of node j. The big M is multiplied by 1 minus the binary value of travelling on the respective arc. This multiplication ensures for subtours to be eliminated.

(2.7) $s_{jk} * x_{ijk} = \left(s_{ik} + t_{ij} + w_i\right) * x_{ijk}, \quad \forall i, j \in N, \forall k \in V$

This constraint adds on constraint (2.6) by also setting the arrival time correctly. This way the waiting time can be computed correctly as the sum of the arrival time for node i and the travel time to j and the waiting time at i must be equal to the arrival time at node j.

(2.8) $a_i \leq s_{ik}, \quad \forall i \in N, \forall k \in V$

This constraint makes sure that the time window for the lower bound are observed by having the start of the service time within the given time window for every customer. This is a hard time window, so it must be honoured.

(2.9) $o_{ik} = \max\{s_{ik} - b_i, 0\}, \quad \forall i \in N, \forall k \in V$

Constraint (2.9) calculates the overtime per node per vehicle. It does this by calculating the difference between the arrival time and the upper bound of the time window for each node. It then takes the maximum of this number and 0, so that it cannot be negative in the case that the vehicle is on time. This way the soft time window is incorporated.

(2.10) $x_{ijk} \leq u_{jk}, \quad \forall i, j \in N \cup 0, k \in V$

By constraint (2.10), certain vehicles are allowed to only visit certain nodes and are blocked from other nodes. This is done by setting the value from node i travelling to j to be smaller or equal to the binary value of allowing the vehicle to visit a certain node when u_{ik} is 1.

(2.11) $\sum_{k \in V} q_{ik} = d_i, \quad \forall i \in N$

In constraint (2.11) the total demand of node i is set to be equal to the sum of total of the split deliveries q_{ik} divided over the different vehicles. This way all the demand must be fulfilled, but may be split over different vehicles and routes.

(2.12) $q_{ik} \leq M * \sum_{j \in N \cup 0} x_{ijk}, \quad \forall i \in N \cup 0, k \in V$

Constraint (2.12) makes sure that q_{ik} can only take on values above 0, if the route that belongs to it is traversed by the corresponding vehicle. This is done by multiplying the sum of x_{ijk} with a big M, which must be bigger or equal to q_{ik}. This means that q_{ik} only takes values if there is a possibility to drive to the corresponding node with the corresponding vehicle.

(2.13) $\sum_{i \in N \cup 0} \sum_{j \in N \cup 0} (q_{ik} * x_{ijk}) \leq Cap * y_k, \quad \forall k \in V$

Constraint (2.13) is the capacity constraint. This makes sure that each vehicle sticks to the imposed capacity limit. This is done by multiplying the capacity with y_k and setting this equal or greater than the sum of the picked-up portions of demand multiplied with their respective binary value for x_{ijk}.

(2.14) $x_{ijk} = 0, \quad i, j \in N \cup 0, k \in V \; if \; i = j$

Constraint (2.14) ensures that no arcs are created where the node of origin is also the node of destination, so i is unequal to j.

Integrality constraints:

(2.15) $x_{ijk} \in \{0,1\}, \quad \forall (i, j) \in A, k \in V$

(2.16) $y_k \in \{0,1\}, \quad \forall k \in V$

Having linked all these extensions of the VRP together, we come to a new variant of it. The extensions which are used are the VRP with Time Windows (VRPTW) and Soft Time Windows (VRPSTW), the Split Delivery VRP (SDVRP) and the VRP with Compatibility Constraints (VRPCC). Combining all these variants, we have formed the Split Delivery Vehicle Routing Problem with (partial) Soft Time Windows and Compatibility Constraints (SDVRPSTWCC).

4.5 Solution approach

The model that is set up will provide results based on the input that is provided. The first steps of preparing this input are given in this chapter. Also, the vehicle requirements are provided to support the option of combining inside and outside routes, which means there is more freedom in the solution space and thus approach.

4.5.1 Running the model

Now it is important to establish how the optimal solution can be determined for different scenarios. The full model will be entered into a program which can compute outcomes for large Vehicle Routing Problems. For this research Python is used where Gurobi assists as a solver for Vehicle Routing Problems and their extensions. We use Gurobi, because the problem is quite diverse and might need some computational force to tackle. This is due to the fact that there are a lot of variables in the problem, because of the possibility to split demand. This increases the difficulty of solving the problem. Gurobi is a leading optimization solver widely used for solving complex optimization problems efficiently. It is particularly useful in Vehicle Routing Problems, where it helps find optimal routes for vehicles to serve customers while minimizing costs or maximizing efficiency. This perfectly fits the scenario for the experiments needed to test.

Gurobi can solve problems of different sizes. The problem instance and the data that belongs to it will influence the duration of the solving of the problem. When an instance is not extremely extensive, the complete solution can be computed by the model. But there can also be significantly bigger instances. That is why a certain time limit must be established for larger instances where not every possible solution can be explored by the solver. If an instance is not solved to optimality, the gap of the unsolved instance should be noted. The gap refers to the difference between the best-known solution found by the solver and the best possible solution. It is expressed as a percentage of the best possible solution. A smaller gap indicates that the solver has found a solution closer to the optimal one, suggesting that it is performing well and likely converging towards the optimal solution. Reversely, a larger gap indicates that the solver has not yet found a solution close to the optimal one, suggesting that more computational resources or better algorithms may be needed to improve the solution quality. This will indicate whether the solver and model function correctly and how much they can take on depending on problem instances as a small gap indicates that the solver is performing effectively and efficiently in finding high-quality solutions to the given problem.

4.5.2 Linearization of the model

As mentioned above, there is a possibility that the problem instances become too extensive. One way to tackle this is to linearize the model where possible. This is not always effective, so this must be verified first, but can be very beneficial for the computation time of the model. That is why, before the experiments regarding outcomes of certain events and scenarios can be done, we must first establish whether it is beneficial to linearize the model. In the model, Constraints (2.7) and (2.13) can be linearized, just like the first element of the objective function. The second element of the objective function cannot be linearized as neither of the variables are binary. The quadratic elements of the model can be replaced by linear elements, only when binary elements are a part of it (Sabo, Kumar, Cohen, & Kingston, 2012). Therefore, the following general expression is shown to illustrate the linearization steps taken for the two constraints and the first part of the objective function:

1. The quadratic elements are in the form: $x_i * y_i$, where x_i is binary and y_i is continuous or integer.
2. An extra variable z_i is added to represent y_i when $x_i = 1$, and 0 if $x_i = 0$.
3. Three equations are added to symbolise step (2) by using a big M to linearize the formulation:
    a. $z_i \leq M * x_i$
    b. $z_i - y_i \leq M * (1 - x_i)$
    c. $y_i - z_i \leq M * (1 - x_i)$

These steps are carried out for the mentioned constraints and objective function and loaded into the model with corresponding variables to model the linearization.

4.6 Conclusion

In this chapter, the set up of the VRP model is explored by answering the third main research question: *"How can we design a model which integrates and optimises the milk-run throughout Thales?"*. Its sub-questions assist in elaborating on this topic. The main findings are discussed in this section.

What model do we use to model the milk-run of Thales?
We choose to utilize the modeling approach to construct a VRP model incorporating the necessary extensions. Given that the typical problem instance of the Thales milk-run does not appear overly complex, we regard it unnecessary to integrate additional optimization techniques or (meta)heuristics into the model. Our expectation is to achieve optimal solutions for the problem instances without these enhancements.

How should the data be prepared before the implementation of the VRP model of the milk-run?
There are three main techniques to get through before using the model. The first is the clustering of demand nodes. There are certain clusters of stations rather than separate stations that must be modelled. A cluster of multiple stations will function as a node in the VRP. This means demand will be aggregated over the whole cluster instead of separate stations.
The second technique is setting up the demand distribution. The demand of the stations/clusters will be addressed through a deterministic optimization method. Fixed routes will be determined using an approach that relies on average demands. These demands follow a consistent distribution pattern each day. While each day of the week is treated equally, the demand varies at different times throughout the day. These demand distributions can be used to create certain demand for time periods over the day. These fractions of the total demand can be modelled using different time windows with their own demand.
The third technique is clustering the demand per time window. To allocate various demands to specific time slots, time windows are used. Multiple time windows are necessary per demand node. To facilitate this, each demand node within the model will be duplicated and assigned a distinct time window.

What sets, parameters and variables are considered for the VRP model of the milk-run?
Chapter 3 largely explains the literature to show the main concepts that are present in the VRP. However, there are also practical insights incorporated into the model to make it feasible. The literature serves as a solid foundation for constructing the model. First, the sets are explained. These are entities such as customers and vehicles. Furthermore, parameters assign numerical values to these entities, such as distances and demands. Lastly, decision variables are those the VRP model resolves, optimizing routes and resource allocations to achieve the VRP objective whilst sticking to the constraints.

What objective function and constraints are considered for the VRP model of the milk-run?
The primary goal of the objective function is to identify a set of routes that minimizes the total time in the objective function while considering other important factors. These other factors are constrained by practical and logistical considerations, ensuring that the solution derived from the objective function sticks to the given constraints, for topics like capacity, routing and fulfilment of demand that came forward from the different extensions of the VRP. Linking all these extensions of the VRP together, a new variant is formed. Combining all these extensions, we have formed the Split Delivery Vehicle Routing Problem with (partial) Soft Time Windows and Compatibility Constraints (SDVRPSTWCC).

How can the optimal solution be determined?
The model that is designed, can be used to provide results based on input data, while first using the initial input pre-processing techniques. The full model will be entered into Python, which computes outcomes for large Vehicle Routing Problems by using Gurobi. Gurobi is an optimization solver for Vehicle Routing Problems and their extensions. For some instances the complete solution can be computed by the model. But there can also be instances which are too large for certain time periods. That is why a certain time limit must be established for larger instances where not every possible solution can be explored by the solver. A check whether linear or quadratic constraints and objective function performs better should be done to optimise computation time. A comparison for the number of solved instances and the gap (the difference between the best-known solution found by the solver and the best possible solution) will be made to assess the effectiveness of the model.

5. Performance

Chapter 5 consists of three main parts. It answers the research question: *"What outcomes does the model provide and how is its overall performance?"* Before this question can be answered, the pre-processing techniques that have been mentioned in Chapter 4 need to be applied. This is done in Section 5.1, where the clusters and the distribution of the demand are shaped and ordered in time window and an elaboration on the linearization is provided. Section 5.2 dives into the performance of the model, where it reflects on the current solution and compares this to the optimal solution in the future and the benchmark instance. Section 5.3 investigates the sensitivity of the outcomes of the solutions by altering the inputs for demand, capacity and penalties. Then, Section 5.4 provides a scalability experiment to see how the model performs when it is expanded. And lastly, Section 5.5 gives a conclusion of the chapter.

5.1 Experimental design

The VRP model is created to facilitate and solve the problem that Thales is facing with their milk-run. The model however should be able to solve all kinds of problems which are similar to such a milk-run or VRP problem instance. Therefore, it is important to test the model with experiments to see what results it produces. This way, not only the current situation is solved, but also other instances are put to the test to see how the model performs. The base case of testing the model is applying the current problem instance to the model and see what comes out of it. We can consequently compare this to several instances. Firstly, the current approach is modelled, so we can see how this compares to the optimal situation. We also want to investigate the future situation with one or two vehicles. The scenarios that will be tested and the analysed output are summarized in the Table 3:

Scenarios	**Output**
Current situation – Current way of working	Frequency
Optimal current situation – Optimal solution without new building	Vehicles used
Optimal future situation 1 vehicle – Includes new building STC	Objective function
Optimal future situation 2 vehicles – Includes new building STC	Routing

Table 3 - Scenarios and output for analysis

As shown in Table 3, there are four main scenarios to be analysed. The current situation entails the current way of working and shows the way the milk-run of Thales currently performs. The optimal current situation investigates the optimal way of working that is possible for the current situation Thales is in. This means no extra building is added yet and the current two vehicles are implemented. For the future situations we differ between the possibility of a new vehicle (which can combine indoor and outdoor driving) or not, but in both cases the new building along with the new spread of demand is in place. Next, we compare it to the benchmark routing for these scenarios to test the performance of the model. This benchmark routing is established with the help of the nearest neighbour algorithm to see how a simple solution based on this heuristic performs.

After this, the applicability and scalability of the model are put to the test to see how it performs computationally. Lastly, a sensitivity analysis is carried out for the original input. This means changing the parameters of the initial model. Parameters that are altered to test the sensitivity of the model, are the demand, capacity and penalties. To initialise the experiments, we must first apply the pre-processing techniques mentioned in Section 4.2.

The visualisation of the outcomes of the experiments is similar for the base cases. The only difference is the number of vehicles used. The sequence of visiting the clusters remains the same when using either one or two vehicles, because of the lack of freedom in the solution space. This means that when using two vehicles, the same routing comes forward as modelling with one vehicle. So, in routes for two vehicles, the LC is visited in between the inside and outside locations. The extra time for switching vehicles is added up in the travel time from indoor to outdoor locations. Since there are these two different scenarios of one or two vehicles, we must use a clear notation to be able to differentiate between them. This is placed behind the frequency of the number of trips in a day. Concluding, the solution is provided in the following manner:

Objective function: X with a frequency of Y with 1 / 2 vehicle(s) used.
Route 1 & 2: LC – Cluster 1 – Cluster 2 – etc.

5.1.1 Demand clusters

The first pre-processing technique to apply, is clustering the demand nodes. This is explained in Section 4.2.1. There are several customers, so multiple clusters have to be created. The following clusters are set up:

- **Cluster W0** – Stations 23, 28, 29 & 31 – This is the biggest cluster as the tugger train visits all the stations below in building W.
- **Cluster W1.1** – Stations 34 & 35 – This cluster contains two stations that must be visited by using the elevator. When this cluster is visited, cluster W0 is also visited as this route must be completed to arrive at one side of the floor W1.
- **Cluster W1.2** – Station 33 – This 'cluster' contains only one station as this station is visited by taking another elevator to reach the other side of the floor of W1.
- **Cluster GCC** – Stations 3 & 36 – This cluster contains the two stations at building GCC-PCB. The vehicle of the milk-run does not have to enter the building to pickup and deliver products here, because the stations are located behind a door next to the street.
- **Cluster Z** – Stations 32 & 40 – This cluster contains the current two stations at building Z. These stations might be spread over the building, but the clustering of this demand remains the same. There is one pathway in building Z through which the vehicle can drive.

There are two other relevant locations after the aggregation of the clustered stations. The first is the logistics centre (**LC**), where the route starts and ends. This can be considered as the depot of the VRP model. The next is the new building **STC**, where in the future, demand will be present as well. These are taken up in the model (when necessary) as well. Finally, station X1 is not considered in a cluster as the demand here is so low and irregular. Regular internal transport will take care of this station.

5.1.2 Demand distribution

There is no data available which represents the demand per station throughout the day. That is why the data that is presented in Section 2.3 will be the basis for the representation of the demand for the VRP. We combine these two datasets, so we create a certain typical demand distribution per day. This demand distribution can be used to create certain demand for time periods over a day. This is necessary as the graphs clearly show that the demand is not level throughout the day. For example, there is significantly more demand from 10.00 to 12.00 and after 15.00 the demand is not so high anymore. For the distribution, we choose to not consider the standard jobs with the SLA of three days. There is no clear data available concerning timestamps. They are therefore taken up in the total demand and spread over the rest of the data according to the known ratios. This is the only available data that is applicable to this situation and therefore we use this to establish the proportions of the demand distribution.

5.1.3 Clustered demand per time window

We base the number of duplicate nodes on the maximum possible trips the milk-run can make per day. The milk-run can drive four times a day when considering the driving preceded/followed by the picking process to get the vehicle ready for a new trip. Therefore, four intermediate time windows are set up for the 1-hour demands and one bigger time window is set up for the other demand, which includes the 24-hour jobs and standard jobs. So, there are five nodes per cluster with their own time window, because the milk-run can drive four times a day. The 1-hour jobs are considered to be a bit broader than the actual one-hour limit, so dividing them into time windows of two hours is a reasonable assumption to make. Certainly, because Thales indicates it might want to get rid of the 1-hour jobs and work towards a more constant stream of delivery without the faster service.

In order to create a typical day concerning demand, we want to assign the demand in such a way that resembles the daily demand for each time window for each cluster. We define the length of the time windows by separating the ten hours of demand over four blocks. The first time window is the early morning: 07:00-09:00. Here, the first batch of picking can be completed. The second time window is the regular morning: 09:00-11:00. This follows the first batch and has no particularities. The third time window includes the lunch break of one hour, which is therefore three hours instead of two: 11:00-14:00. And lastly, the fourth time window contains three hours as well, as the final hour has very little demand (±0.6% of the total): 14:00-17:00. The established time windows must then be linked to the demand in proportion to the total. A ratio is calculated based on the requested demand per time window and is shown in Table 4. Lastly, we have to establish the division between 1-hour jobs and 24-hour jobs. For the measured year, 1-hour jobs have 29.2% of the total requests and 24-hour jobs have 70.8% of the total requests. The measured demand (pickup and delivery) during the milk-run is divided accordingly. The numbers for the 24-hour jobs are directly placed into the table. The numbers for the 1-hour jobs are multiplied with the ratio of the time window.

Time Window	Ratio	W0	W1.1	W1.2	GCC	Z
07:00-09:00	0.19	2	1	1	0	0
09:00-11:00	0.28	2	2	1	1	1
11:00-14:00	0.36	3	2	1	1	1
14:00-17:00	0.17	1	1	1	0	0
07:00-17:00		20	15	7	6	4
Total		28	21	11	8	6

Table 4 – Daily cluster demand per time window

This creates the data input that is necessary for each time window within the model for the current situation. As mentioned, a new building will be added to the layout which will impact the demand distributions. This will be handled in Section 5.2.3. The jobs that desperately need immediate delivery are left out of scope, but the regular 1-hour jobs are computed for throughout the research. This creates a clear resembles of a typical day concerning demand.

5.1.4 Linearization of the model

As mentioned in Section 4.5.2, we linearized the model. The objective function can only partially be linearized, because the second element of the objective function cannot be linearized as neither of the variables are binary, so it is still partially quadratic. However, the partially linearized objective function will be referred to as linearized and the original objective function as quadratic. To test whether a linearized model performs better than a quadratic model, we discriminate between having a linearized objective and/or having linearized constraints. Table 5 shows the outcomes per problem instance, where each instance becomes bigger by adding one time window and thus adding extra customers. We run the first two instances for a maximum of ten minutes and the third instance for an hour to compare the computational results. These are the result of the objective function, the running time and the gap (if the running time of ten minutes is reached, which means the model stops). Section 4.5.1 explains the gap. The number of continuous and integer (and number of binary) variables are presented together with the number of quadratic objective terms and quadratic constraints. The results are shown in Table 5.

# of Time windows	Linear?		Objective	Duration (s)	Gap (%)	Variables			Quadratic objective terms	Quadratic constraints
	Constraints	Objective				Continuous	Integer	Binary		
2	Quadratic	Quadratic	1657	19	-	0	165	122	131	101
	Quadratic	Linear	1657	77	-	0	286	122	10	101
	Linear	Quadratic	1657	10	-	242	286	122	131	0
	Linear	Linear	1657	17	-	242	407	122	10	0
3	Quadratic	Quadratic	2860	600	36.8	0	640	514	542	452
	Quadratic	Linear	2845	600	38.9	0	1152	514	30	452
	Linear	Quadratic	2845	600	30.1	1024	1152	514	542	0
	Linear	Linear	2845	600	28.4	1024	1664	514	30	0
4	Quadratic	Quadratic	4508	3600	52.8	0	1575	1326	1383	1203
	Quadratic	Linear	4493	3600	53.7	0	2898	1326	60	1203
	Linear	Quadratic	4493	3600	58.9	2646	2898	1326	1383	0
	Linear	Linear	4457	3600	48.6	2646	4221	1326	60	0

Table 5 - Linear and Quadratic output

Table 5 shows clear results in terms of best performing combination of linear and quadratic elements. First of all, it becomes apparent that using linearized constraints always performs better using quadratic constraints. Next to that, the best performing combination for the smallest instance is using linear constraints and a quadratic objective function. For the two larger instances, which are more representative for the remaining experiments, linearizing both the objective function and the constraints perform better. Therefore, these remaining experiments will be carried out with the linearized model. Summing up the variables, objective terms and constraints will always give a lower result for the quadratic constraints, even though they perform worse. It is therefore apparent that the computation time is largely dependant on the number of quadratic objective terms and constraints as they are the most time consuming to compute. An unexpected finding is that only using the linearized objective function and not the linearized constraints will severely worsen the computational results for the smaller instance.

5.2 Base case – The current and future solution for Thales

The first experiments are focussed on analysing the base case. This is based on the problem of Thales and their usage of the milk-run, meaning that we perform four different experiments to test the current performance and to investigate where it can improve. We look into the current performance, the optimal solution now, the optimal solution in the future and the benchmark routing results. This means the demand for the 24-hour jobs and the 1-hour jobs is as provided in Table 4, the capacity is 50 per vehicle and the penalty for delivering a product in overtime is 0.2 per second (Sections 5.3.2 & 5.3.3 elaborate further on these numbers).

5.2.1 Current situation – Current solution

First, the current situation is modelled. In order to resemble the current situation and calculate the current performance of the milk-run, we have to force the model to represent the current way of working for the milk-run. A big difference is the usage of the 1-hour jobs compared to the optimal solution. Because in this solution, the usage of 1-hour jobs is diverted into blocks of around two hours, which is acceptable according to Thales. The current situation lets the vehicle only drive twice a day, this means the blocks that the demand is divided in are a lot bigger, namely more than four hours. This is not acceptable when considering the 1-hour jobs.

We can model the current situation by forcing the vehicle to drive the second and fourth shift as these are the regular times it currently drives. Forcing can be done with the help of the compatibility extension of the VRP where we allow the vehicles to only be compatible with certain time windows. This way the jobs of the first shift and the third shift all have to be picked up by the internal transport. Internal transport is not incorporated in the model, so these penalties have to be assigned manually to resemble the current situation. When we model the current situation with manually giving out penalties afterwards, the following objective function and routing is created (Appendix A.1):

Objective function excluding manual penalties: 3,374 with a frequency of 2 with 2 vehicles used.
Route 1 & 2: LC – W1.1 – W1.2 – W0 – Z – GCC – LC.

We get the same optimal loop for both the shifts, which is the optimal complete route for visiting all five clusters. Optimal complete route means the route that visits all clusters in the optimal order. Now, the internal transport drives during the first and third period to deliver the 1-hour jobs that are now unpunished. The manual calculation for the internal transport means they have to deliver each 1-hour job that is not included in the driven routes. This is done in a separate tour per product, just like the real scenario. This would come down to twelve products using up four minutes extra. So, 2,880 (12*4*60). Adding this to the initial solution gives a final result of 6,254 for the current objective function. It becomes apparent that another way of working should be established. This is done in the next section concerning the optimal solution.

Objective function including manual penalties: 6,254 with a frequency of 2 with 2 vehicles used.

5.2.2 Current situation – Optimal solution

Next, we calculate the optimal solution for the routing problem of Thales. Opposed to the previous section, for this instance, we do not force the model to represent the current way of working for the milk-run, but provide full freedom, so the model can choose the strategy and routing that is optimal. This solution is computed for the current situation, which means there are two vehicles (and the new building STC is not incorporated). The following objective function and routes were created (Appendix A.2.i):

Objective function: 5,615 with a frequency of 4 with 2 vehicles used
Route 1 & 4 are done with one vehicle. Route 2 & 3 swap vehicles after W0.
Route 1 & 4: LC – W1.1 – W0 – W1.2 – LC
Route 2 & 3: LC – W1.1 – W1.2 – W0 – Z – GCC – LC

The route of vehicle 2 and 3 is the complete route for traversing along all the clusters in one route. This route is shown in Figure 20. One remark to be made here is that clusters W0, W1.1 and W1.2 are interchangeable in the order of the route, as they are all connected to the same circle that is visited. This is directly seen in the two remaining routes; the routes for vehicle 1 and 4. They have the same clusters, but a different route according to the model. In practice, this means the vehicle will drive the same route, but only the order of stops is different. Since there is no pick-up, this order does not make a difference, so the routes are equal. This route is shown in Figure 21.

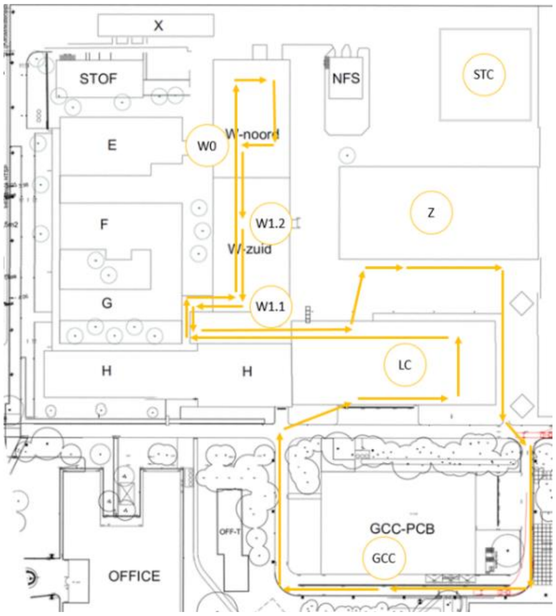


Figure 20 - Optimal route current situation [All clusters]

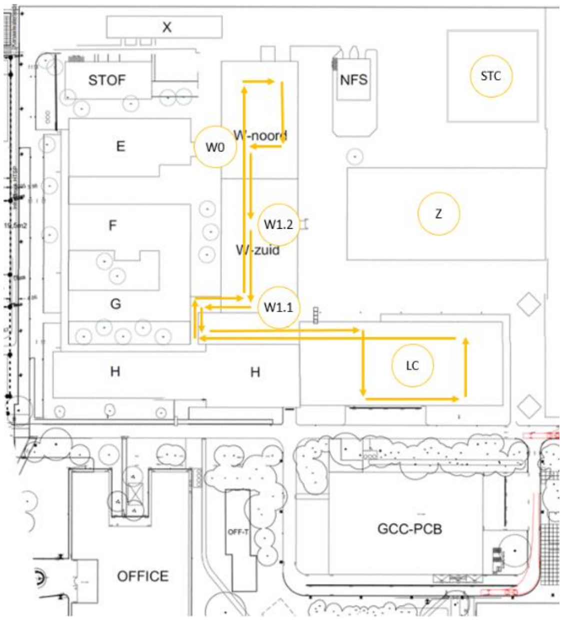


Figure 21 - Optimal route current situation [Inside clusters]

The solution with an objective function of 5,615 performs significantly better than the current situation, which has a score of 6,254. This means the objective function is decreased by 10.2%. This is due to the fact that vehicles now drive four separate tours and the demand is all fulfilled within their time window, which reduces penalties. The extra trips and thus travel time is justified by these newly provided advantages.

Because of computational time limits, the optimal solution is found in two steps. The first step is to run the model with hard time windows until optimality. The computational time significantly decreases because of the decrease in freedom of the solution. The second step is to run the initial problem until it reached the same optimal point, which indicates that this is the optimal solution for this instance as well. These two steps could be used as it became apparent that in smaller solution spaces the time windows were not violated. This allows the solution to be optimal so no gap is left.

The routing that is created is optimal, because it minimises the travel time between the nodes that are desired to be visited in each time window. The possibility of delivering products outside their time window is not utilised, because there is no incentive to take demand from other time windows as the penalty of delivering outside the time window is higher than the ‘reward’ in terms of not travelling somewhere twice. Delivering outside the time window costs at least one hour of waiting time, which is multiplied with a factor of 0.2. This comes down to a penalty of 720 seconds. Compare this to travelling somewhere extra, e.g. to GCC, which is 2*150 = 300 seconds extra to travel to and from the logistics centre. Delivering in overtime therefore does not weigh up to driving extra for the current parameters, especially the penalty.

However, it is interesting to investigate the effect of reducing this penalty in such a way that it might be beneficial to combine nodes more often. Section 5.3.3 details these effects.

5.2.3 Future situation

After having established the optimal solution for the current situation, it is also very important to establish the solution for the future situation. In the future, the milk-run of Thales is expected to cover the new building STC as well. Some new demand comes along with this location. The overview of the new demand is shown in Table 6. This is based on the expectancy of how Thales expects the demand will change when the new building is used. Important to note is that the total demand goes up instead of splitting over more locations. Next to the building and demand change, it is expected that a new vehicle will be introduced together with a change in the layout of hall W0. This would allow the vehicle to leave the hall there and drive a lot quicker to the outside buildings STC and Z as there is a quick outdoor route instead of driving slow indoors. We differentiate between these two options as this is not a certainty.

Time Window	W0	W1.1	W1.2	GCC	Z	STC
7:00-9:00	1	1	1	0	0	1
9:00-11:00	2	2	1	1	1	1
11:00-14:00	2	2	1	1	1	2
14:00-17:00	1	1	1	0	1	1
7:00-17:00	20	15	7	6	4	10
Total	26	21	11	8	7	15

Table 6 - Demand future situation

5.2.3.1 Future situation – 1 vehicle

The first future case to be studied is the most desirable scenario, which includes only one vehicle. This vehicle is able to drive inside and outside and can therefore be used for the complete route and thus the total delivery of the products of the milk-run. The following solution is created by the model (with the same steps as in Section 5.2.2, so the solution is optimal) (Appendix A.3.i):

Objective function: 5,803 with a frequency of 4 with 2 vehicles used

Route 1: LC – W1.1 – W0 – W1.2 – STC – Z – LC
Route 2 & 3: LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC
Route 4: LC – W1.1 – W0 – W1.2 – STC – GCC – LC

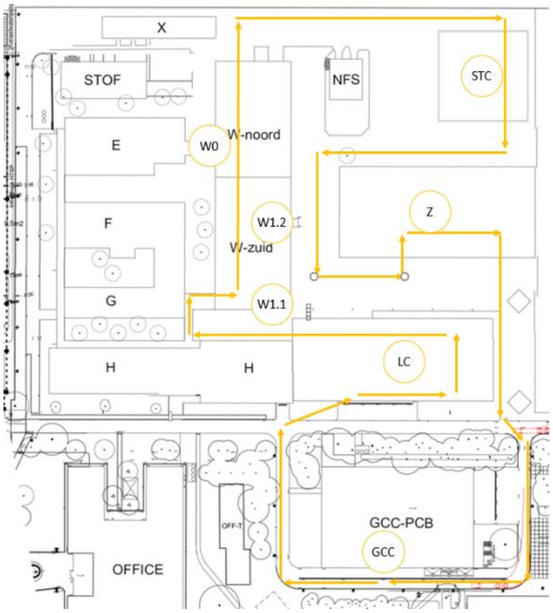


Figure 22 – Optimal route future situation – 1 vehicle [All clusters]

The route of vehicle 2 & 3 is shown in Figure 22. This route includes all the customers and thus travels along all the clusters. The routes for vehicle 1 and 4 that are taken, are very similar, but they omit GCC and Z from their route respectively as there is no demand in this time window. The routes are similar to the initial solution for the current situation, but there is one significant difference. Namely, the vehicle leaving W0 and traveling to the outside locations without returning to the LC first. This guarantees a drop in travel time from 5,615 to 5,264 (Appendix A.2.iii), which is 351 when STC would not be included, so an improvement for the current situation already. This also becomes clear in the objective function for the future situation, as this only increased by 188, while it now additionally visits STC four times with a stopping duration of 4*60 = 240, which is already more than the increase of 188.

When modelling the future situation with the current way of working the same as the second way of computing for the current situation (without penalties), we obtain an objective function of 3,950. However, the amount and length of 1-hour jobs here increases compared to the current situation. The number of 1-hour jobs rises to fourteen and because the new building STC is far away, the average delivery time also increases with 30 seconds. This comes down to an additional penalty of 3,780 (14*4.5*60), which gives a solution of 7,730. The improvement that is accomplished for the future situation, therefore is 24.9%. So, these four routes combined provide the optimal solution for the future when the STC building will be used. It is similar to the optimal solution for the current situation in a sense that no products are delivered in overtime, because of the heavy penalty. Furthermore, the intuitive routing is followed, and no particularities are present in this solution.

5.2.3.2 *Future situation – 2 vehicles*

The other future case to be studied is the scenario in which there is no new vehicle to combine driving indoor and outdoor. This means that two vehicles will be in usage for the milk-run. Just as in the other experiments, for finding the optimal solution with two vehicles, the same routing comes forward as for one vehicle (when there is no possibility to leave through W0). The following solution is provided by the model (the solution is optimal) (Appendix A.3.ii):

Objective function: 7,115 with a frequency of 4 with 2 vehicles used
Route 1: LC – W1.1 – W0 – W1.2 – STC – Z – LC
Route 2 & 3: LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC
Route 4: LC – W1.1 – W0 – W1.2 – STC – GCC – LC

These routes are very similar to the solution for having one vehicle. The only difference is that now the vehicle has to travel indoor from any cluster in W to STC, see Figure 23. This takes up significantly more time, which creates a larger route and therefore a larger objective function. The frequency of visits to nodes and the sequence of the routing do not change at all. When comparing both optimal solutions, it is clear that using a vehicle that can combine driving inside and outside is extremely beneficial for the objective function of the model: 5,803 vs 7,115. This is an advantage of 2,312, which is approximately 40 minutes per day.

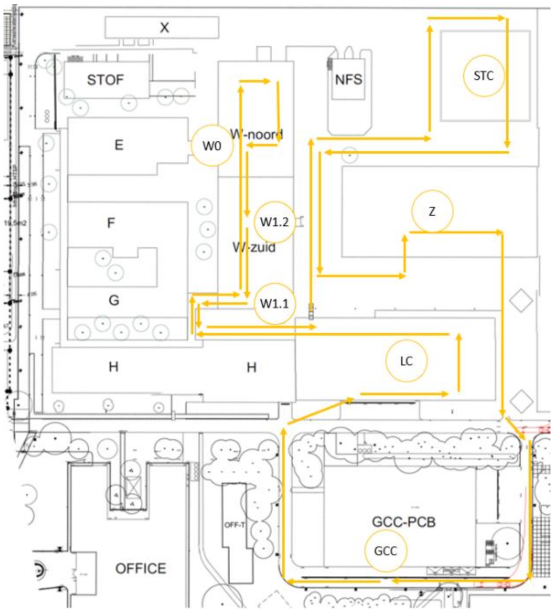


Figure 23 - Optimal route future situation - 2 vehicles

To summarize all the results of the base cases (current and future situation), Table 7 is setup based on the scenarios introduced in Table 3. This table provides the full overview of the outcomes of these situations.

Situation	Frequency * vehicles	Objective	Difference with current situation (%)	Routing	
				Shift	Routes
Current situation	2*2	6,254	-	1 & 2	LC – W1.1 – W1.2 – W0 – Z – GCC – LC
Optimal current situation	4*2	5,614	-10.2	1	LC – W1.1 – W0 – W1.2 – LC
				2 & 3	LC – W1.1 – W1.2 – W0 – Z – GCC – LC
				4	LC – W1.1 – W1.2 – W0 – LC
Optimal future situation 1 vehicle	4*1	5,803	-7.2	1	LC – W1.1 – W0 – W1.2 – STC – Z – LC
				2 & 3	LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC
				4	LC – W1.1 – W0 – W1.2 – STC – GCC – LC
Optimal future situation 2 vehicles	4*2	7,115	13.8	1	LC – W1.1 – W0 – W1.2 – STC – Z – LC
				2 & 3	LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC
				4	LC – W1.1 – W0 – W1.2 – STC – GCC – LC

Table 7 - Output base case analysis per situation

5.2.4 Benchmark routing

In order to validate the productivity of the model, we want to compare the optimal solution that is created by the model with a benchmark solution. A benchmark can be established by using logic reasoning and simple and intuitive rules to create a basic solution. There are a few assumptions one can make when creating such a routing. The first assumption is that the vehicles should travel as often as possible. The second assumption is that each routing is the same and should include all customers, so no products are missed. The third assumption is that creating this route from scratch can be done by using a simple variant of the nearest neighbour algorithm, because this method is the simplest and has the characteristics of forming distribution routes according to the real conditions in the field, presented by Harahap (2023).

The nearest neighbour algorithm solves optimization problems as in VRP. The steps are as follows:

1. Select a point that represents the starting location. Which is the logistics centre for Thales.
2. Then select the destination point to be visited next, with consideration only choosing the location that has the closest distance to the location that was previously visited. This means the inside locations will be visited first, followed by the outside locations that are closest to the last inside location (W0), which is Z for the current situation and STC in the future situation. And then completing the route by going to their respective nearest neighbour.
3. When all locations have been visited, close the trip route by returning to the point of origin, the logistics centre again.

These assumptions together create a scenario where the complete route when visiting all five clusters is driven four times. This routing does not consider ‘empty’ nodes/customers, which causes a rise in the objective function compared to optimal instances. The benchmark routing can be modelled by setting a mandatory visit to all nodes by all vehicles. The cluster sequence within the routing created by the nearest neighbour algorithm is the same as for the optimal routes created by the VRP. We do this for the current situation and for the two future situations. The results are shown in Table 8.

Scenario	Frequency * vehicles	Objective function	Increase vs. optimal	Routing
Current	4*1	6,523	908 (5,615)	LC – W1.1 – W1.2 – W0 – Z – GCC – LC
Future	4*1	6,138	335 (5,803)	LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC
Future	4*2	7,450	335 (7,115)	LC – W1.1 – W1.2 – W0 – STC – Z – GCC – LC

Table 8 - Benchmark routing

For all instances, all four routes are all constructed in the same sequence (with STC added in the future) (Appendix A.4). This means the route to visit all clusters is driven four times and the 24-hour job demand is spread over the routes. There is no differentiation regarding demand, which causes the vehicles in route 1 and 4 to also visit empty stations GCC and Z. The difference between the duration of the routes is that for some of the routes, multiple nodes are combined, so extra service time is implied. When looking at the comparison between the current situation, the optimal solution and the benchmark instance, it can be concluded that the benchmark already significantly improves the solution, but the current optimal solution still improves the benchmark instance by 14%, while the future optimal solutions improve the benchmark by around 5%. The future instances are closer to their respective benchmark instances. This is due to the fact that the demand increases in the future, so more visits have to be done, which is more in line with the benchmark. But there is clear improvement for all three cases.

5.3 Sensitivity Analysis

To evaluate the sensitivity of the solution and how it reacts to changes in parameters that might change in the future, we carry out multiple sensitivity analyses. These are done for the demand, where the number of 24-hour jobs and 1-hour jobs will be altered. Also, the capacity of the vehicles is decreased in order to see what influence this has on the outcome and what vehicle capacity is really necessary. Lastly, the penalty that is awarded for delivering outside the soft time window is adjusted. All the experiments concerning the sensitivity will be conducted for the instance of half a day (three time windows). This scenario is equal to a whole day in terms of decision-making, but uses up significantly less computation time. The basis of these experiments therefore takes half of the regular 24-hour demand over a day and the first two time windows of the day.

5.3.1 Demand

The experiments to see if demand influences the outcome of the model are split up in three parts. The first experiment only alters the number of 24-hour jobs. The second experiment alters the number of 1-hour jobs and the third experiment alters both of them.

5.3.1.1 Demand – 24-hour jobs

To evaluate the effect of more or less demand in terms of 24-hour jobs, we raise the numbers with factors 2 and 3 and lower the numbers with factors 0.67 and 0.5. These factors provide a broad range around the current situation, so a clear picture can be shaped of the impact of the change in parameters. Analysing scenarios that include even more demand is unrealistic and therefore unnecessary. Analysing scenarios that include less demand is useful to model quieter days. The results of these experiments all provided the same optimal solution with an objective value of 2,845. This came down to driving the complete route for both time windows and fully splitting the 24-hour job demand in such a way that the least nodes were visited an extra time. The demand had no effect on this as the number of visits necessary to deliver all demand did not change for each factor. Therefore, the route remained the same and the model showed that the current solution is not very sensitive for the influence of the 24-hour job demand. This solution will change once the demand surpasses the combined capacity of all the vehicles combined. In this case, an extra trip is necessary and thus extra travel time is needed.

5.3.1.2 Demand – 1-hour jobs

The factors for the 1-hour jobs have slightly been altered for GCC and Z, so that also demand is created when there is none in the regular scenario. This is based on the division provided in Section 5.1.3. It still comes closest to the factors of the previous experiment, but now with a possible added product for outside locations depending on the abovementioned division. The results are provided in Figure 24.

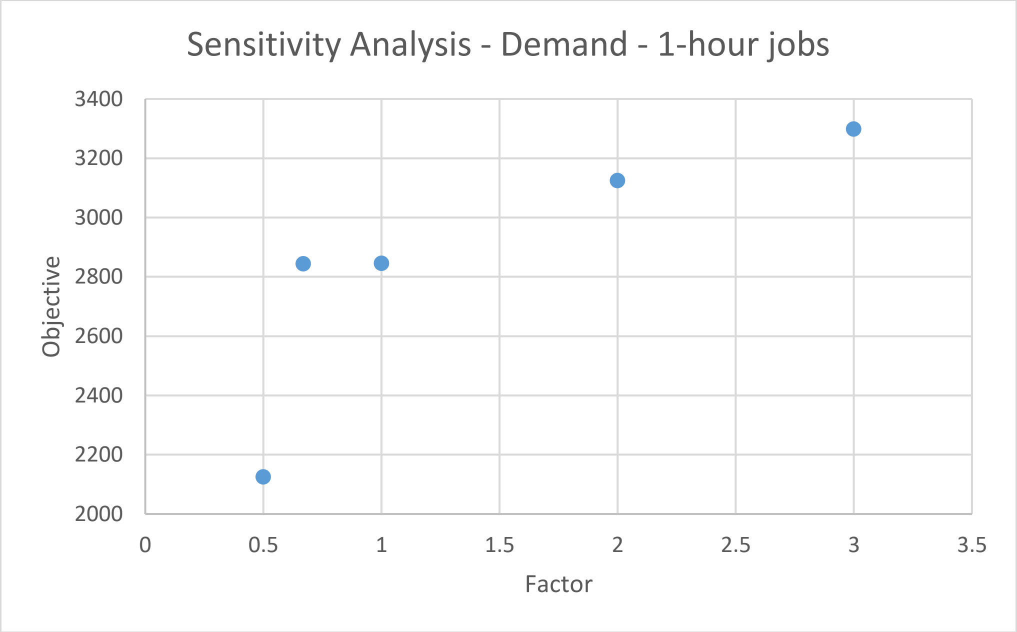


Figure 24 - Sensitivity Analysis - Demand - 1-hour jobs

As shown in Figure 24, the 1-hour job demand does influence the objective value of the model. This is due to the fact that 1-hour job demand is often close to zero. So, when the factor increases, more nodes have to be visited more often, which takes more (processing) time to handle. The reverse is also true; when the factor decreases, more nodes turn to no demand at all, so (processing) time is saved by not visiting these nodes anymore. For factors 2 and 3, the objective function increases by 280 (10%) and 454 (16%) respectively. A conclusion that once again can be drawn from this, is that the initial solution is not very sensitive and only slightly increases compared to the factor the demand is multiplied with. On the other hand, the objective function stays the same for a factor 0.67, but decreases significantly for a factor 0.5. This is due to the same number of nodes being present at 0.67 (13 nodes), but disappearing at 0.5 (8 nodes left). Having to visit less nodes significantly decreases the processing time at nodes and the necessary travel time between the remaining nodes. Because of this, the objective function then becomes 2,126.

5.3.1.3 Demand – All jobs

The last section of sensitivity analysis on the demand includes both 24-hour and 1-hour jobs. We will apply the same factor to both of them to simulate bigger growth or decline in demand. Using a factor 3 for both job demands will create an infeasible solution, when we stick to using only two vehicles, because of the capacity restriction. That is why an extra vehicle is used for this instance to illustrate the possible scenario this will cause. Figure 25 shows the results of the sensitivity analysis for both jobs combined.

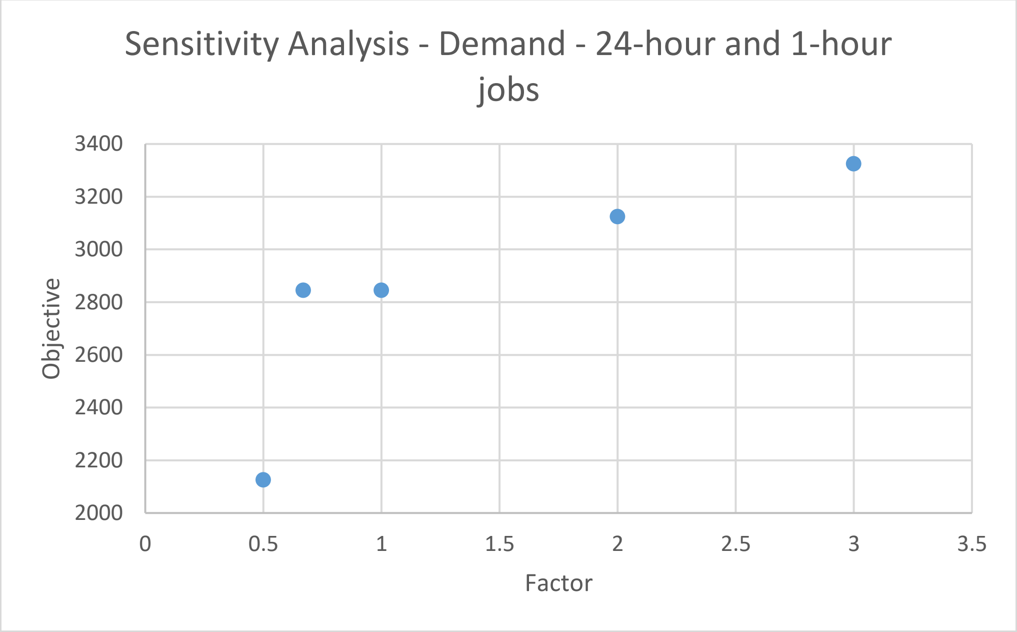


Figure 25 - Sensitivity Analysis - Demand – 24-hour and 1-hour jobs

When looking at Figure 25, we can see that for an increase in demand with a factor 2, the objective function only rises by 280, which is around 10% of the initial result. This is the same situation as for the rise in 1-hour jobs on its own. For the factor 3, an extra vehicle is added again and used by the model as the demand cannot fit in the vehicle anymore. This bumps up the costs as using an extra vehicle means using an extra route and thus driving 'unnecessary' extra lengths from and to the nodes and the logistics centre. This impacts the solution in the same way as the scenario for the 1-hour jobs alone, as now the objective function is increased by 763 to 3,608 (an increase of 27%). The decrease of the demand gives the same results as for only decreasing the demand of 1-hour jobs as also decreasing the 24-hour jobs does not change the amount of nodes any more and there is no further advantage for this scenario.

5.3.2 Capacity

The second parameter that is prone to a possible change is the capacity of the vehicle that drives the milk-run. When searching for a new vehicle, it is very relevant to know how the capacity of the vehicle influences the performance of the overall milk-run. The current capacity for the vehicle is 50 (boxes of) products. With an overall demand of 74 products per day, Thales has ample capacity for its current milk-run. Currently, the vehicle is quite large and quite bulky. If we can conclude that a smaller vehicle also satisfies the needs of Thales, this can be beneficial for the future. Therefore, the tested problem instances will all be lower capacities for the vehicle. The results are shown in Figure 26.

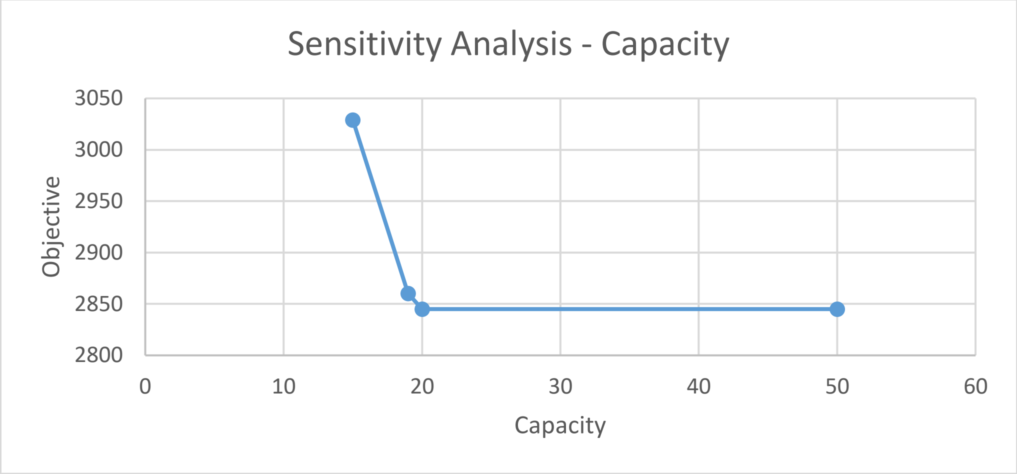


Figure 26 - Sensitivity Analysis - Capacity

Figure 26 shows the results of the sensitivity analysis of the capacity. The first and most important conclusion to be drawn is that the capacity does not influence the objective function until the products have to be spread out over the vehicles exactly 50/50. This supports the earlier conclusion of the solution being not sensitive at all as even with half of the capacity the initial solution still stands. For all the instances above a capacity of 19, the regular optimal solution is found as demand can evenly be spread over the two routes. When the capacity is 19, the 24-hour job demand for W1.1 is split over two vehicles and thus adds an extra 15 seconds to the objective. After this capacity, there is no possibility to fulfil the demand with two vehicles, so an extra vehicle is used. This causes the objective function to rise once more as the usage of an extra vehicle causes extra travel time.

5.3.3 Penalties

The last parameter that we alter is the penalty that is awarded to products that are delivered outside of the soft time window and thus have overtime. The penalty for bringing a product too late is based on how late the product is delivered, so how far after the time window deadline it was. This means that the penalty gets multiplied with the number of seconds it is too late. To establish the value of the penalty, we base it on the undesirability of the situation. Being 20 minutes late is regarded as equal to letting the product being delivered by the internal transport. A separate delivery of internal transport takes four minutes on average. Therefore, the penalty is set to 0.2 per second as this is the ratio when dividing the separate delivery by being 20 minutes late. This means that when a product is delivered 10 minutes too late, the penalty is expressed as 2 minutes in the objective function. When a product is delivered in the next time window, which can be 2 hours later, this means the penalty is 24 minutes.

So, the overtime of the visit to the node is multiplied with the number of products that are brought by the vehicle and then with the penalty. For the base case, no overtime is ‘used’. As mentioned in Section 5.2.2, the penalty of delivering outside the time window is higher than the ‘reward’ in terms of not travelling somewhere twice. In this section other penalties will be implemented to see when it is worth to combine the delivery of products for certain nodes to compensate for the penalties. The results are shown in Figure 27.

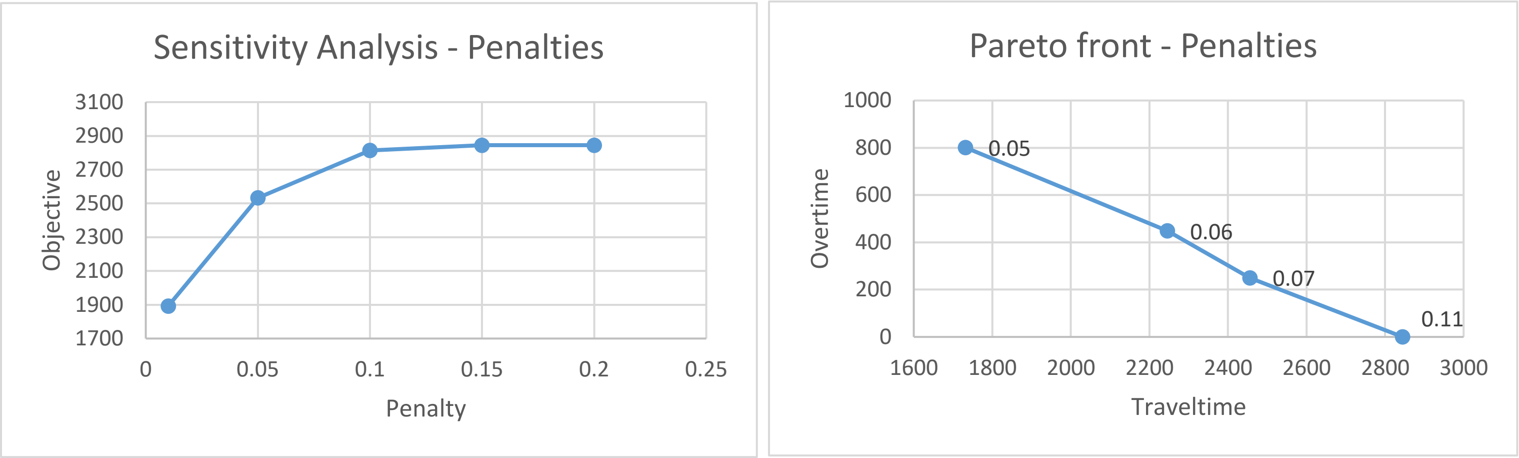


Figure 27 - Sensitivity Analysis & Pareto front – Penalties

As shown on the left of Figure 27, for values of 0.2 and 0.15 the standard optimal solution of 2,845 is found, where the routes visit all the necessary nodes and the 24-hour jobs are divided as a whole. But, when a penalty of 0.1 or lower is in place, the objective function gets lower. The reason for this drop differs per instance. For 0.1, there are still two routes that are driven, but the decision is made to exclude cluster W1.1 in the shorter route, so only W0 and W1.2 are visited and the overtime is accepted for W1.1. This leads to the objective function of 2813, which is a decrease of 32. When decreasing the penalty even more, we get more significantly different results. For a penalty of 0.05, which equals perceiving being late 80 minutes and an internal transport job of four minutes, the objective function reduces to 2,532.85. This is because now only one route is driven and the overtime is accepted for all products that are outside of this. The more we decrease the penalty from here, the lower the objective function will get as the route will remain the same and thus the overtime as well, it will only be penalised less. 0.1 can be regarded as the trade-off factor for accepting overtime or using the internal transport. This is verified by a separate experiment with a penalty of 0.11, which still produced the regular optimal solution as for 0.2. The factor of 0.1 resembles a ratio of one internal transport movement being equal to 40 minutes of overtime (time outside the time window) for a product. Between 0.11 and 0.05 there are some Pareto optimal points to be found, where the objective function still changes with 312.25 in such a short interval. From 0.05 downwards, only the overtime element will decrease, because of the penalty that is multiplied and keeps decreasing. The Pareto frontier contains a set of solutions that represents the best trade-off between the two elements of the objective function. A solution not dominated by any other solution in the feasible solution space is considered to be on the Pareto frontier (Datta, 2024). Because of the short interval in which both objective elements change, the Pareto front is not very extensive and not convex or concave, but rather linear.

5.4 Scalability

The last section of experiments is done to test the scalability of the model. The scalability of the model is relevant for experimenting in future scenarios, where even more nodes or time clusters might be added. To gain an estimate of what computational time is necessary for the model to run, we look at the development of total computational results compared to the total nodes that have to be evaluated as input. The total nodes of input can be calculated by multiplying the number of time windows with the number of customers. This is because each customer has the same number of time windows, which together make up all the nodes to be evaluated for the solution. Other parameters like the number of vehicles, capacity (compared to demand) and penalties might also influence the freedom of the solution and therefore the computational time, but are left out of scope for this experiment. In order to gain multiple results, we set the maximum running time of a single experiment to two hours (7,200 seconds). If in this time the gap has become zero, the computational time is noted. Otherwise, when the time limit has been reached, the gap is indicated. The results of these experiments are shown in Table 9.

# of Time windows	# of customers	Total nodes	Objective	Duration (s)	Gap (%)	Variables			Quadratic objective terms
						Continuous	Integer	Binary	
1	5	5	1,612	1	-	72	132	37	5
2	5	10	1,657	17	-	242	407	122	10
3	5	15	2,845	5,352	-	1,024	1,664	514	30
4	5	20	4,457	7,200	48.9	2,646	4,221	1,326	60
5	5	25	5,615	7,200	54.8	5,408	8,528	2,708	100
5	6	30	5,803	7,200	61.9	7,688	12,028	3,848	120

Table 9 - Scalability Analysis [All freedom]

As shown in Table 9, the computational time (duration) increases extremely quickly. Up until ten nodes, the model can finish almost instantly, but the time increases enormously afterwards. The solution for fifteen nodes is still found within the running time of two hours. After this, a big gap forms and increases with the addition of extra nodes to be considered in the model. As found in Section 5.1.4, the quadratic objective terms and constraints have the biggest influence on the computation time. There are no quadratic constraints as these are linearized, but the quadratic objective terms increase quicker. Together with the steady rise in continuous and integer variables, this causes a quick spike in computation time. It can be concluded that this VRP model is not suitable to be scaled up a lot more, because of the high computation time.

As the optimal solutions did not include the choice of vehicles combining demand from non-overlapping time windows, we can give the model some less freedom and take this as input. This means each vehicle is assigned to a certain time window for the 1-hour jobs and can all deliver the 24-hour jobs. Here the soft time windows are replaced by hard time windows as all demand is supplied within each respective time window. This is a variant on a regular VRP. The only difference now is that the 24-hour demand has to be spread and four routes are created. This variant provides the same answers, but muck quicker. These results are shown in Table 10. The variables and quadratic objective terms are not influenced and therefore are the same as for the situation with full freedom. This simplification is not always applicable, as for some situations, it may be beneficial to work with overtime and soft time windows. For the situation of Thales, this is not the case.

# of Time windows	# of customers	Total nodes	Objective	Duration (s)	Gap (%)
1	5	5	1,612	1	-
2	5	10	1,657	6	-
3	5	15	2,845	27	-
4	5	20	4,457	401	-
5	5	25	5,615	540	-
5	6	30	5,803	798	-

Table 10 - Scalability Analysis [Restricted freedom]

Table 10 shows that when the freedom is decreased, the same solutions are still provided, but a lot faster. Now all instances can be completed within fifteen minutes and the computational time seems to increase in a linear manner instead of exponentially from 20 nodes until 30 nodes. This means that in the future small scenarios can point out whether demand over non-overlapping time windows is combined. When this is not the case and the larger instances do not differ in characteristics concerning the connections of time windows and the demand in the assumption can be made that this also not happens for more nodes. This is not a given, but it is relatively certain. This can save a lot of computational time.

Relevant to mention is that all found optimal solutions (also for the earlier experiments) are found significantly earlier than the total running time. Their gap goes to 25% within ten minutes, but the remaining gap decreases very slowly. In these ten minutes, all these optimal solutions are already found, with most being even found within a minute already. Therefore, the optimal solution is always already found quite quickly, but the last 25% of the gap takes significantly longer to compute. In these cases, the solution did not improve any further within this gap, but this cannot be taken for granted for each scenario.

5.5 Conclusion

In this chapter, the performance of the VRP model is tested and an answer is provided for the fourth main research question: *"What outcomes does the model provide and how is its overall performance?"*. Its sub-questions contribute to elaborate on this topic. The major findings are provided in this section.

How can the pre-processing techniques be applied to provide a stable basis for the experiments?
There are four steps that have to be taken to prepare the experiments. The first pre-processing technique to apply is clustering the demand nodes. Consequently, the demand distribution is determined and is then placed in fitting time windows. Lastly, the model should be linearized to decrease the computational time as much as possible.

How does the solution improve with the help of the model currently and in the future situation?
The solution for the current situation improved from 6,254 to 5,615, which is a decrease of 10.2%. Which means that 10.2% less time is needed to complete the daily internal transport. This is due to vehicles driving four separate tours and the demand is all fulfilled within their time window, which reduces the initial penalties. Not having to drive 1-hour jobs is the most important cost-solving. Comparing this to the benchmark routing of 6,523, there is an improvement of 14%. Looking at future scenarios, which is divided in using one or two vehicles, the objective function improves even more. The rise of 1-hour jobs and the new building cause a bigger need for a more regular milk-run. Providing this regularity as an optimal solution decreases the solution with 24.9% (from 7,730 to 5,803). The routing of the overall optimal solution that is created, is always a combination of driving the complete route for visiting all customers and one or two routes that are shorter, because of the lack of demand for certain nodes.

How sensitive are the solutions?
To test the sensitivity of the solution, a sensitivity analysis is carried out for multiple parameters. First, the demand is altered. Changing the 24-hour job demand does not change the optimal solution at all. Creating extra 1-hour job demand does add to the objective function, but only 10% and 16% for multiplying the demand with a factor of 2 and 3 respectively. Creating both extra 24-hour job and 1-hour job demand gives the same solution for a factor 2, but increases the objective function by 27% for a factor 3, because of the need for an extra vehicle. These outcomes support the non-sensitiveness of the solution. Secondly, the capacity of the vehicle is modified. It became clear that the solution is created in such a manner that only when the capacity drops below half of the total demand (of half a day), the objective function

increases. This means that for the current situation a capacity of 19 is sufficient. For future instances, this needs to be slightly higher as demand is higher as well. But the current capacity of 50 boxes is and will be sufficient for a long time. And a smaller vehicle can be considered for the future. Thirdly, the penalty that is awarded for delivering a product in overtime is changed. When this ratio of one internal transport movement being equal to a certain number of minutes of overtime for a product drops below 0.11, the model might decide for delivering products in overtime. This means there is a crucial point at equalling one internal transport movement to 40 minutes of overtime, which is half of the current penalty.

How does the model perform in terms of scalability?
The scalability of the model is limited as the computational time seems to expand exponentially when increasing the number of nodes it has to take into account. It will therefore be very time-consuming to complete full analyses of extremely large scenarios. A quick fix for this is to check on a smaller instance whether the solution combines demand from non-overlapping time windows. When this is not the case and the larger instances do not differ in characteristics concerning the connections of time windows and the demand in it, one can assume that this is the case for the whole solution. This freedom can then be taken away from the model, which guarantees a much quicker computational time.

6. Conclusions and recommendations

This final chapter presents the conclusions and recommendations of this research. In Section 6.1 the main findings are recapitulated. This is followed by the recommendations belonging to these findings, which are presented in Section 6.2. Subsequently, the limitations and the possibilities for future research are discussed in Section 6.3. Finally, Section 6.4 elaborates on the contribution of the research in terms of both practice and literature.

6.1 Conclusion

To summarize the crucial elements of this research, we conclude the main findings and answer the main research question. The goal of this research is to gain insight in the milk-run process of Thales and to improve where possible. For this to be possible, a plan needs to be set up, where the integration of the milk-run is investigated and optimised. To be able to adhere to this, we need to provide an answer the main research question:

How should the plan, which integrates and optimises the milk-run throughout Thales, be built up?

The proposed solution to improve the milk-run mainly follows from an improvement in the routing and the frequency of driving of the milk-run. Currently, the milk-run drives twice a day with a main separation of two vehicles. It will be very beneficial for Thales to move to a scenario where an increase of the frequency of the milk-run to four times a day is implemented along with a new vehicle. This vehicle should allow the milk-run to drive one tour instead of two loose tours. There are some challenges that arise when implementing this plan, like the poor ground surface outside and the little freedom for driving the (milk-run) vehicles inside the production facilities. Nevertheless, there are some good opportunities to tackle these challenges and implement an improved plan for the milk-run for Thales. To provide this new way of working, a VRP model is set up to model the milk-run of Thales. This VRP model is adjusted and extended in such a manner that it resembles real life as close as possible.

A lot of effort currently is put into the transport of 1-hour jobs, where separate products are delivered by Thales employees. When deploying the milk-run more often, a significant number of these 1-hour jobs can be taken up in the daily routine of driving instead of ad-hoc movements. This will substantially decrease the time spent on internal transport at Thales. With the usage of the VRP model, we can compute the improvements in time for the milk-run. For the current situation, a decrease in time of 10.9% can be achieved. For the future situation, this is even more, namely 24.9%. This is due to the fact that a new building far away from the logistics centre opens, combined with a general rise in demand. The optimal solution profits even more, because of a current lack of support for this future situation.

The general description of the solution design created by the VRP model is straightforward. The model allocates demand within their given time window, since the penalty for being outside the time window is not worth it compared to possible advantages like driving less often. Subsequently, this means that the vehicle should drive as much as possible, which is four times. If a certain time window has demand on every node, all these nodes will be visited and the optimal route through all the clusters is taken. If a node has no demand in the time window, this node will be omitted from this route. The 24-hour job demand is spread out over these four routes. Combining these elements, the optimal routing schedule is created for each scenario. Not visiting all nodes every time is something that the benchmark solution does not do, which is therefore always performing worse than the optimal solution.

So, the plan to integrate and optimise the milk-run mainly consists of the new and more frequent routing. The other element for this plan is to replace the current vehicles by a vehicle that can drive both indoor and outdoor. This allows the vehicle to exit the indoor building at optimal points for the route instead of returning to the logistics centre and wasting time. This is incorporated in the experiments.

The main objective of the model was to improve the established KPIs. According to this plan, when a new vehicle is in place, savings of 10.9% and 24.9% can be reached for the current and future scenario respectively. This means that the most important KPI, the productivity rises as this can be regarded as the inverse of the costs. The main goal of the research is to maximize this productivity while sticking to the other established KPIs: the on-time delivery level and lead time targets. The second part of this goal is therefore also retained as the on-time delivery level remains stable and the lead time itself decreases for the 24-hour job demand and is stable for the 1-hour job demand.

6.2 Recommendations

After gathering the conclusions, it became clear that there are some improvements to be made for Thales. There are multiple scenarios to consider, each with their own recommendations. Furthermore, some general recommendations are provided concerning the milk-run and its usage in this section.

The experiments with the model have proven that driving the milk-run four times a day causes improved performances. This is the case for the current situation, and even more so for the future situation. There is no obstacle to allow the milk-run to drive more often as this will only save time. The trade-off that can be made here now, is allowing Thales to drive the milk-run four times a day as this saves time compared to driving two times a day. Even though extra resources have to be put in driving the milk-run more often, other resources will be saved, because the 1-hour jobs will be incorporated in the milk-run, causing a drop in total resources used. This can be implemented as soon as possible as this benefits the current situation already.

Combining the increase in frequency of driving the milk-run with an optimal routing brings even more advantages. The current route that is driven is already optimal for using two vehicles. But this routing will become even better when a new vehicle can be found that can combine indoor and outdoor clusters within one routing. Therefore, it is crucial to continue the process of finding a suitable vehicle that matches the desires of Thales for its milk-run. If it seems infeasible to find such a vehicle, other aspects might be interesting to investigate, e.g. upgrading the outdoor ground surface to facilitate the usage of vehicles from a different segment. Allowing a vehicle to drive one bigger route will cause the biggest decrease in time needed for the milk-run. This is even more relevant in the future as the new building is the farthest away from the logistics centre. To conclude, the routing proposed for the current and future situation should be implemented when possible.

A remark to make on the proposed routing is that the milk-run does not visit all the clusters every route. This is based on demand that has been sampled for a week. This week does not necessarily provide a perfect representation of the actual situation at Thales. An example where this becomes clear is that the cluster GCC carries the desire to be visited three times a day, because it wants to create products in a certain rhythm, where the milk-run picks it up as soon as possible. So, it is recommended to keep certain wishes of nodes or clusters in mind when designing the actual routing.

Next to the straightforward example of time savings, there are also other advantages of driving one route, more often, in the same schedule every day. This schedule with more moments that employees notice the milk-run will increase awareness of it and its reliability. Currently, employees occasionally request 1-hour jobs, because the waiting time until the next milk-run is too long. Allowing the milk-run to drive more often will lower this waiting time. A next step in this development is to get rid of the possibility to request 1-hour jobs, replacing them with ‘next milk-run jobs’. There might be some occasions where employees really cannot wait any longer, because they cannot continue working. In this case they can decide to pick up their own product if possible or discuss with the internal transporters to bring it. Removing 1-hour jobs significantly decreases loose product movements, which relatively costs the most time and separate movements of employees. Getting rid of separate movements causes a rise in soft advantages like professionalism, health and safety as less vehicles are driving on the shop floor and the work for the employees is further standardised.

Part of implementing this new plan in the future comes with a new division of the costs related to the milk-run. The advantage in time won by the new way of working can be expressed in less required employees/more productivity. It is difficult to attribute a monetary value to this, but an estimate would be that in the future the time saved is 32 minutes per day, which comes down to 2 hours and 40 minutes per week, which is a welcome benefit for a problem that needs readjusting anyway. For this improvement, a new vehicle must be bought or rented. This also means that the previous two vehicles can be disposed of. The costs of a new vehicle are still unclear, but are estimated to be equal to the current tugger-train. This way a saving can be made on the other vehicle by not using this one anymore.

Besides the practical recommendations on the frequency and routing of the milk-run, there are some additional recommendations to be made. These concern add-ons for the milk-run. The first add-on is to include buildings/nodes that are currently not in the regular loop of the milk-run by implementing a button system. For example, buildings X & STOF sometimes need a pick-up. When a button is placed at the pick-up and drop-off location, the internal transport employees know to include this building in their route if the button is turned on. This is mostly useful for these buildings which have a very low and irregular demand, but can also be used for future instances if demand shifts over buildings again and similar instances are created at other locations. Another add-on is to create some visual management to raise awareness of the milk-run. When people get attended on the milk-run more often, their awareness will rise and the trust in the milk-run will rise along. This might lead to less 1-hour jobs being requested, since employees trust the next milk-run to be there quick and on time again soon. This can be done by providing a visual element on drop-off and pick-up locations, like posters or stickers that contain some general information of the milk-run, like the frequency and visiting times of it.

Going further in the future, it might be interesting to look at a way to digitalise the milk-run information. This can include a device that is present on the vehicle of the milk-run which mentions what stations to visit and what the best routing is for those stations combined. Some information needs to be provided initially on what products are present on the vehicle and what products need picking up. This pick-up can be done with the button system and the drop-off can be linked to the picking process before the milk-run starts. This will guarantee an optimal route for each tour and it will improve the traceability problem Thales faces with the distribution of the products.

6.3 Limitations and future research

This section reflects on the limitations of this research and the gaps that are left for future research on this topic. The largest limitation of this research is the quantity of data that could be used for analysis of the current situation and input for the model to work with. Only, a sample of one week was taken to provide a general overview of the division of the demand over all the stations. This week is not perfectly representative for all separate weeks and days. Some remarks were made validating the representativeness by experts, but the stochasticity is missing. When more data is available on this required input about the demand for pick-up and delivery, a stochastic demand can be analysed. Chapter 5.3.1 reflects on changes in demand, but a more accurate representation of the real-life situation containing stochasticity would be more fitting. This means that no stochasticity is taken up in the input. So, it is not taken into account that there are possibilities for demand to be present at nodes that currently have no demand. A trade-off can be made to always visit certain nodes even though demand is not always expected.

An addition for the model would be to include a heuristic to speed up the computational time. For the instances in this research, we could gather exact solutions, but for bigger instances a heuristic might be necessary to complete experiments in a reasonable time. Using a metaheuristic to efficiently explore large solution spaces and find good solutions, would be the best addition to a VRP model which has a larger solution space.

Another limitation of the research is that the pick-up of products is not considered in the VRP model. This is due to the assumption that the total pick-up will always be less than the total delivery for the milk-run tours, combined with a broad capacity for the products. This means the assumption that demand for pick-up can always be satisfied is put in place. This is very likely, but in the case, demand increases or capacity decreases, it might be the case that for some tours of the milk-run coincidentally the demand in pick-up is suddenly very high. This might affect the solutions provided by the model, so future research could investigate incorporating this extension in the VRP model as well.

An extension that adds on the combination of pick-up and delivery, which could be incorporated to model the situation at Thales even better, is having intermediate demand between customers. This means that station A requests a certain product that is present at station B instead of the logistics centre. Then the possibility to immediately bring this product is available. Currently, this can be done when the sequence of the routing allows, otherwise the product is first stalled at the logistics centre and brought along with the next milk-run. This happens rarely, but it can be an element to take into account. This is not incredibly relevant for Thales as there is not a lot of freedom in the optimal routes and going from outdoor to indoor back to outdoor locations would not be worth it for these low levels of intermediate demand. But, for future research on this topic, where more freedom is present in the solution space, this can be very useful.

6.4 Contribution to literature and practice

In this section, the theoretical and practical contribution of the research is discussed. In general, this research adds on the optimization techniques of a milk-run handled as a Vehicle Routing Problem. It is not uncommon for a milk-run to be approached as a VRP. Gyulai et al. (2013) have created a clear base for approaching the milk-run as a VRP and this research continues on that, adapting it to the situation of the milk-run of Thales. This desired adaptation has led to a unique combination of extensions on a VRP. The extensions that are discussed, implemented and combined in this research are the Split Delivery VRP, a variant on the VRP with Soft Time Windows and the VRP with Compatibility Constraints. All in all, this creates the SDVRPSTWCC (Split Delivery Vehicle Routing Problem with (partial) Soft Time Windows and Compatibility Constraints).

To initialise the model that computes the solution for the VRP, some pre-processing techniques have been introduced. Grouping nodes (customers) in a cluster, duplicating them, and assigning each of them to separate time windows is also a unique technique setup to tackle the problem of having multiple instances of demand at the same location throughout a time period which also have different time windows. The research also adds empirical prove for the fact that a combination of linearized constraints and objective function performs better than a combination with quadratic constraints and/or quadratic objective function.

The practical contribution for Thales lies in the insights they get into the milk-run. Currently, the routing and frequency is done on gut feeling. The VRP model supports the decision-making concerning these topics and guarantees an optimal solution. This model can create a new routing when Thales wants. This can be in case of a new additional building or a big shift in terms of demand or capacity of the vehicle. Sensitivity analyses on various parameters show the impact of certain aspects on the performance of the milk-run, which they can keep in mind for future scenarios.

References

Archetti, C., & Speranza, M. G. (2008). The split delivery vehicle routing problem: A survey. *The vehicle routing problem: Latest advances and new challenges*, 103-122.

Baker, B. M., & Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research, 30*(5), 787-800.

Baudin, M. (2005). *Lean logistics: the nuts and bolts of delivering materials and goods*: CRC press.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & industrial engineering, 99*, 300-313.

Brar, G. S., & Saini, G. (2011). *Milk run logistics: literature review and directions.* Paper presented at the Proceedings of the world congress on engineering.

Brink, J. (2023). *Modeling and analyzing the use of electric vehicles for Van Schoot Pompcentrum BVs routing logistics.* University of Twente,

Chiang, W. C., & Cheng, C. Y. (2017). Considering the performance bonus balance in the vehicle routing problem with soft time windows. *Procedia manufacturing, 11*, 2156-2163.

Dahiya, C., & Sangwan, S. (2018). Literature review on travelling salesman problem. *International Journal of Research, 5*(16), 1152-1155.

Datta, S. (2024). Defining Multiobjective Algorithms and Pareto Frontiers. Retrieved from https://www.baeldung.com/cs/defining-multiobjective-algorithms-and-pareto-frontiers

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., & Soumis, F. (2002). VRP with Pickup and Delivery. *The vehicle routing problem, 9*, 225-242.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL), 37*(3), 383-402.

Droste, M., & Deuse, J. (2012). *A planning approach for in-plant milk run processes to optimize material provision in assembly systems.* Paper presented at the Enabling Manufacturing Competitiveness and Economic Sustainability: Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV2011), Montreal, Canada, 2-5 October 2011.

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science, 22*(3), 123-131.

Esparrago Jr, R. A. (1988). Kanban. *Production and Inventory Management Journal, 29*(1), 6.

Gyulai, D., Pfeiffer, A., Sobottka, T., & Váncza, J. (2013). Milkrun vehicle routing approach for shop-floor logistics. *Procedia Cirp, 7*, 127-132.

Harahap, R. F. (2023). *Study vehicle routing problem using Nearest Neighbor Algorithm.* Paper presented at the Journal of Physics: Conference Series.

Heerkens, H. (2017). *Solving Managerial Problems Systematically*.

Hersyah, M. H., & Derisma. (2018). *A Literature Review on Business Continuity Based on ISO 22301, Six Sigma and Customer Satisfaction Evaluation.* Paper presented at the 2018 International Conference on Information Technology Systems and Innovation, ICITSI 2018 - Proceedings.

Horowitz, E., Sahani, S., & Rajasekaran, S. (2007). *Fundamentals of computer algorithms*: Galgotia Publications.

Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). *Vehicle routing problem with time windows*: Springer.

Karakash, T. (2024). *Optimizing Meilink Borculo BV's Transportation Planning: Cost-Effective Logistics through Vehicle Routing.* University of Twente,

Klenk, E., Galka, S., & Günthner, W. A. (2015). Operating strategies for in-plant milk-run systems. *IFAC-PapersOnLine, 48*(3), 1882-1887.

Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants.

Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & industrial engineering, 59*(1), 157-165.

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics (NRL), 54*(8), 811-819.

Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research, 148*(3), 559-569.

Lenstra, J. K., & Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*(2), 221-227.

Mei, H., Jingshuai, Y., Teng, M., Xiuli, L., & Ting, W. (2017). The modeling of milk-run vehicle routing problem based on improved CW algorithm that joined time window. *Transportation research procedia, 25*, 716-728.

Ni, Q., & Tang, Y. (2023). A bibliometric visualized analysis and classification of vehicle routing problem research. *Sustainability, 15*(9), 7394.

Sabo, C., Kumar, M., Cohen, K., & Kingston, D. (2012). VRP with minimum delivery latency using linear programming. In *Infotech@ Aerospace 2012* (pp. 2561).

Salani, M., Battarra, M., & Gambardella, L. M. (2016). *Exact algorithms for the vehicle routing problem with soft time windows.* Paper presented at the Operations Research Proceedings 2014: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), RWTH Aachen University, Germany, September 2-5, 2014.

Setak, M., Azizi, V., Karimi, H., & Jalili, S. (2017). Pickup and delivery supply chain network with semi soft time windows: metaheuristic approach. *International Journal of Management Science and Engineering Management, 12*(2), 89-95.

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

Toneva, M. What Is a Pull System? A Practical Guide to Increasing Process Efficiency. Retrieved from https://businessmap.io/lean-management/pull/what-is-pull-system

Wicaksono, P. A., Puspitasari, D., Ariyandanu, S., & Hidayanti, R. (2020). *Comparison of simulated annealing, nearest neighbour, and tabu search methods to solve vehicle routing problems.* Paper presented at the IOP Conference Series: Earth and Environmental Science.

Yu, M., Nagarajan, V., & Shen, S. (2017). *Minimum makespan vehicle routing problem with compatibility constraints.* Paper presented at the International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems.

Appendix

A. Workcell interviews key take-aways

Sensors

The sensors workcell is fully located in the high hall of W0. Station 29 and its substations are present here. The substations are used for delivering directly at the workplace. Station 29 itself is used for other (regular) products. The shop floor of this workcell will increase as some other work lines are transferred to another location. More big radars will be built here, which might have implications on the route of the milk-run through the hall. The lay-out is still unclear, but it might not be desirable to drive through the building very often. On the other hand, it will be more beneficial for the sensor department if the milk-run visits more often and directly at the workplace, so employees can continue their work quicker and more efficiently.

Reverse Logistics (Overhauls)

The other workcell that is (partly) located at W0 is reverse logistics. They share the high hall of W0, where they use station 29 and also have a small location next to station 28 (W0 low hall) and in a different building. This last building barely requires any products so is not included in the milk-run. The storage space for reverse logistics is fairly low, some extra storage or delivering directly at the workstation would help tackle this. They share their opinion on passing through the whole hall with sensors.

Subs *(Micro-Electronics, Optronics & Subs)*

Subs is one of the three parts of the bigger workcell Micro-Electronics, Optronics & Subs. This part is located close to station 28 (W0 low hall), which is also the station they use to gather and send products from.

Racks & Cabinets

Another workcell which is located in the lower hall of W0 is racks & cabinets. They also make use of station 28 and use station 29 for sending bigger 'finished' products for pick up. Visiting this workcell more often would help them as they would need to use the 1-hour service a lot less. Furthermore, the workcell leader indicated that the milk-run causes some noise disturbance when passing along their workcell. It is therefore also not desired to drop off products at specific workstations as they work very precise sometimes, so the noise levels should be minimised.

Combat Systems Hardware (CSHW)

CSHW is the last workcell that is located in the low hall of W0. They use station 31 and its substations as these are located in the middle of their shop floor. Currently, too much inventory is placed at the station. They would prefer if more was stored at the central warehouse and the milk-run would visit more often to supply when needed. Also, a lot is sent by pallet, which should also be possible to take with the milk-run instead.

Aerospace & Building Blocks

The main workcell that is located on W1 is aerospace & building blocks. They use station 34, which is located inside of a lock between a hallway and the workcell. The products can be brought into the lock and placed on racks. The other station they use is 33 with its substations. This station is entered by taking the elevator to another lock, then putting on a jacket and entering through two doors. Stopping at more workstations would be beneficial for this workcell as well. This way the products can be divided over more workstations, so employees can keep on working and less storage is needed. Currently, it sometimes takes too long for products to arrive. Having more frequent milk-runs would tackle this problem.

Micro-Electronics ***(Micro-Electronics, Optronics & Subs)***
Micro-Electronics is the second of the three parts of the bigger workcell Micro-Electronics, Optronics & Subs. This part is located on W1, behind the aerospace and building blocks workcell. Another door has to be passed to access this area. It is therefore more convenient for the milk-run to drop and pick up the products at the racks at station 34 as well. 1-hour service is used a lot as well, because otherwise their production cannot continue.

Service Parts Assembly, Repair & Test
The last workcell on W1 is Service Parts Assembly, Repair & Test. They have a part of this floor as well. Station 34 is their main station that they use, but sometimes station 35 is used for bigger loads. 1-hour service is used a lot here as well, because otherwise their production cannot continue. And station 34 needs to be checked often in order to create space for the next milk-run. This is due to the limited storage space for them at station 34.

Mechanics & Inspection
There are two workcells in building GCC-PCB. Chemistry & Clean room does not use the milk-run standard, but can be visited at station 36 when a part is suddenly added to the milk-run vehicle. But Mechanics & Inspection does make use of it. They use station 3 for pickup and drop-off. Internal transport along workstations is not needed as they have a flow production themselves. Some extra space can be helpful, but visiting the location more often to get rid of stored products will have the same effect. Sometimes delivery to other departments is done by themselves instead of the milk-run, because this can take too long and the (internal) customer might be waiting for it. When the milk-run visits more often, this will partly be tackled.

Track Sensor Systems
Currently, the workcell Track Sensor Systems is located in building Z. This will be transferred to the new building STC. They now use the stations 40 and 32.1, but this gets irrelevant when it is transferred. In the new building, driving along the workstations is a possibility and is desired.

Search Sensor Systems
The workcell search sensor systems is divided over two buildings. In building W0, inside the high hall, the STIR (one radar) is made. This will be relocated to the new building, where it will be put in a flow line. The other part of the workcell is located in building Z. Here other radars are built. This building layout will be redesigned to create more capacity for these radars as the production will go up. Currently, the milk-run stays outside for this workcell, but the new layout will allow the train to go inside and deliver directly at the workstation.

Optronics ***(Micro-Electronics, Optronics & Subs)***
Optronics is the third of the three parts of the bigger workcell Micro-Electronics, Optronics & Subs. This part is located on X1. This building is quite remote from the rest of the buildings. That is acceptable, because this building barely has to be visited by the milk-run as their demand and supply is low.

B. Data milk-run overview

i. Total pickup and delivery of products

Building	Station	Pickup 10.00 AM	Pickup 2.00 PM	Delivery 10.00 AM	Delivery 2.00 PM	Pickup	Delivery	Total
LC	1	141	109	61	40	250	101	351
	12	0	0	4	7	0	11	11
	Spares	0	0	6	0	0	6	6
	Outbound	0	0	3	1	0	4	4
W0	23	9	0	2	6	9	8	17
	28	0	0	23	17	0	40	40
	29	11	10	20	14	21	34	55
	31 Train	6	0	12	8	6	20	26
	31 Heftruck	0	0	0	0	0	0	0
	SPH	0	0	3	0	0	3	3
W1	33	16	11	16	7	27	23	50
	34	14	26	23	25	40	48	88
	35	1	0	8	10	1	18	19
Z0	32	0	0	1	1	0	2	2
	40	0	1	6	18	1	24	25
	Outside test	0	0	0	0	0	0	0
Z1	32.1 Train	0	0	0	0	0	0	0
	32.1 Heftruck	0	0	5	0	0	5	5
GCC-PCB	3	9	8	10	9	17	19	36
	36	0	0	4	0	0	4	4
X1	X1	0	0	0	0	0	0	0

ii. Total count of visits for pickup and delivery of products per station

		Pickup		Delivery				
Building	Station	10.00 AM	2.00 PM	10.00 AM	2.00 PM	10.00 AM	2.00 PM	Day
LC	1	5	5	5	5	5	5	10
	12	0	0	3	3	3	3	6
	Spares	0	0	1	0	1	0	1
	Outbound	0	0	1	1	1	1	2
W0	23	2	0	1	4	3	4	7
	28	0	0	5	5	5	5	10
	29	4	3	4	5	5	5	10
	31 Train	4	0	4	4	4	4	8
	31 Heftruck	0	0	0	0	0	0	0
	SPH	0	0	2	0	2	0	2
W1	33	5	3	5	3	5	4	9
	34	4	4	5	5	5	5	10
	35	1	0	4	4	4	4	8
Z0	32	0	0	1	1	1	1	2
	40	0	1	2	3	2	3	5
	Outside test	0	0	0	0	0	0	0
Z1	32.1 Train	0	0	0	0	0	0	0
	32.1 Heftruck	0	0	2	0	2	0	2
GCC-PCB	3	4	5	4	4	5	5	10
	36	0	0	3	0	3	0	3
X1	X1	0	0	0	0	0	0	0

C. Vehicle Requirements

When determining the optimal solution for Thales, we need to keep as much freedom as possible. This means, we should also investigate the possibility of combining the inside and outside route with one vehicle, instead of keeping the current vehicles that are used for the milk-run and keeping a division between the inside and outside locations, which means there is a minimum connection. Therefore, it is important to establish the requirements for a new vehicle for future scenarios. The requirements of the vehicle should fit the desires of Thales. These are the following:

- Shock absorption must be incorporated in the vehicle (the vehicle must be able to drive on a bumpy road)
- It must be able to tow at least 8000 kg
- The vehicle must fit through the current lanes present on the shop floors of Thales
- The current carts must fit on the wagons of the vehicle or new carts have to be designed to fit the boxes Thales uses for its transport
- The vehicle and its wagons must be waterproof (when driving outside in the rain)

D. Results experiments

1. Current situation

Solution: 3,374

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2482
Arrival times: {0: 37534.0, 2: 35895.0, 4: 37354.0, 5: 37064.0, 6: 36414.0, 7: 35910.0, 8: 36189.0, 11:
36399.0, 12: 35880.0, 13: 36204.0, 14: 37339.0, 15: 37079.0}
Departure times: {0: 35712.0, 2: 36015.0, 4: 37384.0, 5: 37124.0, 6: 36714.0, 7: 36030.0, 8: 36309.0, 11:
36699.0, 12: 36000.0, 13: 36324.0, 14: 37369.0, 15: 37139.0}
Total delivered demand: 36.0
Total demand per customer : {'W0': 4.0, 'W1.1': 18.0, 'W1.2': 2.0, 'GCC': 7.0, 'Z': 5.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2692
Arrival times: {0: 55504.0, 1: 54399.0, 3: 54174.0, 16: 54414.0, 17: 53880.0, 18: 54159.0, 19: 55324.0,
20: 55064.0, 21: 54384.0, 22: 53880.0, 23: 54189.0}
Departure times: {0: 53712.0, 1: 54699.0, 3: 54294.0, 16: 54714.0, 17: 54000.0, 18: 54279.0, 19:
55354.0, 20: 55124.0, 21: 54684.0, 22: 54000.0, 23: 54309.0}
Total delivered demand: 38.0
Total demand per customer : {'W0': 24.0, 'W1.1': 3.0, 'W1.2': 9.0, 'GCC': 1.0, 'Z': 1.0}
```

2. Current situation

i. Current situation – Restricted freedom

Solution: 5,615

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'LC')]
Duration: 1278
Arrival times: {0: 29790.0, 6: 28884.0, 7: 28680.0, 8: 29259.0}
Departure times: {0: 28512.0, 6: 29184.0, 7: 28800.0, 8: 29379.0}
Total delivered demand: 4.0
 Total demand per customer : {'W0': 2.0, 'W1.1': 1.0, 'W1.2': 1.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1852
Arrival times: {0: 37459.0, 3: 36159.0, 11: 36369.0, 12: 35880.0, 13: 36174.0, 14: 37279.0, 15: 37019.0}
Departure times: {0: 35712.0, 3: 36279.0, 11: 36669.0, 12: 36000.0, 13: 36294.0, 14: 37309.0, 15:
37079.0}
Total delivered demand: 14.0
Total demand per customer : {'W0': 2.0, 'W1.1': 2.0, 'W1.2': 8.0, 'GCC': 1.0, 'Z': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2242
```

```
Arrival times: {0: 50550.0, 1: 49430.0, 2: 48926.0, 4: 50355.0, 5: 50080.0, 16: 49415.0, 17: 48941.0, 18: 49220.0, 19: 50370.0, 20: 50095.0}
Departure times: {0: 48758.0, 1: 49730.0, 2: 49046.0, 4: 50385.0, 5: 50140.0, 16: 49715.0, 17: 49061.0, 18: 49340.0, 19: 50400.0, 20: 50155.0}
Total delivered demand: 50.0
Total demand per customer : {'W0': 20.0, 'W1.1': 17.0, 'W1.2': 1.0, 'GCC': 7.0, 'Z': 5.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'LC')]
Duration: 1278
Arrival times: {0: 54990.0, 21: 54354.0, 22: 53880.0, 23: 54159.0}
Departure times: {0: 53712.0, 21: 54654.0, 22: 54000.0, 23: 54279.0}
Total delivered demand: 6.0
Total demand per customer : {'W0': 4.0, 'W1.1': 1.0, 'W1.2': 1.0}
```

ii. Current situation – Full freedom

Solution: 5,615

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'LC')]
Duration: 1518
Arrival times: {0: 32811.0, 2: 31686.0, 3: 32265.0, 6: 31890.0, 7: 31671.0, 8: 32280.0}
Departure times: {0: 31503.0, 2: 31806.0, 3: 32385.0, 6: 32190.0, 7: 31791.0, 8: 32400.0}
Total delivered demand: 26.0
Total demand per customer : {'W0': 2.0, 'W1.1': 16.0, 'W1.2': 8.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2092
Arrival times: {0: 39750.0, 1: 38630.0, 5: 39295.0, 11: 38645.0, 12: 38156.0, 13: 38435.0, 14: 39570.0, 15: 39310.0}
Departure times: {0: 37988.0, 1: 38930.0, 5: 39355.0, 11: 38945.0, 12: 38276.0, 13: 38555.0, 14: 39600.0, 15: 39370.0}
Total delivered demand: 31.0
Total demand per customer : {'W0': 22.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 5.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1762
Arrival times: {0: 50464.0, 4: 50284.0, 16: 49359.0, 17: 48885.0, 18: 49164.0, 19: 50269.0, 20: 50009.0}
Departure times: {0: 48717.0, 4: 50314.0, 16: 49659.0, 17: 49005.0, 18: 49284.0, 19: 50299.0, 20: 50069.0}
Total delivered demand: 14.0
Total demand per customer : {'W0': 3.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 7.0, 'Z': 1.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'LC')]
Duration: 1278
```

```
Arrival times: {0: 59736.0, 21: 59100.0, 22: 58626.0, 23: 58905.0}
Departure times: {0: 58458.0, 21: 59400.0, 22: 58746.0, 23: 59025.0}
Total delivered demand: 3.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 1.0}
```

iii. Current situation - 1 vehicle

Solution: 5,264

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'LC')]
Duration: 1282
Arrival times: {0: 29794.0, 5: 29634.0, 7: 29154.0, 8: 28680.0, 9: 28959.0}
Departure times: {0: 28512.0, 5: 29694.0, 7: 29454.0, 8: 28800.0, 9: 29079.0}
Total delivered demand: 8.0
Total demand per customer : {'W0': 2.0, 'W1.1': 1.0, 'W1.2': 1.0, 'Z': 4.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1682
Arrival times: {0: 37289.0, 2: 35895.0, 13: 36369.0, 14: 35880.0, 15: 36174.0, 16: 37109.0, 17: 36849.0}
Departure times: {0: 35712.0, 2: 36015.0, 13: 36669.0, 14: 36000.0, 15: 36294.0, 16: 37139.0, 17:
36909.0}
Total delivered demand: 22.0
Total demand per customer : {'W0': 2.0, 'W1.1': 17.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2012
Arrival times: {0: 48119.0, 1: 47169.0, 3: 46974.0, 4: 47924.0, 19: 47184.0, 20: 46680.0, 21: 46959.0, 22:
47939.0, 23: 47664.0}
Departure times: {0: 46512.0, 1: 47469.0, 3: 47094.0, 4: 47954.0, 19: 47484.0, 20: 46800.0, 21: 47079.0,
22: 47969.0, 23: 47724.0}
Total delivered demand: 41.0
Total demand per customer : {'W0': 23.0, 'W1.1': 2.0, 'W1.2': 8.0, 'GCC': 7.0, 'Z': 1.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'LC')]
Duration: 1278
Arrival times: {0: 59811.0, 25: 58905.0, 26: 58701.0, 27: 59280.0}
Departure times: {0: 58533.0, 25: 59205.0, 26: 58821.0, 27: 59400.0}
Total delivered demand: 3.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 1.0}
```

3. Future situation

i. Future situation – 1 vehicle

Solution: 5,803

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'LC')]
```

```
Duration: 1412
Arrival times: {0: 29879.0, 5: 29719.0, 6: 29569.0, 7: 28884.0, 8: 28680.0, 9: 29259.0, 12: 29554.0}
Departure times: {0: 28512.0, 5: 29779.0, 6: 29629.0, 7: 29184.0, 8: 28800.0, 9: 29379.0, 12: 29614.0}
Total delivered demand: 18.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 1.0, 'Z': 4.0, 'STC': 11.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2052
Arrival times: {0: 37374.0, 1: 36369.0, 3: 36159.0, 13: 36384.0, 14: 35880.0, 15: 36174.0, 16: 37194.0,
17: 36934.0, 18: 36784.0}
Departure times: {0: 35712.0, 1: 36669.0, 3: 36279.0, 13: 36684.0, 14: 36000.0, 15: 36294.0, 16:
37224.0, 17: 36994.0, 18: 36844.0}
Total delivered demand: 35.0
Total demand per customer : {'W0': 22.0, 'W1.1': 2.0, 'W1.2': 8.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1632
Arrival times: {0: 48144.0, 19: 47154.0, 20: 46680.0, 21: 46959.0, 22: 47964.0, 23: 47704.0, 24: 47554.0}
Departure times: {0: 46512.0, 19: 47454.0, 20: 46800.0, 21: 47079.0, 22: 47994.0, 23: 47764.0, 24:
47614.0}
Total delivered demand: 9.0
Total demand per customer : {'W0': 2.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 2.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'GCC'), ('GCC', 'LC')]
Duration: 1742
Arrival times: {0: 55334.0, 2: 53895.0, 4: 55154.0, 25: 54099.0, 26: 53880.0, 27: 54474.0, 28: 55139.0,
30: 54769.0}
Departure times: {0: 53712.0, 2: 54015.0, 4: 55184.0, 25: 54399.0, 26: 54000.0, 27: 54594.0, 28:
55169.0, 30: 54829.0}
Total delivered demand: 26.0
Total demand per customer : {'W0': 1.0, 'W1.1': 16.0, 'W1.2': 1.0, 'GCC': 7.0, 'STC': 1.0}
```

ii. Future situation – 2 vehicles

Solution: 7,115

```
Vehicle 1: Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'LC')]
Duration: 2040
Arrival times: {0: 30250.0, 1: 28927.0, 5: 30090.0, 6: 29925.0, 7: 28912.0, 8: 28708.0, 9: 29302.0, 12:
29940.0}
Departure times: {0: 28540.0, 1: 29227.0, 5: 30150.0, 6: 29985.0, 7: 29212.0, 8: 28828.0, 9: 29422.0, 12:
30000.0}
Total delivered demand: 38.0
Total demand per customer : {'W0': 21.0, 'W1.1': 1.0, 'W1.2': 1.0, 'Z': 4.0, 'STC': 11.0}

Vehicle 2: Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'GCC'),
('GCC', 'LC')]
```

```
Duration: 1960
Arrival times: {0: 37672.0, 13: 36084.0, 14: 35880.0, 15: 36459.0, 16: 37492.0, 17: 37232.0, 18: 37082.0}
Departure times: {0: 35712.0, 13: 36384.0, 14: 36000.0, 15: 36579.0, 16: 37522.0, 17: 37292.0, 18:
37142.0}
Total delivered demand: 8.0
Total demand per customer : {'W0': 2.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1960
Arrival times: {0: 48472.0, 19: 46884.0, 20: 46680.0, 21: 47259.0, 22: 48292.0, 23: 48032.0, 24: 47882.0}
Departure times: {0: 46512.0, 19: 47184.0, 20: 46800.0, 21: 47379.0, 22: 48322.0, 23: 48092.0, 24:
47942.0}
Total delivered demand: 9.0
Total demand per customer : {'W0': 2.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 2.0}

Vehicle 4: Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'GCC'), ('GCC', 'LC')]
Duration: 2190
Arrival times: {0: 55677.0, 2: 53895.0, 3: 54489.0, 4: 55482.0, 25: 54099.0, 26: 53880.0, 27: 54474.0, 28:
55497.0, 30: 55112.0}
Departure times: {0: 53712.0, 2: 54015.0, 3: 54609.0, 4: 55512.0, 25: 54399.0, 26: 54000.0, 27: 54594.0,
28: 55527.0, 30: 55172.0}
Total delivered demand: 33.0
Total demand per customer : {'W0': 1.0, 'W1.1': 16.0, 'W1.2': 8.0, 'GCC': 7.0, 'STC': 1.0}
```

4. Benchmark routing

i. Benchmark routing – Current situation

Solution: 6,523

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1762
Arrival times: {0: 30259.0, 4: 30064.0, 6: 29154.0, 7: 28680.0, 8: 28959.0, 9: 30079.0, 10: 29804.0}
Departure times: {0: 28512.0, 4: 30094.0, 6: 29454.0, 7: 28800.0, 8: 29079.0, 9: 30109.0, 10: 29864.0}
Total delivered demand: 10.0
Total demand per customer : {'W0': 2.0, 'W1.1': 1.0, 'W1.2': 1.0, 'GCC': 6.0, 'Z': 0.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2032
Arrival times: {0: 37459.0, 1: 36354.0, 11: 36369.0, 12: 35880.0, 13: 36159.0, 14: 37279.0, 15: 37019.0}
Departure times: {0: 35712.0, 1: 36654.0, 11: 36669.0, 12: 36000.0, 13: 36279.0, 14: 37309.0, 15:
37079.0}
Total delivered demand: 27.0
Total demand per customer : {'W0': 22.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
```

Duration: 1912
Arrival times: {0: 48274.0, 3: 46959.0, 5: 47819.0, 16: 47169.0, 17: 46680.0, 18: 46974.0, 19: 48094.0, 20: 47834.0}
Departure times: {0: 46512.0, 3: 47079.0, 5: 47879.0, 16: 47469.0, 17: 46800.0, 18: 47094.0, 19: 48124.0, 20: 47894.0}
Total delivered demand: 19.0
Total demand per customer : {'W0': 3.0, 'W1.1': 2.0, 'W1.2': 8.0, 'GCC': 1.0, 'Z': 5.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1852
Arrival times: {0: 55444.0, 2: 53865.0, 21: 54354.0, 22: 53880.0, 23: 54159.0, 24: 55264.0, 25: 55004.0}
Departure times: {0: 53697.0, 2: 53985.0, 21: 54654.0, 22: 54000.0, 23: 54279.0, 24: 55294.0, 25: 55064.0}
Total delivered demand: 18.0
Total demand per customer : {'W0': 1.0, 'W1.1': 16.0, 'W1.2': 1.0, 'GCC': 0.0, 'Z': 0.0}

ii. Benchmark routing – Future situation – 1 vehicle

Solution 10: 6,138
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1782
Arrival times: {0: 30174.0, 3: 29274.0, 4: 29979.0, 7: 28884.0, 8: 28680.0, 9: 29259.0, 10: 29994.0, 11: 29719.0, 12: 29569.0}
Departure times: {0: 28512.0, 3: 29394.0, 4: 30009.0, 7: 29184.0, 8: 28800.0, 9: 29379.0, 10: 30024.0, 11: 29779.0, 12: 29629.0}
Total delivered demand: 17.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 8.0, 'GCC': 6.0, 'Z': 0.0, 'STC': 1.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1932
Arrival times: {0: 37359.0, 1: 36099.0, 13: 36084.0, 14: 35880.0, 15: 36474.0, 16: 37179.0, 17: 36919.0, 18: 36769.0}
Departure times: {0: 35712.0, 1: 36399.0, 13: 36384.0, 14: 36000.0, 15: 36594.0, 16: 37209.0, 17: 36979.0, 18: 36829.0}
Total delivered demand: 28.0
Total demand per customer : {'W0': 22.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1692
Arrival times: {0: 48159.0, 5: 47704.0, 19: 47154.0, 20: 46680.0, 21: 46959.0, 22: 47979.0, 23: 47719.0, 24: 47554.0}
Departure times: {0: 46512.0, 5: 47764.0, 19: 47454.0, 20: 46800.0, 21: 47079.0, 22: 48009.0, 23: 47779.0, 24: 47614.0}
Total delivered demand: 13.0
Total demand per customer : {'W0': 2.0, 'W1.1': 2.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 5.0, 'STC': 2.0}

```
Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 1812
Arrival times: {0: 55374.0, 2: 53895.0, 6: 54784.0, 25: 54369.0, 26: 53880.0, 27: 54174.0, 28: 55194.0,
29: 54934.0, 30: 54769.0}
Departure times: {0: 53712.0, 2: 54015.0, 6: 54844.0, 25: 54669.0, 26: 54000.0, 27: 54294.0, 28:
55224.0, 29: 54994.0, 30: 54829.0}
Total delivered demand: 30.0
Total demand per customer : {'W0': 1.0, 'W1.1': 16.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 0.0, 'STC': 11.0}
```

iii. Benchmark routing – Future situation – 2 vehicles

Solution: 7,450

```
Vehicle 1:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2020
Arrival times: {0: 30487.0, 5: 30047.0, 7: 29154.0, 8: 28680.0, 9: 28959.0, 10: 30307.0, 11: 30032.0, 12:
29882.0}
Departure times: {0: 28512.0, 5: 30107.0, 7: 29454.0, 8: 28800.0, 9: 29079.0, 10: 30337.0, 11: 30092.0,
12: 29942.0}
Total delivered demand: 8.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 1.0, 'GCC': 0.0, 'Z': 4.0, 'STC': 1.0}

Vehicle 2:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2410
Arrival times: {0: 37717.0, 1: 36384.0, 3: 36159.0, 4: 37522.0, 13: 36369.0, 14: 35880.0, 15: 36174.0, 16:
37537.0, 17: 37262.0, 18: 37112.0}
Departure times: {0: 35712.0, 1: 36684.0, 3: 36279.0, 4: 37552.0, 13: 36669.0, 14: 36000.0, 15: 36294.0,
16: 37567.0, 17: 37322.0, 18: 37172.0}
Total delivered demand: 41.0
Total demand per customer : {'W0': 22.0, 'W1.1': 2.0, 'W1.2': 8.0, 'GCC': 7.0, 'Z': 1.0, 'STC': 1.0}

Vehicle 3:
Route: [('LC', 'W1.1'), ('W1.1', 'W1.2'), ('W1.2', 'W0'), ('W0', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2080
Arrival times: {0: 48487.0, 2: 46695.0, 19: 47169.0, 20: 46680.0, 21: 46974.0, 22: 48307.0, 23: 48047.0,
24: 47897.0}
Departure times: {0: 46512.0, 2: 46815.0, 19: 47469.0, 20: 46800.0, 21: 47094.0, 22: 48337.0, 23:
48107.0, 24: 47957.0}
Total delivered demand: 24.0
Total demand per customer : {'W0': 2.0, 'W1.1': 17.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 1.0, 'STC': 2.0}

Vehicle 4:
Route: [('LC', 'W1.1'), ('W1.1', 'W0'), ('W0', 'W1.2'), ('W1.2', 'STC'), ('STC', 'Z'), ('Z', 'GCC'), ('GCC', 'LC')]
Duration: 2020
Arrival times: {0: 55700.0, 6: 55095.0, 25: 54097.0, 26: 53893.0, 27: 54472.0, 28: 55520.0, 29: 55260.0,
30: 55110.0}
```

```
Departure times: {0: 53725.0, 6: 55155.0, 25: 54397.0, 26: 54013.0, 27: 54592.0, 28: 55550.0, 29:
55320.0, 30: 55170.0}
Total delivered demand: 15.0
Total demand per customer : {'W0': 1.0, 'W1.1': 1.0, 'W1.2': 1.0, 'GCC': 1.0, 'Z': 0.0, 'STC': 11.0}
```