Optimizing transportation in the network of food banks in the region of Twente-Salland based on the Vehicle Routing Problem.

Student:
K. Schotman (Kady)
s2335026
k.schotman@student.utwente.nl

University supervisors:
Dr.ir. E.A. Lalla (Eduardo)
Ing. J.P.S. Piest (Sebastian)
Company supervisor:
H. Ganseman (Henny)
Chairman Voedselbank Almelo

Department of Industrial Engineering & Business Information Systems

August 19, 2022

Management Summary

This research is conducted at the food bank in Almelo. The food bank is a non-profit organization, to provide people with little financial means food aid. This means that they aim to form food packages that contain different products, to fulfill the nutritional needs a human has. They do this by collaborating with companies, the township and volunteers. Like the food bank in Almelo, there are another 170 food banks in the Netherlands doing the same work. Currently, a sixth of the annual budget of the food bank in Almelo is spend on transport of products between their location and the Regional Distribution Centre (RDC) in Deventer. This RDC supplies food to another ten food banks in the region of Twente-Salland. Each of them travels to the RDC multiple times a week to pick up food, and occasionally to deliver food in case of a donation larger than their needs. This means that half of the trips they make are empty. Due to the large influence this has on the annual budget of the food banks, it is important that a more efficient method of transportation is found.

This transportation problem of the food banks can be formulated as the Vehicle Routing Problem (VRP), which can extensively be found in literature. Specifically, a multi-commodity VRP with time windows and heterogeneous fleet of vehicles forms the basis of the transportation problem faced by the food banks. However, the biggest difference between the VRP and the problem the food banks are facing, is the location of the vehicles. In VRP literature, the vehicles are located at the depot (comparable to the RDC) and routed to visit the customers (the food banks) in the most efficient manner. Here the customers (food banks) are the ones with the vehicles, and visiting the depot (RDC). Furthermore, not all products can be delivered with the use of the same vehicle due to food safety regulations.

The goal of this research is to optimize the transport between the RDC and the food banks. In doing so, a Mixed Integer Linear Program (MILP) has been developed taking the several constraints into consideration. Unfortunately, a MILP is not able to solve large instances in an acceptable time. Therefore a metaheuristic, the Variable Neighborhood Search (VNS) algorithm to solve this problem is developed. Unlike the MILP, the VNS is able to take the supply towards the RDC into consideration, but it is also less strict on multiple usages of the same vehicle. Both the MILP and VNS have been fine tuned with the use of six test data instances. For the MILP, it was concluded that a maximum running time of 600 seconds results in decent solutions while have an acceptable running time. The VNS is consisting of an initialization, shaking phase and local search phase, for which the parameters had to be determined. Thus, experiments for the parameter settings were conducted and it was concluded to use a randomized initialization, have an extensive shaking phase leading to solutions further away, and ordered the operators used for the local search.

Once the settings for both the MILP and VNS were chosen, numerical experiments with different scenarios were conducted. As two of the food banks, Vaassen and Zutphen, have notified the others that they are not willing to collaborate, they have been excluded for most of the scenarios. The scenarios can be summarized to four types: a collaboration with their own vehicles, a collaboration with their own vehicles where Vaassen and Zutphen do join, transport beginning and ending at the RDC with new vehicles, and the new vehicles being

distributed over the four largest food banks. The results are summarized in Table 1. It must be noted that the costs of the new vehicles, and their cost per km, used in these calculations are based on 2022, while the other costs were based on 2021.

Table 1. Overview of the daily costs. On the extended days the food banks in Vaassen and/or Zutphen are also taken into consideration. A dash means that no result has been found.

Day	Original(€)	MILP (€)	Improvement (%)	VNS (€)	Improvement (%)
Wednesday	226.09	237.11	-4.87	173.58	23.23
Wednesday extended	243.41	259.09	-6.44	180.41	25.88
Wednesday centralized	226.09	369.30	-63.34	-	-
Wednesday new vehicles	226.09	-	-	278.42	-23.15
Thursday	203.35	186.68	8.20	223.14	-9.73
Thursday extended	246.10	203.20	17.43	238.55	3.07
Thursday centralized	203.35	366.07	-80.02	-	-
Thursday new vehicles	203.35	368.28	-81.11	-	-
Friday	137.97	-	-	104.99	23.90
Friday centralized	137.97	363.69	-163.60	-	-
Friday new vehicles	137.97	-	-	-	-

From these results it follows that there is clearly room for improvement, up to 25% of the current costs involved in the transportation between food banks and the RDC. However, it does not become clear whether it is best to use the MILP or the VNS. It is scenario dependent, as both have their limitations, which occasionally means that no solution can be found. What is clear is that the expenses of the vehicles are leading in what is the best solution. As the new vehicles are significantly more expensive, they do not improve, but instead worsen, the costs involved. It is therefore recommended, based on the data used, to find a collaboration between the food banks with their existing fleet of vehicles. When the costs of the current vehicles increase, it may be better to use the new vehicles. This is a trade-off that has to be made by the food banks. In the generated routes, the demand of each food bank and capacity of the vehicles are leading. Meaning that these characteristics are of the biggest influence on the found routes. All in all it is recommended that the food banks decide on the use of new vehicles or not, knowing that the costs of the current vehicles will increase, and base their routes on that. Regardless of vehicle type, it is recommended that they do start a collaboration when it comes to transport between themselves and the RDC, as costs can be reduced up to 25%.

Preface

Before you lies my master thesis which marks the end of my time as a student. After obtaining my bachelors degree at the University of Technology Eindhoven I made the move to join the University of Twente. It took a semester to realize I wanted to study Industrial Engineering and Management, but the study counselors and program directors have been supportive from the start. Shortly after finding the right study, COVID-19 showed up, which means that most of my time studying at the UT has been online. Luckily, the last few months, while working on this thesis I have been able to work on site. I want to thank the volunteers of Voedselbanken Nederland, especially those volunteering in Almelo and at the RDC in Deventer. Not only for allowing me to work on site, but also because they have always been willing to help me with any questions, and shown an interest in my work. I specifically want to thank Frans van Nijenhof and Henny Ganseman for guiding me through this process.

I also want to thank my supervisors from the University, Eduardo and Sebastian. They have helped me shape this research, asked critical questions to improve this thesis. They have helped steer this research in the right direction, and frame the problem. With their help this thesis would not have been possible.

Lastly, I want to thank my family and friends whom have supported me all this time, through ups and downs. They have let me discover my interests, shown me around town and made sure I took some time to relax. I would not have been able to do this without you.

I hope you enjoy reading my thesis.

Kady Schotman
August 2022

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

CVRP Capacitated Vehicle Routing Problem

GRASP Greedy Randomized Adaptive Search Process

KPI Key Performance Indicator

MCVRP Vehicle Routing Problem with Multiple Commodities

MDVRP Multi-Depot Vehicle Routing Problem

MILP Mixed Integer Linear Program

NVWA Nederlandse Voedsel- en Warenautoriteit[1]

RDC Regional Distribution Centre

VNS Variable Neighborhood Search

VRP Vehicle Routing Problem

VRPB Vehicle Routing Problem with Backhauls

VRPTW Vehicle Routing Problem with Time Windows

[1] English translation: Netherlands Food and Consumer Product Safety Authority

1 Introduction

First, a description about the food bank is provided. This description aims to explain what the food banks do, and what their goals are. Next, the problem statement is given and described in Section 1.2. This results in the research goal, which is provided in Section 1.3. Based on the problem statement and research goal, the research questions are determined. The research questions are given in Section 1.4. Lastly, Section 1.5 describes the research design that is followed in this thesis.

1.1 About the food bank

In the Netherlands, over a million people live below the poverty line (Voedselbanken Nederland, 2021d). This means that these people do not have access to healthy food, a home, healthcare or education (Voedselbanken Nederland, 2021a). In order to help these people, food banks work together with several businesses, organizations, government agencies and individuals, with the aim of providing food to those people in financial need. Besides providing food, they also decrease the food surpluses and lower the burden on the environment (Voedselbanken Nederland, 2021b). In the Netherlands, there are 171 food banks. Each of these food banks is a charitable, autonomous, non-profit organization. Together these food banks have 13.000 volunteers and helped 160.500 people per year (Voedselbanken Nederland, 2021b). Most food banks have an ANBI-status, which means that donations are tax-deductible (Voedselbanken Nederland, 2022). This also means that their finances are publicly available when asked for.

1.1.1 Voedselbanken Nederland

Each of the 171 food banks in the Netherlands is affiliated to the umbrella organization, Voedselbanken Nederland. Unlike the food banks which are foundations, Voedselbanken Nederland is an assocation. However, they also have the ANBI-status (Voedselbanken Nederland, 2022). These food banks are also affiliated to a Regional Distribution Centre (RDC). In total there are ten RDCs affiliated with Voedselbanken Nederland. As an umbrella organization Voedselbanken Nederland is there to help and support the local food banks. In addition they set the national guidelines and direction of the organization. They are also responsible for national campaigns, such as Missie 538 in 2020 (Voedselbanken Nederland, 2021c), during which more than 1 million euros were raised. This money can be used to support food banks that are falling short financially, or to make investments that help the local food banks in their way of working.

1.1.2 The food bank in Almelo

This research is done at the food bank in Almelo. With the use of 85 volunteers, they help 500 people from 200 households in the city of Almelo (Voedselbank Almelo, 2021). Food bank Almelo is part of the region of Twente-Salland. The region of Twente-Salland has an RDC in Deventer and 11 food banks throughout the region. The food bank in Almelo is, like other food banks, financially supported by local companies who want to contribute to the good cause.

Figure 1 shows the different flows of food supply to and from the food bank in Almelo, as well as its flows to and from the RDC. This includes different local supermarkets, bakeries and other (not pictured) food (re-)sellers. They also receive food via the RDC in Deventer. Besides, they occasionally receive food from local food producers such as Bolletje[2]. As the food provided by such producers often comes in big quantities, the food bank in Almelo brings part of this supply to the RDC in Deventer, whom can divide that over the rest of the food banks in the region.

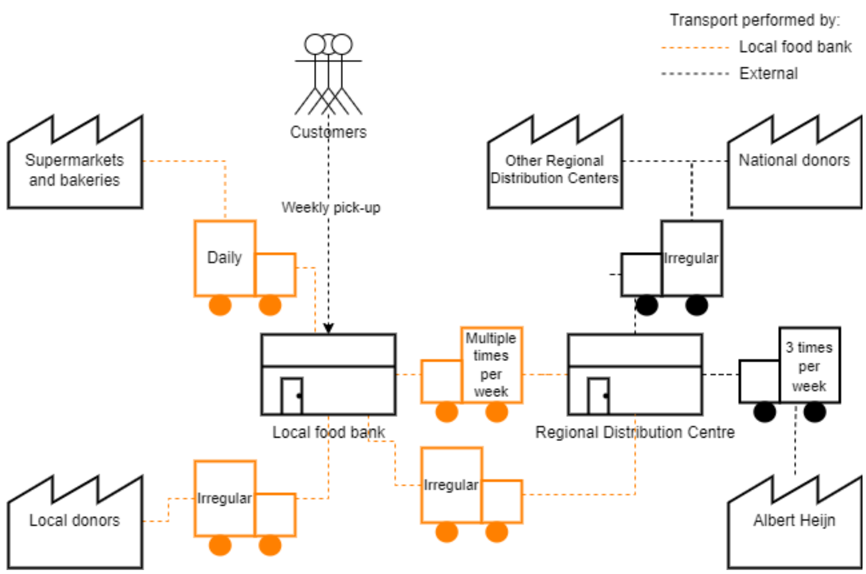

Figure 1. Simplified overview of the different transportation flows between the local food banks, RDC, suppliers and customers. The flows in orange are performed by the local food banks, while the flows in black are performed by external parties.

Figure 2 shows a more detailed overview of the transportation flows between the food bank in Almelo and the RDC in Deventer. Two to three times a week the food bank goes to Deventer to pick up supply. Occasionally, there are extra trips to bring supply to Deventer due to the supply from local donors. They also need to return the packaging, which happens at different moments. For the other food banks in the region this is similar, with more or less supply to and from the RDC.

1.1.3 Regional Distribution Centre in Deventer

The RDC in Deventer obtains nationally sourced products via the network of Voedselbanken Nederland. If a food producer has food left, they may offer it to Voedselbanken Nederland. It is up to the RDC to decide if they want to collect that type of product or not. Upon deciding to take this food, they organize transport via a partner company, or have it delivered to them via the food producer if possible. They cannot pick up supply themselves, as they

[2]Bolletje is a producer of bread substitutes and confectionery products. Bolletje is located in Almelo. https://bolletje.nl/

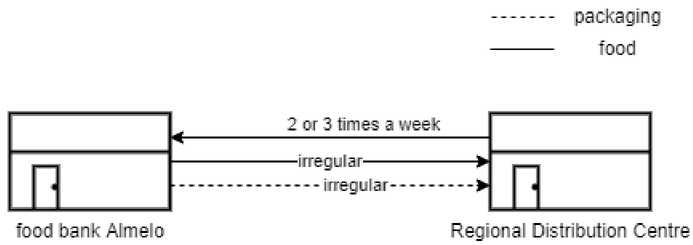

Figure 2. Transportation flows between the RDC and the food bank in Almelo are illustrated. The direction of the arrow is equal to the direction of the supply, which is either food or packaging materials.

do not have any modes of transportation.

The RDC in Deventer also gets deliveries from Albert Heijn[3]. That food supply needs to be handed to the RDCs on the same day. Albert Heijn usually delivers early in the day, so the RDC has time to divide the food over the local food banks that come for pick up that day.

All food that they receive in Deventer is divided over the different food banks in the region. They do this based on the number of people a food bank supplies. Occasionally, they may also hand out food to other RDCs and/or to the food bank in Zwolle. Oppositely, they may also receive some food from those organizations. However, that happens rarely.

The RDC is financially supported by Voedselbanken Nederland for the largest part. Besides the national funding, the local food banks need to support their respective RDC. The reason the RDC needs to be helped by the local food banks to because it does not easily get local companies to sponsor them. This is due to their work not being noticeable to the average person, even though they cannot be missed within the supply chain.

1.2 Problem Statement

In Figure 3, an overview of the problem is outlined. From this overview, it is noticeable that there is one big point of action for the food bank, namely the high transportation cost, and several sub problems that create this problem. These sub problems are discussed within this section.

[3]Albert Heijn has been the leading supermarket chain in the Netherlands for years (Distrifood, 2021), https://www.ah.nl/over-ah

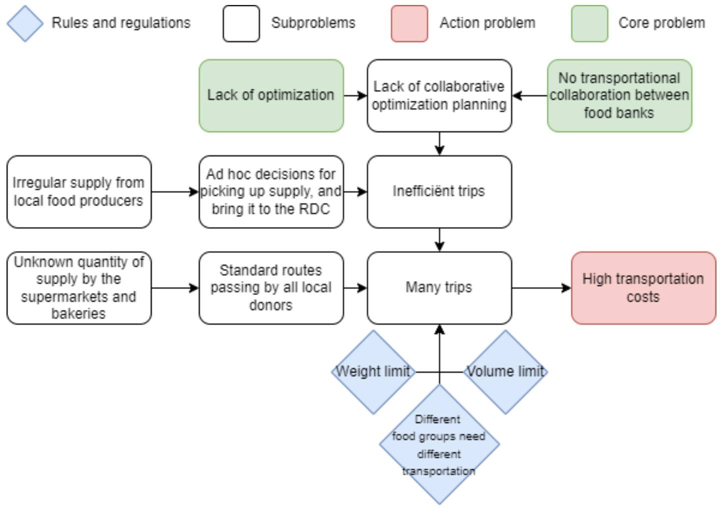

Figure 3. Problem cluster describing the core and sub problems at the food bank. The node in red is the action problem to be solved, which can be solved once the problems noted in the other nodes are solved. The blue diamond nodes are rules and regulations the food bank needs to adhere to and cannot be changed. In green are the core problems to be solved by this research.

1.2.1 Rules and regulations

The different food banks aim to create food packages containing products from the a practical information tool used by the Netherlands Nutrition Centre, known as the Schijf van vijf[4]. Some of the products have to remain frozen, to ensure that they are safe to eat. Others only have to be kept cool, while there are also dry groceries for which cooling is not needed. Like any other company and/or organization, the food bank has to comply with the rules set by the Nederlandse Voedsel- en Warenautoriteit[5] (NVWA). This means that, e.g., frozen and dry foods cannot be transported together, as the frozen food must remain frozen, while the dry foods may not be frozen. Besides, the drivers need to have special licenses in order to transport vehicles weighing more than 3500 kilograms (Rijksoverheid, 2021). This imposes a weight limit to the supply of foods per vehicle. Naturally, there also is a limit on the amount of volume which can be transported. Next to this, transportation of food happens on pallets and in crates. These packaging materials have monetary value and therefore it is important that the packaging received from the RDC is also returned to the RDC.

Although rules and regulations cannot be changed by the food bank, and will need to be adhered to, they provide limitations. As this limitations are significant, they need to be taken into consideration when solving the core problem. For this reason they have been explained in this section, as well as been added to Figure 3.

[4]English translation: wheel of five
[5]English translation: Netherlands Food and Consumer Product Safety Authority

1.2.2 Irregular supply

Another sub problem is the unexpected supply from food producers. When these food producers have food which they cannot sell anymore, e.g., due to demand changes, they ask the food bank to come pick it up. The logistics coordinator of the food bank responds ad hoc to this and determines a transport plan to go to the supplier. As these food producers aim to optimize their own production, there is no regularity in having (left over) food. Therefore, the supply from food producers to the food bank is irregular.

There is a more regular supply via the local supermarkets and bakeries. Most donate on a daily basis. However, whether these supermarkets and bakeries are able to donate something, and if so, how much, is not known beforehand. The food bank therefore makes use of standardized routes. Upon arrival at a supermarket or bakery the food bank will find out how much of, if any, supply there is.

What needs to be mentioned is that this means that there is a supply-driven supply chain. Food only goes from the RDC, local producers and supermarkets to the food banks if they have supply. The food banks thus do not have a say in how much they get from the RDC. However, as the food banks also sometimes supply the RDC this is confusing. For that reason, it is decided to name all down stream flows demand, and upstream flows supply.

1.2.3 Lack of data

The RDC in Deventer keeps track of the food they received, and what each local food bank receives. However, this is usually tracked in pieces and not in units of weight or volume. Furthermore they know what percentage of the products goes to which food bank. An example would be receiving a 100 boxes of cookies. These boxes are divided based on a predetermined division. The data they track is that they received a 100 boxes of cookies, and each food bank received their own percentage of those. On occasion the weight or volume may be known, but this is an exception.

The RDC in Deventer also tracks the packaging materials which they give to and receive from suppliers and local food banks. They know the amount of pallets (and which kind of pallet), and the amount of crates (and which crates) were sent and received from the different food banks, but also suppliers. What is not known is whether these were loaded or not, therefore it is unclear if the received pallets are just packaging materials or also contain food to be redistributed. This means that the available data is limited.

1.2.4 Lack of transportational collaboration

One of the problems is the lack of collaboration between the different local food banks. Although the food banks share food and knowledge, there is no collaboration in relation to the transport of food. Each food bank uses their own means of transportation between the RDC and themselves. Besides, when there is supply to be picked up at one of the local food banks, there is no collaborative transport between the food banks that will pick up (part of) the supply.

1.2.5 Lack of optimization

Finally, there is a lack of optimization for the transport between food banks and the RDC. Without optimization there is no guarantee that transport resources are efficiently used. Together with the lack of collaboration discussed earlier, this leads to inefficient trips. These inefficient trips in turn lead to more trips. In order to solve the action problem, the high transportation costs, the number of trips should be lowered. This can be achieved via more efficient routing. By including a collaborative optimization, routing can become more efficient for each of the local food banks. A big influence in this is including optimization in the routing schedule. Another possibility is that more food can be transported and utilized by the different food banks. This could allow for more variability in the types of product, or help more people in need.

1.3 Research goal

The objective of this research is to solve the core problem, the lack of optimization, as given in the problem statement (Section 1.2). The focus for achieving this is by solving the underlying Vehicle Routing Problem (VRP). It is important to ensure that all rules and regulations regarding transport of food and limitations due to specific drivers licenses are taken into consideration, as well as the upstream supply consisting of both food and packaging materials. The main goal is to reduce the transportation costs for the food banks. It is aimed to do this via a model that solves the VRP in such a manner that the given characteristics are considered. Preferably this solution outperforms the current methods for each of the food banks in the region of Twente-Salland. Moreover, a consistent VRP can help with an increase in demand, which will need to be transported too.

1.4 Research questions

With the research goal as determined in Section 1.3, the main research question is formulated as follows:

How can the transportation within the network of food banks in the region Twente-Salland be optimized?

Several subquestions have been defined in order to find an answer to the main research question in a systematical manner.

It is important to know and analyse the current situation. Without this information the transportation planning of the food bank cannot be optimized. Information needed are the different characteristics of the current transportation model in the region Twente-Salland, the parties involved, and its current limitations. Besides, the current transportation costs and the Key Performance Indicator (KPI) for the food banks have to be determined. This leads to the following research questions:

1. How is the transportation in the region Twente-Salland currently planned?

1.1. What are the characteristics of the transportation used in the region of Twente-Salland?
1.2. Which parties are involved in transportation in the region of Twente-Salland?
1.3. What are the limitations of the current planning process?
1.4. What are the current transport costs for the food bank?
1.5. What are the KPIs for the food banks in the region of Twente-Salland?

It is important to make use of existing models and theories. Literature plays an important role in this. Therefore a literature review is performed in order to answer the following questions:

2. What does the literature say about VRPs?
   2.1. What types of VRP align with the characteristics of the food banks and RDC in the region of Twente-Salland?
   2.2. What methods are there to solve the VRP?
3. What does literature mention about food bank supply chains?
   3.1. Does literature have solutions for the problems faced by the food bank?

Once it is known what methods are currently used and which are suggested by literature, it is possible to define the solution approach. This includes the data, assumptions, and other information to be defined.

4. How should the solution approach be designed?
   4.1. What requirements does the solution approach have to adhere to?
   4.2. What assumptions have to be made in order to solve this VRP?

A solution can only be accepted or rejected once it is known how it compares to the current situation. It needs to be known whether the new solution is outperforming the current way of working. Therefore the following questions will have to be answered:

5. How does the solution approach compare to the current situation?
   5.1. Which are the different scenarios under which we test the solution approach?
   5.2. How does the solution approach perform under the different scenarios considered?

Finally, conclusions must be drawn from the analysed results.

6. What are the recommendations for the food bank from the results of the experiments?

1.5 Research design

The answer to the main research question can be found when we systematically find the answers to the other questions. It is for that reason that this research consists of several phases. Each phase will find the answer to one of the research questions. In Figure 4, an overview of the relations between all sub questions is given. This figure also portrays what information is needed in order to do so.

The first phase of problem identification and analysis is covered in Chapters 2 and 3, where the context analysis and literature discussed respectively. This is followed by the solution approach in Chapter 4. The evaluation of the solution approach takes place in Chapter 5. Lastly, the overall conclusion, including recommendations and suggestions for further research, follows in Chapter 6.

Figure 4. Overview of the different phases and research questions, and how they relate to each other. It also shows what input is needed and what output is generated with each question.

2 Context analysis

In this chapter the following topics are discussed. First, Section 2.1 discusses food insecurity. This is followed by the characteristics of the current situation at the food banks in Section 2.2. Section 2.3 then explains the pilot to be carried out after previous research. Lastly, Section 2.4 gives the Key Performance Indicators (KPIs).

2.1 Food insecurity

Food insecurity can be defined as the lack of availability of nutritionally adequate and safe foods (Campbell, 1991). In the Netherlands, research found a food insecurity prevalence in 72.9% of food bank recipients (Neter, Dijkstra, Visser, & Brouwer, 2014). A 9.5% of the participants in that study reported to not eating while being hungry, as they could not afford food. This means that even in a wealthy country like the Netherlands food insecurity is present, also in extreme forms. Food insecurity leads to several social implications, such as: impaired learning for children and adults, loss of productivity, increased need for health care and erosion of transfer of knowledge and practices to the next generation (Hamelin, Habicht, & Beaudry, 1999). This means that a food insecurity has consequences for the entire society.

Food banks play an important role in the immediate provision of food. Although they cannot solve food insecurity, they do have the potential to improve food security outcomes when, amongst others, operation resources are adequate (Bazerghi, McKay, & Dunn, 2016). Therefore, resources needed should be optimized with the transportation planning of the food bank.

2.2 Characteristics

2.2.1 Demand

The food bank in Almelo visits the RDC two to three times per week. The day of pick-up is related to the availability of drivers at each of the local food banks. Per trip the supply differs between 1 and 7 pallets. As the vehicles used by the food bank Almelo can transport at most 4 pallets at the time, every trip with more than 4 pallets actually means that multiple vehicles took the trip. Other food banks, like the one in Hellendoorn, only make one trip per week, while, e.g., the food bank in Enschede transports a bigger quantity, and therefore likely takes more trips. The average number of pallets a food bank gets is given in Table 2. These numbers have been calculated by finding the total amount of pallets transported on a particular weekday, and dividing this by 47. It is divided by 47 as the data contains 47 weeks. It happened that a food bank would visit the RDC on, e.g., Tuesday in only 10 weeks. Thus, the average is lower than the average amount of pallets actually transported in that situation. However, it is not a regular visit, which had to be accounted for. Values have been rounded to the nearest integer value, as it is only possible to transport full pallets. Besides, the amount going to the food bank in Deventer is unknown. As the food bank and RDC are in the same location, there is no transport between the two, and therefore no record of transported pallets.

Table 2. The demand, in number of pallets, for each of the food banks listed per day.

Location	Tuesday	Wednesday	Thursday	Friday
Deventer	unknown	unknown	unknown	unknown
Almelo	2	3	2	-
Enschede	-	7	1	8
Hellendoorn	-	-	2	-
Losser	-	-	2	-
Midden-Twente	-	7	-	5
Oost-Twente	-	3	-	3
Raalte	-	-	5	-
Rijssen	-	1	2	-
Vaassen	-	-	3	-
Zutphen	-	7	3	-

2.2.2 Supply

There are two types of supply from a local food bank to the RDC. The most common is the return of packaging. This happens regularly, as the packaging is needed to receive food as well. Occasionally there is some supply of actual food. The latter only happens if the food banks receives large quantities of the same product and these quantities are too much to distribute amongst only their own clients. Although we know the amount of pallets being returned to the RDC, it is unknown whether it was food or a return of packaging. Therefore an assumption has to be made. Table 3 shows the expected supply from the local food bank to the RDC.

Table 3. The number of pallets given to the RDC per food bank. These numbers are based on the expertise of the volunteers at the RDC.

Location	Supply
Deventer	none
Almelo	2 pallets per month
Enschede	2 pallets per week
Hellendoorn	none
Losser	none
Midden-Twente	none
Oost-Twente	none
Raalte	none
Rijssen	none
Vaassen	10 to 15 pallets per year
Zutphen	3 pallets per month

2.2.3 Locations

A map of the different local food banks in the region of Twente-Salland is provided in Figure 5. The food bank and RDC in Deventer are located at the same location, hence why only the RDC is pictured. Excluding the food bank in Deventer, as no transportation is needed there, the distance from a food bank to the RDC is between 13 and 65 kilometers. The full overview of distances between different food banks and the RDC is given in Table 4.

Figure 5. Map of the food banks in the region Twente-Salland.

Table 4. Distance matrix from the RDC to the local food banks in kilometers. This also includes the distances between the different food banks. These distances were found using Google Maps.

	Deventer (RDC)	Almelo	Enschede	Hellendoorn	Losser	Midden-Twente	Oost-Twente	Raalte	Rijssen	Vaassen	Zutphen
Deventer (RDC)	-	46,7	59,9	31,6	65,0	47,8	58,3	19,9	25,2	18,8	12,7
Almelo	45,4	-	28,3	18,7	31,6	13,9	24,7	30,0	16,6	67,9	33,8
Enschede	59,8	26,1	-	39,3	10,7	9,7	9,9	51,4	37,4	82,2	41,8
Hellendoorn	31,7	19,5	41,5	-	46,6	19,4	39,9	12,4	9,7	57,0	30,3
Losser	64,8	30,8	11,2	44,3	-	18,3	8,3	56,4	43,2	87,3	52,1
Midden-Twente	48,6	13,9	9,0	28,1	19,1	-	12,1	40,2	26,2	71,1	31,0
Oost-Twente	58,1	24,0	10,3	37,6	8,3	11,5	-	49,7	35,7	80,5	45,3
Raalte	20,0	30,6	51,5	12,4	56,6	39,5	50,0	-	21,5	37,2	31,6
Rijssen	25,2	18,1	37,2	10,0	42,3	25,2	35,6	26,0	-	52,4	23,7
Vaassen	18,8	69,9	82,8	48,9	92,3	71,9	81,2	37,3	52,9	-	28,7
Zutphen	12,8	33,3	41,7	30,2	51,3	31,0	44,7	33,6	23,8	28,2	-

2.2.4 Vehicles

The RDC does not have transportation of their own. In order to receive their supply they make use of external organizations. These are either the supplier of the food, or a local transportation business. The local food banks do have vans that are used to transport supply and packaging materials. Each food bank has different vehicles, meaning that they can transport a different amount of supply. The type of vehicles, and specifically how many pallets they can transport, form a limitation. This is summarised in Table 5.

Table 5. The number of vehicles and the number of pallets a food bank can transport at once. If two different amounts of pallets are given, it is dependent on the vehicle, and refers to the vehicle in the same order as the number of vehicles is given.

Location	Number of vehicles	Number of pallets (per vehicle)
Deventer	1	unknown
Almelo	3	3 or 4
Enschede	2	4
Hellendoorn	1	3
Losser	1	3
Midden-Twente	2 (1 van and a trailer)	3 or 4
Oost-Twente	1	4
Raalte	2	6 or 3
Rijssen	1	4
Vaassen	unknown	10
Zutphen	1	4

2.2.5 Depots

The situation at the food banks is different from common situations discussed in a VRP. In most VRPs the depot, with the supply, is also responsible for the distribution of their product. These products are delivered from the depot to the customers and vehicles used return back to the depot. In the case of the food banks in the region Twente-Salland, the local food banks go to the RDC to pick up supply. The local food banks are thus the ones responsible for the transportation. As explained, they are also the ones that are responsible for the vehicles. This results in a situation where each of the food banks is a depot, and the RDC can be seen as a hub. Each of the depots (food banks) has a request for pick up from or delivery to the hub (RDC).

2.2.6 VRP specific characteristics

Each of the previously discussed characteristics are relevant to the VRP. In order to calculate a VRP, the demand, supply, locations, and vehicles should be known. In this case there the demand and supply are assumed to be deterministic, while the locations are deterministic. There could be different scenarios with only some of the locations participating, but this does not change the distances between two locations. The vehicles are a heterogeneous fleet, as each food bank has different vehicles. Besides, these vehicles are different from one another, meaning that they can transport a different amount of pallets per trip. More interesting is the classification of customers and depots. This information is also needed for the VRP. As determined in the previous section, this situation deals with multiple depots and a single hub. Besides, as there is both demand and supply, this is a pickup and delivery problem. Due to all supply going towards the RDC and all demand coming from the RDC too there is a one-to-many and many-to-one relationship between the food banks and the RDC.

2.3 Pilot in Twente-Salland

A preliminary research about the logistics of the food bank was performed by Ipskamp (2021) and Rienstra (2021). As there was not yet anything known, Rienstra investigated whether centralizing the logistics of the food bank was advantageous, while Ipskamp looked into optimizing a centralized logistics system. They solved this with the use of a VRP. Ipskamp also looked into different scenario's such as an increase in demand, not have every food bank participate, and the use of sub hubs. It was concluded that the food bank could save up to 50% of the variable transport costs they currently face. Therefore it was advised to deliver supply from the RDC to the food banks, instead of having each food bank pick up supply individually. However this was to be done by multiple trucks, instead of the current vehicles used by the different food banks. Altogether their researches have resulted in a pilot. During this pilot there will be multiple trucks that follow a route from the RDC to the different food banks in the region. This pilots aims to decrease the amount of individual trips to the RDC. However, this pilot includes only the regular trips to the RDC. All other trips are not included. This means that only the downstream allocation of food is looked after.

In order to perform this pilot, Voedselbanken Nederland will support the food banks in the region of Twente-Salland. This means that the requirements of multiple trucks and a pool of drivers that are allowed to drive those trucks, as suggested by the previous research, are looked after.

Moreover, the RDC in Deventer does note some challenges that should be taken into consideration:

Retour of packaging
Packaging is circulated between suppliers, the RDC and local food banks, which has monetary value. The local food banks now bring back pallets and crates when they come by the RDC to pick up supply. As these trips no longer happen, another way of returning the packaging must be considered.

Cost division
A question that was raised is on how to divide the costs of this new transportation method among the different food banks and the RDC. The RDC finds it important that there is a fair division of costs. Especially since they do not have any method of transport in the current situation, costs for transport would significantly rise for them, while for the local food banks the costs should lower. However, due to the circulation of financial means, it is likely the costs will still fully fall on the local food banks.

Different food groups
As mentioned in Section 1.2.1, different food types cannot be transported together. The trucks should have different compartments at different temperatures in order to deliver everything to a single food bank at the same time. Another option is to have a truck transport a different food group, but this would likely result in split deliveries. As the food banks depend on volunteers, it would be preferable to have all products delivered at the same time.

Unknown time of supply
The RDC does not know beforehand at what time supply will be delivered. As this supply still has to be divided before it can be send to the different food banks, it cannot be said at what time a truck could leave the RDC. The communication between RDC and local food bank has to be well-maintained to prevent local food banks from waiting for supply that will not come.

Storing goods at local food banks
Not every food bank has as much storage space as the RDC in Deventer has. It may be the case that the local food banks cannot store all the supply. This means that the day of delivery is important for these food banks. Therefore it can be the case that not all food banks can get deliveries on the same day.

Unloading of goods at local food banks
Like the possible lack of storage at local food banks, they may also be located at a location where trucks cannot unload the goods. In case it is not possible to deliver to a food bank with a truck, this would mean that those food banks will have to continue the current method of picking up the supply themselves.

2.3.1 Difference in characteristics related to the VRP

Some of the characteristics between the current situation and pilot are different. The demand, supply, locations, and KPIs will remain the same, but the vehicles and depots are different. While the current situation uses different vehicles, all belonging to one of the local food banks, the pilot will run with 2 or 3 trucks that start and end at the RDC in Deventer. There will also be drivers that are allowed to drive these vehicles. Although there will still be a limit on both weight and volume, it will not be nearly as limiting as the current situation. This also means that the RDC is the only depot, and all food banks are customers.

2.3.2 Changes in pilot

In the duration of this research the pilot has changed, as COVID-19 no longer created limitations for the food banks. The general idea is to allow customers to pick up their package during multiple days of the week. This means that food is needed multiple days a week, in smaller quantities. Instead of using big trucks they want to use smaller vehicles, like the vehicles already being used, possibly in combination with a trailer. Instead of each of the food banks going to the RDC in Deventer, only the four biggest food banks will go there. They will be paired to four smaller food banks, who will pick up their demand at those. The bigger food banks thus become a hub. As the food bank in Deventer is close to the RDC, they do not need the transportation. The other two food banks, Vaassen and Zutphen, already decided they do not want to take part in this pilot. For the VRP this means that instead of one hub (the RDC) and multiple depots, there now are multiple hubs and depots. This also changes the problem into a set of one-to-one problems. This is because each food bank only receives/picks up demand from/at one other location. The same applies for the supply towards the RDC.

2.4 Key Performance Indicators

The main KPI is the costs associated with the transportation of supply. At this moment, the food bank in Almelo spends a third of their annual budget on transportation. Half of this comes from the trips to and from the RDC in Deventer. This adds up to roughly €12.000 per year. For the RDC in Deventer the costs for transportation are estimated to be €10.750 in the year 2022. The high costs for the RDC may seem surprising as they do not have their own transportation. However, these costs come from the external transportation required for picking up some of the supply. For an organization like the food bank, that only relies on donations from local entrepreneurs and companies, this adds up to a large sum. It shows the importance of improving the transportation, in order to lower the costs. The model will therefore have to have an objective function that minimizes the costs. For this, there need to be assumptions regarding the cost of fuel and the material used.

A second KPI would be the distance driven to get the products from the RDC to the local food banks. By decreasing food surpluses and thus lowering the burden on the environment they are already helping to reach a more sustainable supply chain within the food industry. By decreasing the distance driven, the burden on the environment is lessened too. Although the distance driven is part of the calculations for the cost, it can also be considered as a second KPI.

Thirdly, the food banks fully operates on volunteers. It is not always easy to find people to work on a voluntary basis, or to change the moment of working for existing volunteers. Therefore, another KPI would be the number of vehicles used, as these determine the number of volunteers. Like the distance, this is already part of the calculations for the costs, but can be considered seperately too.

2.5 Conclusion

The chapter starts with an introduction to food insecurity. This shows why there is a need for initiatives like the food bank in the Netherlands. This is followed by explaining how the food banks and their transport works. Although it is supply-driven, the transportation flows are described in a demand-driven manner for clarity. The local food banks have fixed routes for picking up supply from the local supermarkets and bakeries. Besides, there are regular trips to the RDC. Each of the local food banks takes care of this themselves, meaning that there is no centralized system in place. If there is other supply available the logistics coordinator of the local food bank will make an ad hoc decision to ensure it gets picked up. This means that the food banks both receive and pick up food without process of optimization within the current transportation planning. The limitations within the current planning process are therefore a lack of knowledge, as it is only known on short notice that there is supply, as well as a lack of collaboration as the different food banks do it all by themselves and not as a collective. From this follow the high costs, a third of the budget, associated with the transportation at the food bank Almelo. It is important to the food bank to lower these costs. Therefore, the costs are the main KPI in this research. The RDC also has high transportation costs, even though they do not have their own transport. These costs arise due to the hiring of external parties that pick up supply throughout the country for the RDC. Besides the costs, the distance driven and number of vehicles used can be considered as individual KPIs. All together this chapter answers the research question *"How is the transportation in the region Twente-Salland currently planned?"*, as well as its sub questions.

3 Literature review

This chapter starts with an introduction to the regular Vehicle Routing Problem. Based on Section 2.2.6, it is important to also look at VRPs with specific characteristics. Firstly, a VRP that includes backhauls, in order to deal with the supply from food bank to RDC, is discussed in Section 3.2. Secondly, to ensure that the deliveries take place during working hours of the food bank, the Vehicle Routing Problem with Time Windows (VRPTW) discussed in Section 3.3. Next, Section 3.4 explains the Multi-Depot Vehicle Routing Problem (MDVRP), which deals with a multi-depot situation. As the food bank deals with different types of products, the Vehicle Routing Problem with Multiple Commodities (MCVRP) is looked into. From there VRPs with a combination of characteristics will be discussed in Sections 3.6, 3.7 and 3.8. This is followed by discussing the different heuristics used for solving the VRP in Section 3.9. Section 3.10 discusses literature related to food banks. Finally the answers to the research questions concerning literature, are given in Section 3.11.

3.1 Vehicle Routing Problem

The VRP calls of the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers (Toth & Vigo, 2002). In 1959 Dantzig and Ramser were the first to introduce the VRP. Dantzig and Ramser named the VRP the "Truck Dispatching Problem" and said it could be considered as a generalization of the "Traveling-Salesman problem" (Dantzig & Ramser, 1959). According to Laporte (2007) the VRP consists of designing optimal delivery or collection routes from a central depot to a set of customers, subject to various constraints. Laporte also mentions that this is a problem thousands of distributors worldwide face on daily basis, and has a significant economic importance. Furthermore, the VRP is NP-hard. This means that it cannot be solved in polynomial time. So while Dantzig and Ramser were responsible for the first mathematical programming formulation, in 1964, Clarke and Wright proposed an effective greedy heuristic that improved the results compared to the approach used by Dantzig and Ramsler. Since then, there has been significant progress in the development of metaheuristics for the VRP (Laporte, 2007).

Since 1959, many different types of VRPs have been proposed in literature. Both Eksioglu, Vural, and Reisman (2009) as well as Lahyani, Khemakhem, and Semet (2015) performed a taxonomy review on the different characteristics of each VRP. The overview of Lahyani et al. (2015) is given in Figure 6. VRPs combine several of the characteristics to match the problem they are trying to solve.

1 Scenario characteristics

1.1 Input data
- 1.1.1 Static
- 1.1.2 Dynamic
- 1.1.3 Deterministic
- 1.1.4 Stochastic

1.2 Decision management components
- 1.2.1 Routing
- 1.2.2 Inventory and routing
- 1.2.3 Location and routing
- 1.2.4 Routing and driver scheduling
- 1.2.5 Production and distribution planning

1.3 Number of depots
- 1.3.1 Single
- 1.3.2 Multiple

1.4 Operation type
- 1.4.1 Pickup or delivery
- 1.4.2 Pickup and delivery
- 1.4.3 Backhauls
- 1.4.4 Dial-a-ride

1.5 Load splitting constraints
- 1.5.1 Splitting allowed
- 1.5.2 Splitting not allowed

1.6 Planning period
- 1.6.1 Single period
- 1.6.2 Multi-period

1.7 Multiple use of vehicles
- 1.7.1 Single trip
- 1.7.2 Multi-trip

2 Problem physical characteristics

2.1 Vehicles
- 2.1.1 Type
  - 2.1.1.1 Homogeneous
  - 2.1.1.2 Heterogeneous
- 2.1.2 Number
  - 2.1.2.1 Fixed
  - 2.1.2.2 Unlimited
- 2.1.3 Structure
  - 2.1.3.1 Compartmentalized
  - 2.1.3.2 Not compartmentalized
- 2.1.4 Capacity constraints
- 2.1.5 Loading Policy
  - 2.1.5.1 Chronological order
  - 2.1.5.2 No policy
- 2.1.6 Drivers regulations

2.2 Time constraints
- 2.2.1 Restriction on customer
- 2.2.2 Restriction on road access
- 2.2.3 Restriction on depot
- 2.2.4 Service time
- 2.2.5 Waiting time

2.3 Time window structure
- 2.3.1 Single time window
- 2.3.2 Multiple time windows

2.4 Incompatibility constraints

2.5 Specific constraints

2.6 Objective function
- 2.6.1 Single objective
- 2.6.2 Multiple objectives

Figure 6. Taxonomy of the VRP literature (Lahyani et al., 2015). The characteristics highlighted in yellow are relevant for this research.

The pickup and delivery VRP takes into consideration both the downstream and upstream flow of goods. Each vehicle depart from and return to a central depot, while each transportation request contains a single destination and single origin location (Savelsbergh & M.Sol, 1995). Moreover, different products may be incompatible, meaning they cannot be transported together into the same vehicle. However, vehicles can transport different products in different trips (Duk Song & Dae Ko, 2016). This is covered by the MCVRP.

The VRPTW is a generalization of the VRP involving the added complexity of allowable delivery times, or time windows (Desrochers, Desrosiers, & Solomon, 1992). Time windows are called soft when they can be considered non-biding, and hard when they cannot be violated (Kallehauge, Larsen, Madsen, & Solomon, 2005). When these time windows are soft the problem relaxes to a VRP.

Although not present in the taxonomy by Lahyani et al., there are VRPs that make use of hubs. A hub is a consolidation center that bundles quantities between depots to achieve economies of scale for depot-to-depot transports (Wasner & Zäpfel, 2004). Hubs also collect, sort and consolidate the freight from many origins, then ship it to the destinations or transfer it to other hubs (Yang, Bian, Bostel, & Dejax, 2019). Thus, with the use of hubs the linehaul structure can be further optimized.

Another characteristic not present in this taxonomy is the distinguishing between one-to-one, one-to-many-to-one and many-to-many problems. A one-to-one problem occurs when each product has one origin and one destination between which it must be transported. The one-to-many-to-one category has certain products being delivered from a depot to many customers and other products to collect from customers and transport to the depot. Finally, the many-to-many problem has one or m-commodities collected from many collection sites to be transported to many destination places (Euchi, 2020).

Next the mathematical model of the general VRP is given. In each of the sections dealing with a specific characteristic the changes in this model are provided in their respective section. This is done for the Vehicle Routing Problem with Backhauls (VRPB), VRPTW, MDVRP, and MCVRP. These characteristics are highlighted as they are part of the underlying problem the food bank aims to solve. After the individual characteristics, two VRPs which combine relevant characteristics are addressed. These form the basis for defining the VRP for the transportation problem the food banks face.

3.1.1 Mathematical model

Christofides, Mingozzi, and Toth (1981) provided the problem formulation of a VRP as an integer program. This formulation is provided by means of Equations (3.1) to (3.8) [6].

Parameters

$\mathcal{N}$	Set of customers, indexed by i and j,
$\mathcal{V}$	Set of vehicles, indexed by k,
Q	Capacity of a vehicle,
q_i	Demand by customer i,
c_{ij}	Cost of travelling between customer i and customer j,
y_i	Dummy variable.

Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ visits customer } j \text{ immediately after visiting customer } i \\ 0 & \text{otherwise} \end{cases} .$$

Model

$$\text{min z} = \sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} (c_{ij} \sum_{k=1}^{\mathcal{V}} x_{ijk}), \tag{3.1}$$

$$\text{s.t.} \quad \sum_{i=0}^{\mathcal{N}} \sum_{k=1}^{\mathcal{V}} x_{ijk} = 1 \qquad j = 1, .., \mathcal{N}, \tag{3.2}$$

$$\sum_{i=0}^{\mathcal{N}} x_{ipk} - \sum_{j=0}^{\mathcal{N}} x_{pjk} = 0 \qquad k = 1, .., \mathcal{V}, p = 0, .., \mathcal{N}, \tag{3.3}$$

[6] Variable names and notation may be changed to ensure consistency throughout the report

$$\sum_{i=1}^{\mathcal{N}}(q_i \sum_{j=0}^{\mathcal{N}} x_{ijk}) \leq Q_k \qquad k = 1, .., \mathcal{V}, \tag{3.4}$$

$$\sum_{k=0}^{\mathcal{V}} \sum_{j=0}^{\mathcal{N}} x_{0jk} \leq |V| \qquad k = 0, .., \mathcal{V}, j = 0, .., \mathcal{N}, \tag{3.5}$$

$$\sum_{j=1}^{\mathcal{N}} x_{0jk} = 1 \qquad k = 1, .., \mathcal{V}, \tag{3.6}$$

$$y_i - y_j + \mathcal{N} \sum_{k=1}^{\mathcal{V}} x_{ijk} \leq \mathcal{N} - 1 \qquad i \neq j = 1, .., \mathcal{N}, \tag{3.7}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j, k, \tag{3.8}$$

The objective function (3.1) is about minimizing the costs involved. Constraints (3.2) and (3.3) state that each customer must be visited, and when visited the vehicle must leave this customer again. Constraint (3.6) states that each vehicle may only be used once, while Constraint (3.7) is the sub-tour elimination, which also forces each route to pass through the depot. Constraint (3.8) are the integrality constraints. Finally, as this model includes capacity constraint, Constraint (3.4), it may also be seen as a Capacitated Vehicle Routing Problem (CVRP). This means that there is limited capacity for each vehicle. The model also includes (3.5) which gives an upper bound to the number of vehicles that can be used.

3.2 VRP with Backhauls

The VRPB is a pickup/delivery problem where on each route all deliveries must be made before any pickups (Goetschalkx & Jacobs-Blecha, 1989). Thus, constraints are added to related to the order in which deliveries and pickups take place. Each delivery must be made before any pickups can be made. This is because rearrangement of the loads on trucks is not deemed feasible. This type of VRP has two type of customers, namely the backhaul (pickup) and linehaul (delivery) customers (Yazgı Tütüncü, Carreto, & Baker, 2009). There are variants where the constraints of deliveries taking place before pickups is relaxed, and deliveries are allowed to take place after pick ups, this is called the Mixed VRPB (Yazgı Tütüncü et al., 2009).

3.2.1 Mathematical model

The model as given here is based on the model provided by Goetschalkx and Jacobs-Blecha (1989). In this model a vehicle can only visit linehaul or backhaul customers, but not both. Only parameters and decision variables that have not yet been specified in Section 3.1 have been given.

Parameters

$\mathcal{B}$	Number of backhaul customers, indexed by $N + 1, N + 2, ..., N + B$,
a_i	Demand of linehaul customer i,

b_i Supply of backhaul customer i.

Decision variables

$$u_{ik} = \begin{cases} 1 & \text{if linehaul customer } i \text{ is serviced by vehicle } k,\ i = 0, .., \mathcal{N} \\ 0 & \text{otherwise} \end{cases}.$$

$$v_{jk} = \begin{cases} 1 & \text{if backhaul customer } j \text{ is serviced by vehicle } k,\ i = 0, \mathcal{N}+1, .., \mathcal{B} \\ 0 & \text{otherwise} \end{cases}.$$

Model

$$\min z = \sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} (c_{ij} \sum_{k=1}^{\mathcal{V}} x_{ijk}), \tag{3.9}$$

$$\text{s.t.} \quad \sum_{k=1}^{\mathcal{N}} a_i u_{ik} \leq Q \qquad k = 1, .., \mathcal{V}, \tag{3.10}$$

$$\sum_{i=\mathcal{N}+1}^{\mathcal{B}} b_i v_{ik} \leq Q \qquad k = 1, .., \mathcal{V}, \tag{3.11}$$

$$\sum_{k=1}^{\mathcal{V}} u_{ik} = 1 \qquad i = 1, .., \mathcal{N}, \tag{3.12}$$

$$\sum_{k=1}^{\mathcal{V}} v_{ik} = 1 \qquad i = \mathcal{N}+1, .., \mathcal{B}, \tag{3.13}$$

$$u_{0k} = 1 \qquad k = 1, .., \mathcal{V}, \tag{3.14}$$

$$v_{0k} = 1 \qquad k = 1, .., \mathcal{V}, \tag{3.15}$$

$$\sum_{i=0}^{\mathcal{N}} \sum_{j=\mathcal{N}+1}^{\mathcal{B}} x_{ijk} = 1 \qquad k = 1, .., \mathcal{V}, \tag{3.16}$$

$$\sum_{i=0}^{\mathcal{B}} x_{ijk} = \begin{cases} u_{jk} & \text{if } j = 1, ..., \mathcal{N}, \\ v_{jk} & \text{if } j = \mathcal{N}+1, ..., \mathcal{B} \text{ and } j = 0, \\ & \quad k = 1, ..., \mathcal{V}, \end{cases} \tag{3.17}$$

$$\sum_{j=0}^{\mathcal{B}} x_{ijk} = \begin{cases} u_{jk} & \text{if } i = 0, ..., \mathcal{N}, \\ v_{jk} & \text{if } i = \mathcal{N}+1, ..., \mathcal{B}, \\ & \quad k = 1, ..., \mathcal{V}. \end{cases} \tag{3.18}$$

The objective function (3.9) is aimed at minimizing the costs. Constraints (3.10) and (3.11) are similar to Constraint (3.4). The difference is that it is specified for both the linehaul and backhaul trucks that they cannot be overloaded. These constraints are similar to Constraint (3.4). Constraints (3.12) and (3.14), as well as Constraints (3.13) and (3.15) indicate that only one vehicle can be assigned to each linehaul and backhaul route respectively. Constraints (3.17) and (3.18) state that exactly one vehicle enter each customer and depot once, and that this vehicle must leave these sites. These differ from Constraint (3.2) as given in Section 3.1, as there is a clear distinction made between the linehaul and backhaul vehicles. Constraint (3.2) remains in place, albeit that the summation is changed from j = 0, .., $\mathcal{N}$ to j = $\mathcal{N}+1, .., N+\mathcal{B}$. Finally, Constraint (3.16) says that there must be exactly one link traveled by each vehicle from linehaul to backhaul on each route. From Section (3.1), the usual subtour elimination constraint, Constraint (3.7), is also used.

3.3 VRP with Time Windows

The VRPTW is a generalization of the VRP involving the added complexity of allowable delivery times, or time windows (Desrochers et al., 1992). Time windows are called *soft* when they can be considered non-biding, and *hard* when they cannot be violated (Kallehauge et al., 2005). When these time windows are soft the problem relaxes to a VRP. In the case of soft time windows penalties will be involved to compensate for breaking the time windows.

3.3.1 Mathematical Model

The constraints added and changed from the model provided in Section 3.1 are given below. The research by Kallehauge et al. (2005) provides the basis for these constraints[7]. Only parameters and decision variables not yet defined in earlier sections have been defined.

Parameters

a_i	Customer i is not available before this time,
b_i	A vehicle must arrive at customer i before this time,
t_{ij}	Time to travel from customer i to j, this time may include service time at customer i.

Decision variables

s_{ik}	The time vehicle k starts to service customer i .

Constraints

$$\sum_{i=0}^{\mathcal{N}} x_{i,n+1,k} = 1 \qquad k = 0, .., \mathcal{V}, \tag{3.19}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \qquad i, j = 0, .., \mathcal{N}, k = 0, .., \mathcal{V}, \tag{3.20}$$

$$a_i \leq s_{ik} \leq b_i \qquad i = 0, .., \mathcal{N}, k = 0, .., \mathcal{V}. \tag{3.21}$$

Constraints (3.7) is not included in this model. All other constraints from the model in Section 3.1 are used, as well as the objective function. Constraints (3.19) are used to indicate that a vehicle must return to the depot. Constraints (3.20) establish the relationship

[7] Variable names and notation may be changed to align with the used notation at the University of Twente and thus within this report.

between the vehicle departure time from a customer and its immediate successor. Lastly, Constraints (3.21) defines the time windows.

Non-linear constraint (3.20) can be linearized to

$$s_{ik} + t_{ij} - M_{ij}(1 - x_{ijk}) \leq s_{jk} \qquad i, j = 1, .., \mathcal{N}, k = 1, .., \mathcal{V}, \tag{3.22}$$

allowing the equation to be used in a linear program.

3.4 Multi-depot VRP

In the MDVRP, more than one depot is considered. Customers are to be served by one of these depots. As with the VRP, a vehicle must leave and return to the same depot. The MDVRP has two stages: first, the customers must be allocated to depots; second, the routes must be built. Ideally, these steps are done simultaneously (Tansini, Urquhart, & Viera, 2001). At present, this MDVRP is given more attention as it offers a more realistic scenario (Jayarathna, Lanel, & Juman, 2020). Like the regular VRP, the MDVRP has been extended by time windows, pick up and delivery, and a heterogeneous fleet. Crevier, Cordeau, and Laporte (2007) studied a MDVRP, where the depots can act as intermediate replenishment facilities along the route of a vehicle. This is also known as the Multi-Depot Vehicle Routing Problem with Inter-Depot Routes.

3.4.1 Mathematical model

Based on the model by Ramos, Gomes, and Póvoa (2020)[8], the model as given in Section 3.1.1 needs a few constraint changes, as well as an added parameter for modelling the MDVRP.

Parameters

$\mathcal{M}$	Set of depots	
$\mathcal{Z}$	Set of nodes, $\mathcal{Z} = \mathcal{N} \cup \mathcal{M}$	
$\mathcal{V}_i$	Subset of vehicles belonging to depot i	(3.23)

Constraints

$$\text{min z} = \sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{\mathcal{Z}} (c_{ij} \sum_{k=1}^{\mathcal{V}} x_{ijk}), \tag{3.24}$$

$$\text{s.t.} \quad \sum_{i=0}^{\mathcal{Z}} \sum_{k=1}^{\mathcal{V}} x_{ijk} = 1 \qquad j = 1, .., \mathcal{N}, \tag{3.25}$$

$$\sum_{j=0}^{\mathcal{Z}} \sum_{k=1}^{\mathcal{V}} x_{ijk} = 1 \qquad i = 1, .., \mathcal{N}, \tag{3.26}$$

[8] The model was partially changed, parameter and variable names, as well as notation, to ensure consistency throughout the report.

$$\sum_{j=1}^{\mathcal{N}} x_{0jk} = 1 \qquad k = 1, .., \mathcal{V}, \tag{3.27}$$

$$\sum_{j=1}^{\mathcal{N}} x_{ijk} \leq 1 \qquad k \in \mathcal{V}_i, i = 1, .., \mathcal{M}, \tag{3.28}$$

$$\sum_{i=1}^{\mathcal{N}} x_{ijk} \leq 1 \qquad k \in \mathcal{V}_j, j = 1, .., \mathcal{M}, \tag{3.29}$$

$$\sum_{i=1}^{\mathcal{N}} x_{ijk} = 0 \qquad k \notin \mathcal{V}_j, j = 1, .., \mathcal{M}, \tag{3.30}$$

$$\sum_{j=1}^{\mathcal{N}} x_{ijk} = 0 \qquad k \notin \mathcal{V}_i, i = 1, .., \mathcal{M}, \tag{3.31}$$

$$\tag{3.32}$$

Constraints (3.28) and (3.29) ensure that each vehicle will leave and return to its home depot at most once. Constraints (3.30) and (3.31) jointly ensure that a vehicle cannot leave and return to a depot other than its home depot. The constraints (3.3), (3.4), (3.5), (3.7), (3.8) from Section 3.1.1 remain in the model.

3.5 Multi-commodity VRP

So far, the VRPs have assumed that all products can be transported together. However, different commodities may be incompatible, meaning they cannot be transported together into the same vehicle. However, vehicles can transport different commodities in different trips. Duk Song and Dae Ko conducted a research of a VRP with both refrigerated- and general-type vehicles for multi-commodity perishable food products delivery (Duk Song & Dae Ko, 2016). Before that Zhang et al. presented an optimization of the structure of the distribution of chilled and frozen food with the use of a tabu search algorithm (Zhang, Habenicht, & Spieß, 2003).

Multi-compartment
Multi-commodity VRPs, like the one from Duk Song and Dae Ko, tend to make use of different vehicles with different characteristics. It is also possible to make use of vehicles that can be divided into different compartments, each compartment having their own characteristic. The most difficult in this is loading the vehicles, rather than planning the route (Fallahi, Prins, & Wolfler Calvo, 2008).

3.5.1 Mathematical model

The new and/or changed variables, decision variables and constraints as given here come from the model as given by Fallahi et al. (2008)[9]. Like Sections 3.2 and 3.3, only the parameters and decision variables that had not yet been defined in earlier sections are given.

[9]Notation and variable names may be changed to ensure consistency throughout this report

Parameters

$\mathcal{P}$	Set of products, indexed by p
$\mathcal{Q}_p$	The demand of product type p

Decision variables

$$y_{jkp} = \begin{cases} 1 & \text{if customer } j \text{ receives product } p \text{ from vehicle } k \\ 0 & \text{otherwise} \end{cases} .$$

Constraints

$$y_{jkp} \leq \sum_{i=0}^{\mathcal{N}} x_{ijk} \qquad j = 0,..,\mathcal{N}, k = 0,..,\mathcal{V}, p = 0,..,\mathcal{P}, \tag{3.33}$$

$$\sum_{k=0}^{\mathcal{V}} y_{jkp} = 1 \qquad j = 0,..,\mathcal{N}, p = 0,..,\mathcal{P}, \tag{3.34}$$

$$\sum_{j=0}^{\mathcal{N}} y_{jkp} q_{jp} \leq Q_p \qquad k = 0,..,\mathcal{V}, p = 0,..,\mathcal{P}, \tag{3.35}$$

$$y_{jkp} \in \{0,1\} \qquad \forall j, k, p, q_{jp} \neq 0. \tag{3.36}$$

Constraints (3.33) to (3.36) differ from their respective constraints in the model as given in Section 3.1 by adding the index of the different products and looking at the capacity per compartment instead of the full compartment. This means that there is a binary value for each commodity and not the total demand requested, as well as the capacity is checked per commodity and not the full demand of a food bank. This can be done per vehicle, as opposed to compartment, as well, which will be the case for the food banks.

3.6 Multi-commodity, multi-depot, pick up and delivery with time windows VRP

Aksoy and Kapanoglu (2012) combined the multi-commodity, multi-depot, heterogeneous vehicle, pickup and delivery characteristics to deal with the problem of air transportation in the Turkish Air Force. They aim to get rid of fixed routes and find routes that satisfy the demand but minimize the costs. There are three factories where parts are transferred to, in order to get repaired. Besides the airbases transport parts and personnel amongst each other. This makes their network a combination of many-to-many and one-to-one problems, resulting in a one-to-many and many-to-one problem. The aircrafts have their own home base where they have to return at the end of the route. They used a commercial solver to solve their linear model.

Ríos-Mercado, López-Pérez, and Castrillón-Escobar (2013) created a GRASP-based heuristic for the multi-depot multi-commodity pickup and delivery problem with time windows and heterogeneous fleet for a bottled beverage distribution company. There was a need to distribute its products across several distribution centers, however the distribution centers were also able to request withdrawal and relocating of products. This makes it a many-to-many problem. They did also let the model make the decision to haul one or two trailers, and whether to put the product on the bottom or the shelf within the vehicle and/or trailer. The model was solved with GRASP. This is a multi-start metaheuristic that relies on greedy randomized constructions and local search.

Rieck, Ehrenberg, and Zimmermann (2014) dealt with a many-to-many problem, where vehicles perform pick up and delivery routes visiting customers only once. These routes started in an established hub. They included inter-hub routes, which are routes that go form one hub to another and back in order to obtain synergy effects by consolidating products in full truckloads. Included in their research were multiple products or commodities. This combines the multi-commodity, multi-depot, pick up and delivery problem with the a VRP that makes use of hubs. Small instances were solved using a commercial solver. However, a multi-start procedure based on a fix-and-optimize scheme and a genetic algorithm were also introduced. These different methods were analyzed. They concluded that an instance with up to 15 nodes can be solved to optimality in reasonable time (1 hour). The fix-and-optimize scheme as well as the genetic algorithm were able to generate promising solutions within an appropriate time limit.

3.7 Single-hub multi-depot VRP

The single-hub, multi-depot VRP combines different types of locations. Irnich (2000) solved a single hub, multi-depot problem where all requests have to be for pickup at or delivery to the hub. They also included time windows. Unlike general pickup and delivery problems, here all requests have to be either picked up at or delivered to the hub. All the pickup and delivery locations are assumed to be the depots of a given heterogeneous fleet of vehicles. In their case the routing information is not as important as the decision which requests are transported by the same vehicle. The problem is solved with set partitioning models. This implies a two-phase algorithm. They first enumerate all relevant route-vehicle combinations and then assign all relevant subsets of requests to the route-vehicle combinations.

Wasner and Zäpfel (2004) also made use of a hub and multiple depots when designing a transportation networks for parcel service providers. In their case the hub is a consolidation center that bundles quantities between depots to achieve economies of scale for depot-to-depot transports. This case is relatively small, as there is only one hub and about six depots. It was therefore possible to solve the problem with the optimization software CPLEX. Using more depot locations will require approximation procedures.

Kevin, Aritonang, and Lesmono (2019) created a model that assess the position of hubs by considering and aiming to reduce the total transport costs. This included a method of distribution of goods for hubs and non-hubs with third-party logistics determined by a VRP. First they determined where the hub needs to be located. With this information they used a VRP to calculate the cheapest routes. This two-model method was also used by Yang et al.

(2019). Their focus was specifically on long distance less-than-truckload freight transportation networks. Kevin et al. (2019) used LINGO software to solve the problem, while Yang et al. (2019) used CPLEX.

For Rafiei, Rabbani, Vafa-Arani, and Khoshnudi (2013) the situation included intermediate facilities and depots. With the use of these a heterogeneous fleet vehicle waste collection problem was formed. An extra addition to this problem was the use of different zones which added extra constraints. The waste was collected from bins throughout the city and once a vehicle was full the waste is deposited at one of the intermediary facilities. A mixed integer programming model is used to solve the problem.

There has also been a substantial amount of research on the multi-echelon routing problems. A multi-echelon routing problem makes use of one ore more intermediate facilities. In this type of problem, each echelon refers to one level of the distribution network (Cuda, Guastaroba, & Speranza, 2015). According to Ramirez-Villamil, Jaegler, and Montoya-Torres (2021) the first two-echelon VRP was presented by Crainic, Ricciardi, and Storchi (2004). For the city of Rome, they introduced a technological framework for the integrated management of urban freight transportation. They identified important planning and operation issues and models associated and described a formulation for one the problems. They made use of so-called satellites, which are the intermediate locations. For this they proposed a logistical structure and discussed algorithmic implementation issues. By doing so, they managed to solve the problem (location-allocation) using CPLEX.

Later, Crainic, Ricciardi, and Storchi (2009) addressed a two-echelon, multi-depot, heterogeneous vehicle routing problem with time windows. Further more this problem dealt with synchronization and scheduling on the short term. City logistics refers to the optimization of such advanced urban freight transportation systems. Thus, this problem is known as city logistics. Their study focused on the city logistics systems where consolidation and coordination of activities are performed at facilities organized into a hierarchical, two-tiered structure with major terminals sited at the city limits and satellite facilities strategically located close to or within the city center. A mathematical formulation was proposed, and they introduced variants which were analysed.

Belgin, Karaoglan, and Altiparmak (2018) solved the two-echelon vehicle routing problem with simultaneous pickup and delivery. However they made use of the same vehicles for both echelons. They solved it by combining a variable neighborhood descent and local search algorithm. Possible application areas for their research include multi-echelon distribution systems in logistics enterprises and express delivery service companies, multi-model freight transportation, grocery and hypermarkets products distribution and city logistics. Paul, Kumar, Rout, and Goswami (2021) solved a two-echelon pollution routing problem with simultaneous pickup and delivery under multiple time windows constraints. Their first echelon consists of depots and satellites, while the second consists of the satellites and customers. They have a separate fleet of vehicles for each of the echelons. The objective of their study was threefold; it investigated an integrated supply chain model of a single-vendor and multiple-buyers dealing with perishable items and imperfect production simultaneously.

Secondly, they constructed an optimal routing plan for the vehicles which entails minimum fuel consumption. Thirdly, various regulatory strategies are incorporated in the developed mathematical model within the bi-objective optimization framework. Altogether they developed a sustainable supply chain inventory management model, as they took inventory management into account. To solve the model they used the VNS algorithm. Finally, Zhou, Qin, Zhang, et al. (2022) solved the two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery. In this variant the same vehicles deliver and pick up in both echelons. Besides the second echelon customers have a specified time window. The problem was solved using the Tabu Search algorithm. Like Belgin et al. (2018) their research was generic, and can be used in different application areas. Both Paul et al. (2021) and Zhou et al. (2022) extended the problem by Belgin et al. (2018) by adding the time window constraints.

3.8 Multi-commodity VRP with time windows

The multi-commodity VRP with time windows is a combination of the multi-commodity VRP and the VRPTW.

According to Cattaruzza, Absi, Feillet, and Vigo (2014), the multi-commodity, multi-trip VRP with time windows calls for the determinations of a routing planning to serve a set of customers that require products belonging to incompatible commodities. They proposed an iterative local search for the problem introduced by Battarra, Monaci, and Vigo (2009). The problem faced is the distribution of merchandise to supermarkets. There is a presence of time windows and different commodities cannot be transported together. The aim is to minimize the number of vehicles used. In the variant of Cattaruzza, Absi, Feillet, and Vigo, the vehicles can transport different commodities in different trips. A customer is replicated as many times as the number of commodities the customer requires. This way, each customer is served once. They note that often a problem with multiple commodities is split into several single commodity problems. This was also done by Battarra et al.. The problems were then solved by using a routing heuristic. With the use of packing heuristic the multiple trips are aggregated.

Gjenstø and Vaksvik (2008) created a model for the one-to-many-to-one single vehicle, pickup and delivery problem with multiple commodities and visit windows. In this the visit windows are like the time windows as discussed in Section 3.3. The context of their research is the problems arising in the supply of offshore drilling platforms off the Norwegian coast using supply vessels. Besides, the platforms have commodities that need to be picked up and transported back to the depot. It was allowed to visit a customer twice. They used CPLEX to solve the mathematical models which were modeled in AMPL. Tabu Search was used to solve the problem.

Naccache, Côté, and Coelho (2018) investigate the multi-pickup and delivery problem with time windows. A client request is composed of several pickups of different items, followed by a single delivery at the client location. The problem was solved exactly with the use of branch-and-bound and heuristically developing a hybrid adaptive large neighborhood search with improvement operations.

Moshref-Javadi and Lee (2016) presented a customer-centric, multi-commodity vehicle routing problem with split deliveries. Because it is a customer-centric problem they minimized the latency, instead of the total travel distance which is minimized by server-oriented problems. In this problem two decisions are made. First, the routes are determined. Secondly, the quantities of the commodities to load and unload on the routes of the vehicles are determined. Assumed was that multiple vehicle distribute multiple types of commodities from a single depot.

3.9 Heuristics

According to Rieck et al. (2014) small instances of the VRP can be solved to optimality within a reasonable time. This is also true for the MDVRP. As most problems deal with larger instances, heuristics are suggested. They can be classified as simple construction and improvement procedures. More recently, metaheuristics like Tabu Search have been used (Crevier et al., 2007). The need for metaheuristics is also shown by Sharma and Saini (2020). Their review showed that almost all MDVRPs are solved with the use of a (meta)heuristic. Feo and Resende mention Simulated Annealing, Tabu Search, Genetic Algorithms and GRASP to be the most promising. Sharma and Saini (2020) too concluded that Genetic Methods, Tabu Search techniques, Simulated Annealing or VNS techniques, amongst others, determine the solution very efficiently. The effectiveness of heuristics does depend on their ability to adapt, avoid local optima and exploit the basic structure of the problem. Besides, restart procedures, controlled randomization, efficient data structures and preprocessing are beneficial (Feo & Resende, 1995).

Similarly, Çağrı Koç, Laporte, and İlknur Tükenmez (2020) reviewed the vehicle routing with simultaneous pickup and delivery. They divided the heuristics over construction and improvement heuristics, population search heuristics and ant colony heuristics. Each mentioned paper made use of different construction heuristics. The most mentioned population search methods are Tabu Search, Genetic Algorithms, GRASP and VNS. The ant colony optimization heuristics make use of different types of population search methods. Therefore, the following subsections will cover the following heuristics: Simulated Annealing, VNS, Tabu Search, Genetic Algorithms and GRASP.

3.9.1 Simulated Annealing

Simulated Annealing starts with a random initial solution. In each iteration, the algorithm takes a new solution form the predefined neighborhood of the current solution. The objective values are compared. If the new solution has a better value for the objective function, the new solution becomes the current solution from which the search continues. A worse new solution may also be accepted with a small probability. This prevents the algorithm from getting stuck in a local optimum. The probability of accepting a worse solution becomes less over time (Lin, Yu, & Chou, 2009).

3.9.2 Variable Neighborhood Search

VNS is a metaheuristic which exploits systematically the idea of neighborhood change. It does so both in the descent to local minima and in the escape from the valleys which contain them (Hansen & Mladenović, 2005). The algorithm for the basic VNS is given in Algorithm 1 and based on Hansen and Mladenović (2001).

Algorithm 1 Variable Neighborhood Search ($\mathcal{N}_k$, x, stopping condition)

Input $\mathcal{N}_k$, where $k = 1, .., k_{max}$, set of neighborhood structures that will be used in the search; x, initial solution; stopping condition
Output best solution found

1: **while** $k \leq k_{max}$ **do**
2: Shaking: Generate a point x' at random from the k^{th} neighborhood of x ($x' \in \mathcal{N}_k(x)$);
3: Local search: Apply some local search method with x' as initial solution; denote with x'' the so obtained local optimum;
4: **if** local optimum better than the incumbent **then**
5: Move there: $x = x''$
6: $\mathcal{N}_1$ (k = 1)
7: **else**
8: k = k + 1

3.9.3 Tabu Search

Duhamel, Potvin, and Rousseau (1997) proposed a Tabu Search for the VRP with both Backhauls and Time Windows. They defined the Tabu Search metaheuristic as follows: at each iteration, a neighborhood of the current solution is generated through different classes of transformations. The best solution in the neighborhood is selected as the new current solution, and the procedure is repeated. This new solution does not have to be better than the previous solution, which allows the method to escape from local optima and explore a larger space. To prevent cycling and thus reaching a local optimum, a tabu list stores the recent searches. Those solutions on the tabu list are not allowed to be chosen, even if they provide the best result. The algorithm for Tabu Search, as given by Duhamel et al. (1997) can be found in Algorithm 2.

3.9.4 Genetic Algorithm

Genetic Algorithms allow for a diversified exploration over the search space due to the management of several solutions at the same time. This is because an initial population of individuals (chromosomes) evolves through generations until satisfactory criteria of quality, a maximum number of iterations or time limits are reached. New individuals (children) are generated from individuals forming the current generation (parents) by means of genetic operators (crossover and mutation) (Cattaruzza, Absi, Feillet, & Vidal, 2014).

Algorithm 2 Tabu Search(maxIteration, maxConsecutiveIterations)

Input maxIteration, number of iterations; maxConsecutiveIterations, number of maximum consecutive iterations without an improvement
Output best solution found

```
Determine an initial solution
Set initial solution as best and current solution
while consecutiveIterations ≤ maxConsecutiveIterations and iteration ≤ maxIteration do
    Randomly select a type of neighborhood
    Discard moves that are infeasible or tabu
    Select best solution in neighborhood
    Set the current solution to the new solution
    Increment iteration
    if new solution improves best known solution then
        Set best known solution to new solution
        consecutiveIterations = 0
    else
        Increment consecutiveIterations
```

3.9.5 Greedy Randomized Adaptive Search Process

A GRASP is an iterative process, with each GRASP iteration consisting of two phases; the construction phase and a local search phase. The best overall solution is kept as the result. In the construction phase a feasible solution is iteratively constructed. Each iteration adds an element, which is determined by ordering a candidate list based on a greedy function. This makes it adaptive, as the benefits are updated at each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. By choosing one of the best, but not necessarily the top candidate randomization is added. It is almost always beneficial to apply a local search to try and improve the constructed solution (Feo & Resende, 1995). The algorithm used by Rieck et al. (2014) is given in Algorithm 3. In this algorithm, R is a set of requests for a fixed value of β, $solve\_TP$ refers to the solving of the transportation problem. This gives the estimate point-to-point requests. Based on that information the iterative phases of construction and local search are applied. For a more detailed description the reader is suggested to read Rieck et al. (2014).

3.10 Food bank Literature

Brock and Davis (2015) looked into the irregular supply from local businesses such as supermarkets and bakeries. They noted that quantifying food availability is complicated due to three reasons. First, there is no definite time interval between food collections. Second, the availability of surplus food at supermarket branches is dependent upon product sales, internal forecast accuracy, and donations to other non-profit organizations. Thirdly, food is perishable and must be collected before disposal. They compared four forecasting methods to estimate the amount of different in-kind food types available for collection. In their results

Algorithm 3 GRASP($\Delta\beta, \alpha, limit\_iter$

Input $\Delta\beta$, step parameter for cost matrix; α, Restricted Candidate List (RCL) quality parameter; $limit\_iter$, number of iterations
Output X_{best}, best solution found

$X_{best} = \emptyset; f(X_{best}) = +\infty$
$find\_shortest\_path(G, C^a, C^d, S^a, S^d)$
$\beta = 0$
while $\beta \leq 1$ **do**
    $R = solve\_TP(\beta)$
    for iter = 1 to limit_iter **do**
        $X = constructSolution(\alpha, R)$
        $X = localSearch(X)$
        if X is better than X_{best} **then**
            $X_{best} = X$
    $\beta = \beta + \Delta\beta$
return X_{best}

all forecasting methods overestimated total collections for the future planning period. The Multi-Layer Perceptron Neural Network (MLP-NN) models outperform the other models, at little additional transportation costs. Therefore usage of these models, with a forecasting period of a calendar week is recommended by the authors.

Hau, Fang, and Shi (2021) created an algorithm to estimate the number of pallets needed for each order by food banks. Food bank staff used to make estimates using heuristics, but due to an increase in demand the limitations of this process became more evident. In order to do this they assumed estimates of each product's dimensions. A heuristic pallet-packing algorithm was run, which resulted in the most likely number of pallets needed. Their algorithm could improve the logistical planning and avoid mis-allocating resources.

Martins, Melo, and Pato (2019) redesigned a food bank supply chain network in a triple bottom line context. A triple bottom line context means that the social and environmental impact are measured, next to the financial performance. Each of these formed an objective function in their proposed mixed-integer linear program. In their study, the food banks represent a warehouse that serves as collection and distribution point and each of the food banks maintains their own fleet of vehicles (or rent transport if needed). With their model they managed to assist decision makers with location decisions and with logistics decisions for food collection and distribution.

Davis, Sengul, Ivy, Brock, and Miles (2014) aimed to schedule food bank collections and deliveries to ensure food safety. They created a periodic vehicle pick-up and delivery model with backhauls. First they selected locations for food banks with the use of a linear model, after which they used a periodic vehicle pick-up and delivery model with backhauls to determine the weekly schedule.

Eisenhandler and Tzur (2019) studied the logistic challenges of a food bank that uses vehicles of limited capacity to distribute food collected from supplier to welfare agencies. There is a maximum travel time involved, so they aim to maintain equitable allocations to the different agencies, while delivering as much food as possible overall. Similarly, Nair, Grzybowska, Fu, and Dixit (2018) presented a scheduling and routing model that aims at simultaneously selecting a visit combination for each food provider and welfare agency. In both cases a linear model was given, but Eisenhandler and Tzur solved it with the use of Large Neighborhood Search, while Nair et al. used a Tabu Search.

Table 6: Summary of the reviewed literature on VRPs

Article	**VRP**	**Objective function**	**Solution method**
Aksoy and Kapanoglu (2012)	multi-commodity, multi-depot, heterogeneous vehicle fleet, pickup and delivery	Minimize costs	MILP
Ríos-Mercado et al. (2013)	multi-depot, multi-commodity, pickup and delivery, time windows, heterogeneous vehicle fleet	Minimize costs	GRASP
Rieck et al. (2014)	multi-depot, multi-commodity, pickup and delivery problem with hubs	Minimize costs	MILP & Fix-and-Optimize Scheme & Genetic Algorithm
Irnich (2000)	multi-depot, single hub, heterogeneous vehicle fleet	Minimize costs	Set Partitioning Algorithm
Wasner and Zäpfel (2004)	multi-depot hub-location vehicle routing model	Minimize costs	Hierarchical method embedded in a local search
Kevin et al. (2019)	hub-location and routing problem	Minimize costs	LINGO / MILP
Yang et al. (2019)	hub-location and routing problem	Minimize costs	CPLEX / MILP
Rafiei et al. (2013)	intermediate facilities, heterogeneous fleet, different zones	Minimize costs	MILP
Crainic et al. (2004)	intermediate facilities	Minimize costs	MILP / CPLEX

Continued on next page

Table 6: Summary of the reviewed literature on VRPs (Continued)

Crainic et al. (2009)	two-echelon, multi-depot, heterogeneous fleet, time windows	Minimize costs	Hierarchical Decomposition
Paul et al. (2021)	two-echelon, simultaneous pickup and delivery, time windows	Minimize travel time and fuel consumption	Objective VNS
Belgin et al. (2018)	two-echelon, simultaneous pickup and delivery	Minimize costs	Variable Neighborhood Descent and local search
Zhou et al. (2022)	multi-echelon, time windows, pick up and delivery	Minimize costs	Tabu Search
Cattaruzza, Absi, Feillet, and Vigo (2014)	multi-commodity, multi-trip, time windows	Minimize costs	Iterative local search
Battarra et al. (2009)	multi-commodity, multi-trip, time windows	Minimize amount of vehicles used	Routing heuristic and Packing heuristic
Gjenstø and Vaksvik (2008)	multi-commodity, time windows, pickup and delivery	Minimize costs	Tabu Search
Naccache et al. (2018)	multi pickup and delivery, time windows	Minimize costs	Branch-and-Bound and Variable Large Neighborhood Search
Moshref-Javadi and Lee (2016)	multi commodity, split delivery	Minimize latency	Simulated Annealing, VNS

Table 7: Summary of the reviewed literature on food banks

Article	Aim
Brock and Davis (2015)	Irregular supply from local business to the food banks
Hau et al. (2021)	Estimate the number of pallets needed per order
Martins et al. (2019)	Redesign of the food bank supply chain network
Davis et al. (2014)	Schedule collections and deliveries to ensure food safety
Eisenhandler and Tzur (2019)	Logistic challenges of equitable allocation and overall delivery
Nair et al. (2018)	Simultaneous visit of each food provider and welfare agency

3.11 Conclusion

The aim of this chapter was to answer the following research questions: *What does the literature say about VRPs?*,*What does literature mention about food bank supply chains?*. It can be said that there is a substantial amount of literature on the VRP. Plenty are similar to the problem this thesis is facing, however none seem to be an exact match. All together there are examples for each of the characteristics. However, often they miss or differ in one of the characteristics needed. Closest to the problem this research is facing is the model proposed by Zhou et al. (2022). The food bank requires a multi-commodity, pick up and delivery VRP with time windows and heterogeneous fleet of vehicles. As Zhou et al. (2022) solves their problem with the use of Tabu Search, it was decided to solve the problem of the food banks with the VNS for variability. Regarding the food bank supply chains, some literature can be found on the creating and changing of food bank supply chains. Often these food banks have a different supply chain than those in the Netherlands, and thus also the region of Twente-Salland. This means, that the research done for food banks has been limited. However, together with the literature on the VRP it becomes clear that a model can be made to determine the cost-efficient method of transport.

4 Solution Approach

This chapter provides the design of a solution to the core problem. First Section 4.1 gives all the requirements the solution must adhere to. Section 4.2 follows this with the assumptions in the model. Next, Section 4.3 gives the mathematical formulation in the form of a Mixed Integer Linear Program (MILP), while Section 4.4 determines how the pronlem will be solved with the use of the VNS. Lastly, Section 4.5 concludes this chapter by answering the related research questions.

4.1 Requirements

Like many problems coming from real-world applications, the solution must adhere to certain requirements. These requirements are specified in this section.

- Each food bank can only be visited once per vehicle.
- A vehicle cannot transport more than its maximum number of pallets per trip.
- A vehicle must start and end at the same location. However, unlike regular VRPs this does not have to be the depot/RDC but can be any of the food banks.
- Frozen, cooled and dry products have to be transported separately in order to apply to the rules and regulations. This means that each vehicle can transport only one product group at once.
- Food banks must be visited when their volunteers are present. Therefore the food banks have time windows during which they must be serviced.

Each of these requirements will become constraints in the MILP, and taken into consideration when creating the VNS.

4.2 Assumptions

In this section the assumptions made for the VRP model are given. These are needed to simplify the modelling of the problem, as well as structuring it.

- Based on the expertise of the volunteers at the RDC ten percent of the products going to the local food banks is frozen. Both cold and dry products make up 45 percent. This means that from the demand as given in Table 2 (Section 2.2), 10 percent will be frozen, 45 percent will be cooled and 45 percent are dry products. This is demand is recalculated and rounded up as it is not possible to transport partial pallets. This also means that the demand is determinstic, and will not change over time, as the average for each day in the week is used.
- It is not precisely known how often the different food banks supply to the RDC. Based on the expertise of volunteers of the RDC Deventer, assumptions have been made. These are summarized in Table 3 in Section 2.2.2. This means that the supply

is deterministic and will not be different over time. It happens on occasion that food going from food banks to the RDC is cooled or frozen food, but this is rarely. Therefore it can be said that the food banks only supply the RDC with dry groceries. This means that there is only one commodity for supply. Therefore, all supply can be transported together.

- Loading a vehicle with the maximum number of pallets, the weight does not go over the legal limit, determined by the vehicle and drivers license of the driver. Besides there is no limitation with regard to the unloading of goods. Assumed is that this is possible at each of the local food banks regardless of the type of vehicle.
- The drivers do not form a limitation. This means that only the number of vehicles and the type of vehicles have to be taken into consideration.
- Empty packaging can be returned via different food banks, without taking capacity, to the RDC.
- The time windows for each of the food banks are independent of the product type, while the RDC in Deventer is not bounded by a time window. Furthermore the travel times are deterministic. Waiting is allowed. If a vehicle arrives at a food bank before the beginning of the food bank's time window, the vehicle will wait.
- Once a vehicle has been assigned to transport a product type, it cannot transport another product type in the same day. The exception to this rule is when demand of a product cannot be satisfied by another vehicle. In that case the food bank's own vehicle is used to travel to the RDC and back. This vehicle is not added to the list of used vehicles, and thus it may happen that it transports products multiple times. After, it becomes part of the local search heuristic, and will be allowed to transport for more than one food bank.
- At the food banks there is enough storage space for all the demand of each product type. Delivery thus is not a problem. The supply towards the RDC is always less than the demand delivered to a food bank. Thus, a vehicle will have enough space if this is combined.

These assumptions form limitations to the model, and influence the results. With different assumptions, different choices are made, possibly resulting in different results. However, the assumptions are deemed to be reasonable, and therefore can be used.

4.3 Mixed Integer Linear Program

Pick up and delivery are no longer to be taken into consideration, as based on the assumptions the amount to be picked up will always fit within the vehicle. This also means that there are no longer multiple depots, but only the RDC functions as a depot. Thus the problem can be seen as a VRP with multiple commodities, time windows and heterogeneous fleet. The next section will give the MILP for this type of VRP based on the formulation by Moshref-Javadi and Lee (2016). However, as the problem context is different, some of the constraints have been adjusted.

4.3.1 Parameters

$\mathcal{N}$ Set of locations, indexed by i and j. Locations 0, .., N were 0 and N refer to the RDC,
$\mathcal{K}$ Set of vehicles, index by k,
$\mathcal{P}$ Set of product groups, index by p,
c_{ijk} Cost of driving from location i to location j using vehicle k,
cv_k Cost of using vehicle k,
s_{ij} Travel time from location i to location j,
Q_k Capacity of vehicle k,
org_k Original location of vehicle k,
d_{ip} Demand of product p at location i,
a_i Location i can only be served after this time,
b_i Location i needs to be served before this time,
M Large positive constant.

4.3.2 Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ visits customer } j \text{ immediately after visiting customer } i \\ 0 & \text{otherwise} \end{cases},$$

$$y_k = \begin{cases} 1 & \text{if vehicle } k \text{ is being used} \\ 0 & \text{otherwise} \end{cases},$$

t_{ik} Arrival time of vehicle k at location i,
z_{ipk} The quantity of product p transported to location i using vehicle k.

4.3.3 Objective function

$$\min \quad \sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{\mathcal{N}-\{0,N\}} \sum_{k=0}^{\mathcal{K}} (c_{ijk} * x_{ijk}) + \sum_{k=0}^{\mathcal{K}} (y_k * cv_k) + \sum_{k=0}^{\mathcal{K}} y_k * c_{org_k 0k} + \sum_{k=0}^{\mathcal{K}} \sum_{i=0}^{\mathcal{N}} x_{iNk} * c_{iorg_k k} \tag{4.1}$$

The objective function minimizes the total costs involved. The total costs are consisting of four parts. The first part is the sum of the distance-travelled cost. This is based on whether a vehicle travels between two locations, binary variable x_{ijk}, which is multiplied by the costs for travelling a distance unit. Travelling back to the depot is excluded, as the vehicle does not actually return to the depot, but to the original location of the vehicle. The second part is the cost of the usage of the vehicles. For that the costs of the vehicle are multiplied with binary variable y_k. The third part adds the distance between the original location of a vehicle and the depot in case the vehicle is used. As this is not part of the routes yet, but the costs are there, this had to be included. Finally, the fourth part determines from which location the vehicle goes back to the depot. For this the distance between that location and the original location of the vehicle is added. By doing so the right distance is used for the calculation of the costs. The difference between the route given by the MILP, and the route used to calculated for the cost becomes visible in Figure 7. The route used to calculate the costs is the route the vehicle will eventually drive.

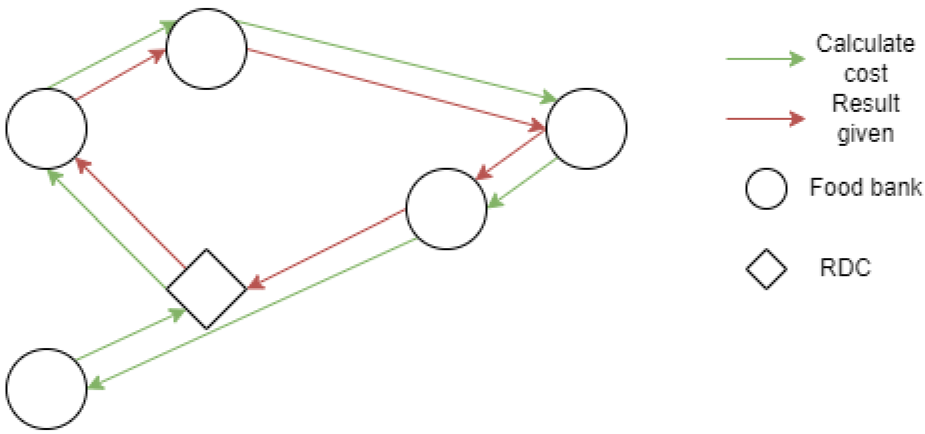

Figure 7. Route and cost explanation

4.3.4 Constraints

$$\sum_{i=0}^{N} x_{ijk} - \sum_{i=0}^{N} x_{jik} = 0 \qquad k = 0,..,\mathcal{K}, j = 1,..,N-1 \tag{4.2}$$

$$\sum_{p=0}^{\mathcal{P}} \sum_{i=1}^{N-1} z_{ipk} \leq Q_k \qquad k = 0,..,\mathcal{K} \tag{4.3}$$

$$\sum_{j=1}^{N-1} d_{jp} \sum_{i=0}^{N} x_{ijk} <= Q_k \qquad k = 0,..,\mathcal{K} \tag{4.4}$$

$$\sum_{j=1}^{N-1} x_{0jk} = y_k \qquad k = 0,..,\mathcal{K} \tag{4.5}$$

$$\sum_{i=1}^{N-1} x_{iNk} = y_k \qquad k = 0,..,\mathcal{K} \tag{4.6}$$

$$\sum_{k=0}^{\mathcal{K}} z_{ipk} \geq d_{ip} \qquad i = 1,..,N-1, p = 0,..,\mathcal{P} \tag{4.7}$$

$$M \sum_{j=1}^{N} x_{ijk} \geq z_{ipk} \qquad p = 0,..,\mathcal{P}, k = 0,..,\mathcal{K}, i = 0,..,N-1, j \neq i, \tag{4.8}$$

$$t_{ik} + s_{ij} - t_{jk} \leq (1 - x_{ijk})M \qquad i = 1,..,\mathcal{N}, j = 1,..,\mathcal{N} k = 0,..\mathcal{K}, i \neq j \tag{4.9}$$

$$t_{ik} \geq a_i \qquad i = 0, ..\mathcal{N}, k = 0, .., \mathcal{K} \tag{4.10}$$

$$t_{ik} \leq b_i \qquad i = 0, ..\mathcal{N}, k = 0, .., \mathcal{K} \tag{4.11}$$

$$\sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} x_{ijk} \leq y_k \qquad k = 0, .., \mathcal{K} \tag{4.12}$$

$$x_{ijk} \in \{0, 1\} \qquad i = 0, .., \mathcal{N}, j = 0, .., \mathcal{N} k = 0, .., \mathcal{K} \tag{4.13}$$

$$y_k \in \{0, 1\} \qquad k = 0, .., \mathcal{K} \tag{4.14}$$

$$t_{ik} \geq 0 \qquad i = 0, .., \mathcal{N}, k = 0, .., \mathcal{K} \tag{4.15}$$

$$z_{ipk} \geq 0 \qquad i = 0, .., \mathcal{N}, k = 0, .., \mathcal{K}, p = 0, .., \mathcal{P} \tag{4.16}$$

Constraints (4.2) ensures that each time a vehicle arrives at a location, the location is also left again. Constraints (4.3) and (4.4) ensure that the amount of products transported by a vehicle does not exceed the capacity of this vehicle. Constraints (4.5) and (4.6) ensure that a vehicle leaves from the depot and goes back. Due to the objective function, the distance used for this is not equal to the distance to the depot, but to the original location of vehicle k. Constraints (4.7) ensures that the amount of products delivered is at least the demand of the location. These products cannot be delivered unless the location is visited by the vehicle. This is enforced by Constraints (4.8). Constraints (4.9) is the subtour elimination, and also calculates the time of arrival at a location. Constraints (4.10) and (4.11) do not allow the location to be visited outside the time windows. Constraints (4.12) determines whether a vehicle is being used or not. Finally, Constraints (4.13) to (4.16) are the integrality constraints.

There are a few differences between this MILP and the MILP implemented. The implemented MILP requires each of the location to be copied as many times as they have demand for different commodities. This also means that the vehicles are copied, as the can be used for any commodity. To ensure that each vehicle is not used more than once, the MILP includes constraints for this. The sum of the vehicle-copies used is set to be less or equal to one. Distances, as well as travel times, between copies of locations are set to be large, and therefore unwanted to be paired as it would significantly increase the costs. This way, it is prevented that different commodities are transported by the same vehicle. Thus, the behavior of the MILP is equal, but the constraints and data are slightly different.

4.4 Solution Methodology

For larger instances it will not be possible to find a feasible solution within an acceptable time with the use of the MILP. Therefore, like literature also determined, a heuristic is needed. It is decided to use the Variable Neighborhood Search as similar VRPs in literature also used this metaheuristic. However, the closest VRP found used Tabu Search, and thus, by using the VNS instead, it is shown that this metaheurisitc is also suitable for this type of problem.

4.4.1 Variable Neighborhood Search

In Algorithm 4 the VNS is given. The difference between this algorithm and the standard VNS algorithm provided Section 3.9 is the initialization and the shaking phase. Two types of initialization are possible, one of which is randomized. Both of these are explained in Section 4.4.2. This is then followed by a shaking phase, in which a random neighbor is chosen. While the standard VNS algorithm has the same type of shaking throughout all iterations, the shaking phase of this algorithm changes throughout the iterations. From this newly generated solution, the local search takes place. In this search, each neighbor is evaluated and the best one returned. The overall best solution is being remembered, before returning to the shaking phase. It does so until the maximum amount of iterations without improvement of the overall best solution has been found.

Algorithm 4 Variable Neighborhood Search

```
Input maxIteration, number of iterations without improvement
Output bestroutes, best solution found; bestcost, cost of bestroutes
InitialSolution() or RandomizedInitialSolution()
while j ≤ maxIteration do
    ShakingPhase(InitialSolution, NumberofShakes, j)
    LocalSearch()
    Update bestroutes, bestcost
return bestroutes, bestcost
```

4.4.2 Initialization

The VNS needs an initial solution in order to be able to search its neighbors. As the initial solution has an influence on the possible outcomes of the VNS, it was determined to create an ordered initial solution, in which the first vehicle in the list is used first, the second vehicle is used second, etc. Besides, a randomized initial solution is introduced, in which the vehicle used is chosen randomly from a list of vehicles that have not yet been used. Furthermore, it is important that the solution is feasible. Thus a list of locations with demand or supply is generated. Each of which needs to be visited. From the vehicle chosen (either by order, or random), it is determined which location in the list is closest, while also being able to visit them with all demand or supply, and within the opening hours (time windows) of the location. If this is the case, the location is added to the route of the vehicle. It can happen that locations cannot be combined with the vehicles that have not been used yet. In that case, the location goes to the RDC themselves to get their demand.

Algorithm 5 InitialSolution()

```
Input
Output allroutes, route per vehicle for each product
for p in products do
    if p = 0 then
        InitializationFunction(supply)
    else
        InitializationFunction(demand)
return allroutes
```

Algorithm 5 refers to Algorithm 6. This is done such that it is more readable. The input, supply or demand, is what needs to be read when Algorithm 6 says variable.

Algorithm 6 InitializationFunction(variable)

Input variable, determines whether to look at supply or demand
Output routes

1: create variable list of locations with variable
2: **for** k in vehicles **do**
3: **if** vehicle location is in variable list **and** vehicle is not used **then**
4: **while** variable list != empty **do**
5: Set variable allocated so far, next stop = variable location, last stop = vehicle location, current time to end of time window original location
6: **for** j in variable list **do**
7: **if** variable allocated < capacity **and** current time - travel time j to last stop within time window j **and** j to last location = shortest distance **then**
8: next stop = j
9: **if** next stop != last stop **and** next stop in variable list **then**
10: Update variable allocation, current time
11: insert next stop in route, remove next stop from variable list
12: **else**
13: break
14: **if** variable list != empty **then**
15: create route [i, 0, i]
16: add the routes to allroutes
17: **Return** allroutes

Besides, a slightly randomized initial solution is created. The algorithm is the same as 5, with the difference being that it calls for Algorithm 7 instead of Algorithm 6 to deal with the random vehicle picking. This difference between the latter two algorithms is shown on line 3. There are no other differences.

Algorithm 7 InitializationFunctionRandom(variable)

Input variable, determines wheter to look at supply or demand
Output routes

1: create variable list of locations with variable
2: **for** k in vehicles **do**
3: pick random, available vehicle
4: **if** vehicle location is in variable list **and** vehicle is not used **then**
5: **while** variable list != empty **do**
6: Set variable allocated so far, next stop = variable location, last stop = vehicle location, current time to end of time window original location
7: **for** j in variable list **do**
8: **if** variable allocated < capacity **and** current time - travel time j to last stop within time window j **and** j to last location = shortest distance **then**

next stop = j
if next stop != last stop **and** next stop in variable list **then**
Update variable allocation, current time
insert next stop in route, remove next stop from variable list
else
break
if variable list != empty **then**
create route [i, 0, i]
add the routes to allroutes
Return allroutes

4.4.3 Shaking

With the use of the initial solution the next step of the VNS can take place. This is the shaking phase. Here a neighbor of the last solution is found. This is done with the help of an operator. These operators are the randomized version of the operators in Section 4.4.5. If the neighbor found is feasible, the route is updated. Else it is ignored. To make it more likely to get out of a local optimum, the number of times a neighbor is generated is input to be given.

Algorithm 8 ShakingPhase(InitialSolution, NumberofShakes)

Input InitialSolution, the solution that needs to be shaken; NumberofShakes, the amount of shakes needed

Output allroutes, route per vehicle for each product

for i *leq* NumberofShakes **do**
Shake
if New solution feasible **then**
Update route to be used

Furthermore, it is possible to adapt the shaking, based on the iteration the VNS is in. Dependent on the iteration a different operator can be chosen to do the shaking, or the number of shakes can be different.

4.4.4 Local Search

After the shaking, the next step is the local search. In the local search, all direct neighbors are evaluated. If an overall best solution has been found, this solution goes to the shaking phase. Else, the next operator for finding neighbors is used. This is the case for all 4 operators, which are described in Section 4.4.5. This means that the solution is only updated when an overall best is found. Only a strictly better solution is returned by the operators, to prevent loops between two solutions having the same best cost.

Algorithm 9 LocalSearch(routeforls, bestcost)

Input routeforls the solution that goes through the local search; bestcost, lowest cost found so far

Output bestroute, route per vehicle for each product

```
best found = False
while best found != True do
    for all operators do
        if best found != True then
            newestroute = Operator(routeforls)
            if cost newestroute < bestcost then
                update bestcost, bestroute
                best found = True
    break
```

4.4.5 Operators

VNS makes use of different operators for the shaking and local search phases. In this case the operators used are Swap, Move, Change Vehicle, and Reverse Route. These operators are based on the research done by Hansen and Mladenović (2001) and Kuo and Wang (2012). In their research they used similar operators. In this section these operators are discussed. Each of the operators only returns a new route if the cost are strictly smaller than the previous best cost. This is done to prevent having a loop in which two solutions have the same best cost.

Swap

In the swap operator two locations are randomly chosen. These locations can be located in the same vehicle, or in two different vehicles. These vehicles will be transporting the same product type, so the different commodities will not be mixed after using this operator. In Figure 8 a swap is shown. The two chosen locations are highlighted in green and blue. After choosing the two locations, they are swapped, creating two new routes. From this figure it follows that the two locations do not Algorithm 10 gives the pseudocode for this operator.

Algorithm 10 Swap(route, bestcost)

Input route, a solution; bestcost, best known result

Output bestroute, a new route found; bestcost, the cost for the route

```
for p in {Products} do
    for k in {Vehicles} do
        if vehicle k transports product p then
            Add visited locations to list
    for x in {Locations} do
        for y in {Locations} do
            if x != y then
                Swap location x and y
                Calculate cost
                if cost < bestcost and solution feasible then
                    update bestroute and bestcost
if Best result found then
    Return bestroute, bestcost
```

```
else
    Return routeforls, bestcost
```

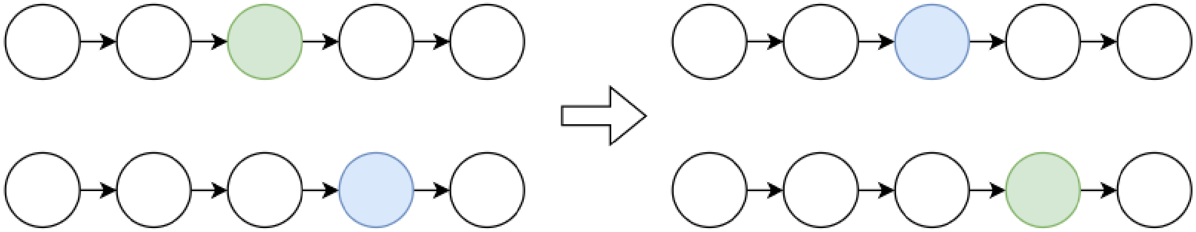

Figure 8. Swap operator

Move

In the move operator only one location is moved simultaneously. It can happen that this happens within the same vehicle, but at a different place in the route, but it is also possible that the location moves to a different vehicle. The selected location is deleted from its original route, and inserted at every other possible spot. As can be seen, Figure 9 shows a move between vehicles. The green location is deleted from its original vehicle's route and moved into the route of another vehicle.

Algorithm 11 Move(routes, bestcost)

Input routes, a solution; bestcost, best known result
Output bestroute, new routes formed; bestcost, the cost for the route

```
for p in {Products} do
    for k in {Vehicles} do
        if vehicle k transports product p then
            Add visited locations to list
    for x in {Locations} do
        for y in {Locations} do
            if x != y then
                Remove location x
                Insert location x at location y
                Calculate cost
                if cost < bestcost and solution feasible then
                    update bestroute and bestcost
                else if cost < movecost then
                    update moveroute and movecost
if Best result found then
    Return bestroute, bestcost
else
    Return routeforls, bestcost
```

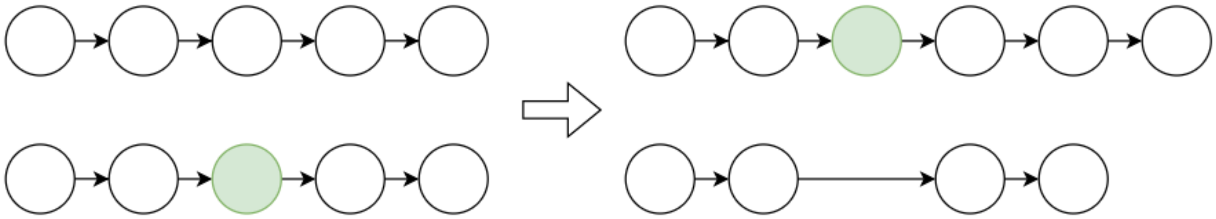

Figure 9. Move operator

Switch vehicles

When switching vehicles, all locations that are not the original location of the vehicle and the depot are being swapped with those of another vehicle. This means that there are two vehicles to be chosen. Once again, this is done for vehicles transporting the same product type. This way the different commodities do not get mixed. Figure 10 shows the original routes and the resulting routes after using this operator. Unlike the previous operators, here multiple locations are moved, if the vehicle's route contains multiple locations that are not the depot and/or original location of a vehicle.

Algorithm 12 SwitchVehicles(routes, bestcost)

```
Input route, a solution; bestcost, best known result
Output bestroute, new route; bestcost, corresponding cost for the route
for p in {Products} do
    Determine vehicles used in transporting the product
    for x in {Vehicles} do
        for y in {Vehicles} do
            if x != y then
                Swap sequence of vehicle 1 and vehicle 2
                Calculate cost
                if cost < bestcost and solution feasible then
                    update bestroute and bestcost
if Best result found then
    Return bestroute, bestcost
else
    Return routeforls, bestcost
```

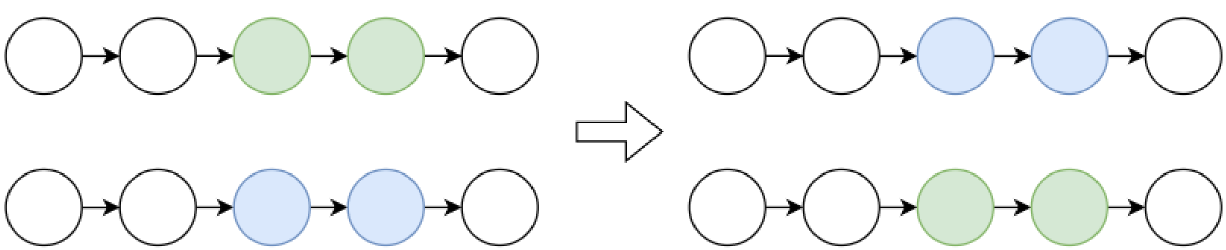

Figure 10. Switch vehicle operator

Reverse route
Lastly, the reverse operator reverses the route between depot and original location of a vehicle. This means that only one vehicle has to be selected. The entire sequence of locations to be visited between depot and original location are then reversed. This is also shown in Figure 11. Where first the green location is visited, the blue location is visited first after the reversing of the routes. The depot and original location of the vehicle are not taken into consideration.

Algorithm 13 ReverseRoute(routes, bestcost)

```
    Input routes, a solution; bestcost, best known result
    Output bestroutes, new route; bestcost, cost corresponding to the route
for p in {Products} do
    for k in {Vehicles} do
        Add vehicles to list
    for x in {VehicleList} do
        Reverse route
        Calculate cost
        if cost < bestcost and solution feasible then
            update bestroute and bestcost
if Best result found then
    Return bestroute, bestcost
else
    Return routeforls, bestcost
```

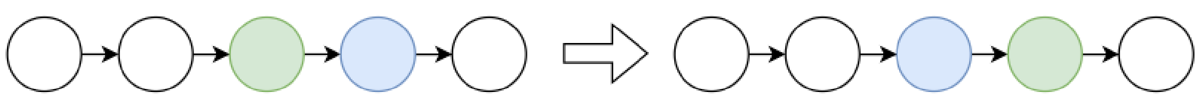

Figure 11. Reverse route operator

The initialization, the amount of iterations without an improvement, and the order of the operators are dependent on the results of the fine tuning of the VNS. This fine tuning is explained, and results are shown, in Chapter 5.

4.5 Conclusion

In this chapter the research question: *How should the solution approach be designed?* and respective sub questions, have been answered. First, the requirements and assumptions related to the MILP and VNS are listed. These consider all the constraints the MILP and VNS must adhere to and the assumptions needed to create suitable solutions respectively. Next a multi-commodity VRP with time-windows and heterogeneous fleet has been defined. Together with the assumptions, this covers the core problem to be solved. As a MILP can not solve large instances within reasonable time, a heuristic has been designed to solve the bigger problem. The objective function and constraints have been given and explained. This heuristic is based on the nearest neighbor to create an initial solution and VNS for further optimization. There are different methods of initialization, namely ordered and random. Besides, the VNS may be adaptive, meaning that the shaking phase differs per iteration. Furthermore, the local search phase of the VNS makes use of different operators, namely Swap, Move, Switch Vehicles and Reverse Route. These operators have been explained in detail. Together, the MILP and VNS should be able to find routes that handle all the requirements.

5 Evaluation

In this chapter, the evaluation of the created MILP and VNS are discussed. First, the experiment design is given in Section 5.1. This describes taken to find the eventual results. Next are the data instances used for fine tuning and numerical experiments in Section 5.2. Section 5.3 describe and analyse the results from the parameter fine tuning, while Section 5.4 give the numerical experiments. This is concluded in Section 5.5.

5.1 Experiment Design

Within this research several experiments are needed. The first experiments are related to the parameter settings for both the MILP and VNS. For the MILP the maximum computational time is determined, while the VNS the experiments are related to the initialization, shaking and local search phases. Once the parameter settings have been determined, the numerical experiments for the food banks can take place. However, for both of these data is needed. Test data is created for the parameter tuning, to make sure the MILP and VNS are generally applicable, and not tuned to the specific situations of the food bank. The size of the data instances are set to be smaller, equal and larger than the size of the real world data. However, the real world data is not ready for use and needs to be (partly) determined, as well as the scenarios that are run for the food banks. Figure 12 summarizes these steps.

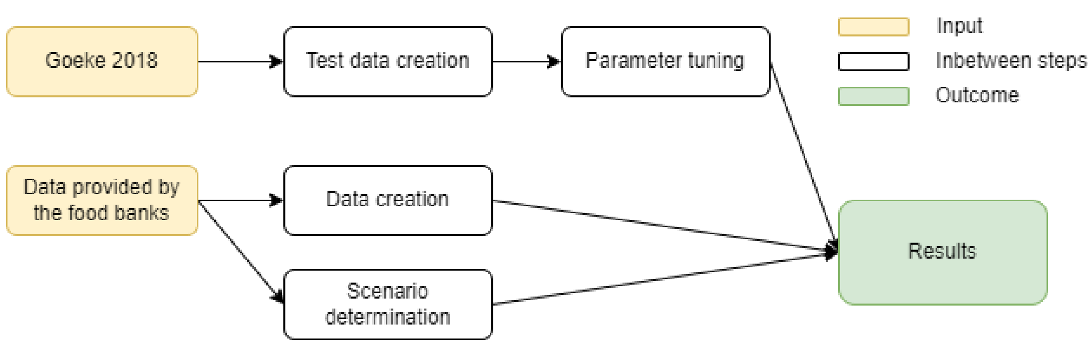

Figure 12. The experimental steps to find the results.

5.1.1 Technical Details

In order to conduct these experiments, a Windows computer with 16GB RAM, Intel i5 core and 2.4 GHz core is used. The VNS code is written in Python 3.8.10, using the Spyder 5.3.1 IDE. The MILP is also coded in Python, making use of the MIP package. The MILP is solved with the help of Gurobi, which is free to use under an academic license.

5.2 Data Instances

This section gives the data instances for the parameter tuning, the real world data, as well as the scenarios run with the real world data. Finally the costs for the current situation are calculated, such that a fair comparison can be made.

5.2.1 Data instances for parameter tuning

The instances will have between 3 and 15 food banks plus one depot. There are between 3 and 8 food banks that have a demand on the same day (see Table 2). To also determine if the MILP and VNS are feasible for larger instances there are test instances with more food banks. All the food banks have at least one vehicle, therefore the different data instances will have a minimum of one vehicle per food bank (number of locations minus one). There are six instances. They are paired such that there are two instances with a similar number of locations. These instances differ on the number of vehicles as one of the instances will have more vehicles. Several of the food banks also have multiple vehicles (see Table 5) and it will be interesting to see the influence on the results. The distances, time windows, and travel times follow from different data sets of Goeke (2018) in the VRP-REP[10]. These data sets included coordinates instead of distances. The distances have been assumed to be Euclidean and calculated accordingly. A speed of the vehicle is given. The speed is used to translate the distances into travel times. A speed of 0.1 means that it takes 0.1*distance to travel the distance. The demand for each location is determined randomly with the help of a random number generator. These demands are similar to those mentioned in Table 2. It is specifically decided to draw these once and added to the data files, and not use Python and draw them every time the data is loaded for repeatability. The capacity of the vehicles are similar to those mentioned in Table 5, but it ensured the total capacity is more than the total demand. Furthermore, for 3 of the 6 instances, there will be some supply that needs to be brought to the depot from the different locations. The full overview of the data instances is given in Table 8.

Table 8. The data instances used to fine tune the MILP and VNS. As well as the adaptions made. Each of the data instances follow from a set by Goeke (2018). The more detailed data used for the testing is available upon request.

Data instance	Data set	Number of locations	Number of vehicles	Speed	Supply
D1	r202C6	4	3	0.1	Yes
D2	c101C6	6	7	0.05	No
D3	lrc208	9	8	0.1	No
D4	c101C12	11	15	0.1	Yes
D5	rc102C12	13	12	0.1	Yes
D6	rc108C16	16	20	0.05	No

5.2.2 Data for numerical experiments

It is important to use consistent data throughout all scenarios. Several food banks provided financial results. These results are given in Table 9. From these results the costs per day and the costs per kilometer are calculated. As the costs per day, and costs per kilometer are used these are averaged using the known data for the food banks of which details are not known. Besides, the assumed costs for new vehicles to be used in some of the scenarios are given. With the exception of new vehicles, the costs are based on 2021. Regarding

[10]An online repository with data sets for different variants of the VRP. http://www.vrp-rep.org/

the fuel, which has seen a significant increase in cost, this makes for a significant difference between transportation methods. The averages, as well as the new vehicles, show the costs per vehicle. It was assumed that the new vehicles have a capacity of 8 pallets per vehicle. Likely these vehicles could transport more pallets at the same time, but otherwise the weight limit may be exceeded.

Table 9. Standard and variable costs per vehicle per food bank. The standard cost consist of either the lease, or the maintenance, taxes and insurance. The variable costs are the fuel costs.

Food bank	Vehicle	Standard cost(€)	Fuel cost(€)	KM driven	cost/day(€)	cost/km(€)
Almelo	1	6000	4200	29400	16.44	0.143
Almelo	2	6000	4000	28000	16.44	0.143
Almelo	3	6000	1400	11200	16.44	0.125
Enschede	Combined	17714	9566	-	16.18	-
Midden-Twente	Combined	14069	10719	72898	19.27	0.147
Oost-Twente	1	8117	4723	-	22.24	-
Raalte	Combined	12131	2382	-	16.62	-
Average	-	-	-	-	17.66	0.140
RDC Transport	Each	25662	5100	30000	70.31	0.17

Furthermore, in Appendix C the detailed calculations regarding the food bank in Almelo are given for a full overview. Data of the other food banks has been calculated similarly.

5.2.3 Scenarios

There are several scenarios to evaluate in order to determine the best solution for the food banks. First of all, from Section 2.2 it follows that the amount of pallets to be transported differ significantly per day. It is important to compare the results for the different days, such that the best overall solution can be found. This is needed for each of the variations. Besides, it is already known that not every food bank is interested in joining a collaboration. It is interesting to see how results differ when certain food banks, and therefore also their vehicles, are excluded. As the idea surrounding the pilot changed, this also lead to different scenarios, namely using the larger food banks as hub or not. As the inclusion of hubs significantly alters the MILP and/or VNS, there are three scenarios where the food banks supposed to act as hub get a new vehicle. During these scenarios, instead of four vehicles located at the RDC this means that each of the four largest food banks will have a vehicle at their location. This is slightly different from the actual hub situation, as these vehicles would each be assigned a commodity, instead of combining the commodities due to the small travel distance (which does not affect the food safety). The different scenarios are summarized in Table 10 which also mentions whether they are solved with the MILP and/or VNS. In the scenarios where the transport starts and ends at the RDC, the food banks in Zutphen and Vaassen are not taken into consideration, as they have indicated not to be willing to join such a collaboration. This is also the case for the scenarios with new vehicles at the locations of the hubs.

Table 10. The different scenarios for the different days and whether they have been solved by the MILP and/or VNS. Another distinction is the inclusion of Vaassen and Zutphen, or the use of transport from Deventer instead of their local vehicles. Lastly, there is the situation where the larger food banks serve as hubs for one of the smaller food banks.

Scenario ID	Scenario	MILP	VNS
1	Tuesday	no	no
2	Wednesday	yes	yes
3	Wednesday with Zutphen	yes	yes
4	Wednesday transport from Deventer	yes	no
5	Wednesday hubs	no	no
6	Wednesday new vehicles	yes	yes
7	Thursday	yes	yes
8	Thursday with Vaassen and Zutphen	yes	yes
9	Thursday transport from Deventer	yes	no
10	Thursday hubs	no	no
11	Thursday new vehicles	yes	yes
12	Friday	yes	yes
13	Friday transport from Deventer	yes	no
14	Friday hubs	no	no
15	Friday new vehicles	no	no

The scenarios where certain food banks are considered hubs changes the MILP and VNS significantly. It is for that reason these scenarios are not considered with the use of either. Instead, these are replaced by assigning the new vehicles to each of the hubs. This is not done for the Friday, as there are only two food banks with a vehicle in this scenario, and three commodities need to be transported, meaning three vehicles are necessary. Thus, no feasible solution can be found for this scenario. Besides, for the Tuesday there is no need to run either the MILP or VNS, as Almelo is the only food bank receiving food that day. Therefore there are no possibilities of combining the transport food for this scenario. Lastly, the code of the VNS contains check to determine if the original location of a vehicle has demand or not. If it does not have demand it cannot be appointed to be used. As the original location of the vehicles is the depot in Scenarios 4, 9 and 13, and the RDC does not have demand, no feasible solution can be found. Therefore, these scenarios are not taken into consideration for the VNS.

5.2.4 Current Situation

First it is important to know the current costs. In this current situation every food bank drives to the RDC and back themselves. For this, the food banks with demand (Table 2) and their vehicles (Table 5) and the costs associated with these vehicles (Table 9) are used. As they currently all drive to the RDC themselves, the costs consists of the distance * cost per km and the daily fixed cost per vehicle used. The reason not all vehicles are considered in the cost is because this will also not be the case for the experiments with the MILP and VNS. These costs do not include occasional trips to the RDC to deliver supply. The overview of these daily cost in the current situation is given in Table 11. Furthermore, the scenarios with hubs, and/or different vehicles have not been considered, as they are currently not used.

Table 11. Costs of the current situation

Scenario ID	Cost (€)	Distance	Number of vehicles
2	226.09	689	7
3	243.41	715	8
7	203.35	590	7
8	246.10	654	9
12	137.97	453	4

5.3 Parameter Fine Tuning

There are four parameters to define within the VNS, such that the optimal VNS regarding result and computational time is achieved. Firstly, the type of initialization is determined. The second is the number of iterations in which there is no progress made. Third, the order of the operators in the local search phase. Lastly, the operators in the shaking phase, and their order, which are applied, based on which iteration the VNS finds itself in. Each of these parameters are tested against multiple data instances. Besides, the MILP should be tuned to have a maximum computational time due to the trade off between result and time it takes to get a result.

5.3.1 Initialization

There are two possible methods of initialization. The first method starts with first vehicle and continues throughout the list, while the second method picks a random vehicle. These initialization are explained in Section 4.4. Because the initial solution determines the vehicles that are used for each commodity, it limits the possible outcomes when applying the VNS. For that reason it needs to be determined which method results in the best initialization. Therefore there are the following experiments:

Experiment I1: Ordered initialization. This initialization always gives the same result with the same data set. It therefore only needs one run.
Experiment I2: Randomized. This initialization generates different results with the same data set. To ensure the result of Experiment $I1$ is not compared to a outlier, this experiment is ran 5 times. The minimal, maximal and average values are reported, but it is the average that is compared.

With these experiments, all possible initial solutions are being tested, while making sure the conclusion is not based on an outlier. The results are given in Table 12. The first column refers to the data instance used. The second and third are the result from Experiment $I1$. For Experiment $I2$ the cost and time columns are split and give the minimum, average, and maximum values found.

Table 12. The results of the experiments to determine the initialization. The lower cost and computational time, compared between $I1$ and the average of $I2$ are given in bold.

Data instance	I1		I2					
	Cost	Time	Cost			Time		
			Minimum	Average	Maximum	Min	Average	Max
D1	183.01	0.0	147.47	**168.79**	183.01	0.0	0.0	0.0
D2	**210.52**	0.0	210.52	**210.52**	210.52	0.0	0.0	0.0
D3	**367.38**	0.0	325.11	381.93	426.24	0.0	0.0	0.0
D4	**477.42**	0.0	467.21	508.52	568.36	0.0	0.0	0.0
D5	608.47	0.0	496.06	**544.50**	602.79	0.0	0.0	0.0
D6	**720.19**	0.0	624.04	727.76	764.79	0.0	0.0015	0.0060

From the results in Table 12, it can be concluded that the random initialization can always find a better, or equal, result than the more systematic initialization. On average, it performs better, or equal, for half of the data sets. There is no clear characteristic that is responsible for these results. Therefore, there is no clear preference between the two initialization methods.

5.3.2 Number of iterations without improvement

First the number of iterations without improvement are determined. This is needed, as a more optimal result is wanted, but not at the cost of a very long computational time. By determining this first, the other experiments will benefit from a good trade off between computational time and results. These experiments are run for each of the data instances determined in the previous section. As there is some randomness involved in the shaking phase, each of the experiment-data instance combinations is run 5 times, and the minimum, average and maximum are given for both the cost and computation time. Besides the initialization is randomized, thus this is repeated 5 times for each of the data instances. The results of these experiments determine the number of iterations used for both the real-world data instances, as well as the experiments to determine the other parameters.

Experiment NI1: This experiment considers a low number of iterations, namely 10.
Experiment NI2: This experiment considers a medium number of iterations, namely 25.
Experiment NI3: This experiment considers a large number of iterations, namely 40.

The complete results for each of the experiments can be found in Appendix A. In Table 13, the average improvement in percentage and average computational time is given.

Table 13. The results of the experiments NI. In bold the best improvement, percentage wise, per data instance is given.

Data instance	NI1		NI2		NI3	
	Improvement(%)	Time	Improvement (%)	Time	Improvement (%)	Time
D1	0	0.0018	0	0.0203	0	0.0301
D2	0	0.0254	0	0.0571	0	0.0861
D3	12.78	0.1512	**18.15**	0.3713	15.20	0.5395
D4	0	0.1249	0	0.2795	0	0.4858
D5	4.04	0.2902	**6.00**	0.6998	5.14	0.9970
D6	**20.68**	0.8172	15.34	2.1178	19.44	3.4300

From Experiment NI several conclusions can be drawn. The first, noticeable conclusion is that for Data Instances $D1$, $D2$, and $D4$ there are no improvements, for each of the experiments. Based on the full results, it did not matter which solution was generated. Secondly, from the other data instances ($D3$, $D5$, and $D6$), it follows that increasing the number of iterations without improvement does not necessarily means a bigger improvement is found. In fact, Experiment $NI3$ did not find better improvements, on average, at all. Thirdly, overall the computational time increased, when the number of iterations without improvement grew. For Data Instance $D6$ specifically there was a significant higher computational time with the increase of the number of iterations without improvement. Fourth, the randomness within the shaking phase does not seem to influence the results of the VNS abundantly. Most of the time, each of the 5 runs resulted in the same solution. Finally, as there were significant differences in the solution dependent on the initial solutions, it can be said that the initial solution is more important for the result than the randomness of the shaking phase.

5.3.3 Order of local search operators

Throughout the previous experiments, the order of operators used is Switch Vehicles - Reverse Route - Move - Swap. However, the order in which the local search operators are applied can make a big difference in run time and solution quality. This is because the first operator is always tried, while the last operator will only be tried once the other operators did not find any improvement of the overall best. Therefore it is needed to test whether this is the best possible order of operators within the local search phase. With four operators, this means that there are twenty-four experiments. Below the experiments are mentioned. They will all be ran in the same manner. Considering the amount of experiments, only Data Instances $D1$, $D4$, and $D5$ are used, which stop after 10 iterations without improvement. Besides, the initialization is ordered, such that each experiment has to only be ran once, since it was concluded that the randomness in the shaking phase has little influence on the results.

Experiment S1: Swap - Move - Reverse Route - Switch Vehicles
Experiment S2: Swap - Move - Switch Vehicles - Reverse Route
Experiment S3: Swap - Switch Vehicles - Reverse Route - Move
Experiment S4: Swap - Switch Vehicles - Move - Reverse Route
Experiment S5: Swap - Reverse Route - Move - Switch Vehicles

Experiment S6: Swap - Reverse Route - Switch Vehicles - Move
Experiment S7: Move - Swap - Reverse Route - Switch Vehicles
Experiment S8: Move - Swap - Switch Vehicles - Reverse Route
Experiment S9: Move - Switch Vehicles - Reverse Route - Swap
Experiment S10: Move - Switch Vehicles - Swap - Reverse Route
Experiment S11: Move - Reverse Route - Swap - Switch Vehicles
Experiment S12: Move - Reverse Route - Switch Vehicles - Swap
Experiment S13: Reverse Route - Swap - Move - Switch Vehicles
Experiment S14: Reverse Route - Swap - Switch Vehicles - Move
Experiment S15: Reverse Route - Move - Swap - Switch Vehicles
Experiment S16: Reverse Route - Move - Switch Vehicles - Swap
Experiment S17: Reverse Route - Switch Vehicles - Swap - Move
Experiment S18: Reverse Route - Switch Vehicles - Move - Swap
Experiment S19: Switch Vehicles - Swap - Move - Reverse Route
Experiment S20: Switch Vehicles - Swap - Reverse Route - Move
Experiment S21: Switch Vehicles - Move - Swap - Reverse Route
Experiment S22: Switch Vehicles - Move - Reverse Route - Swap
Experiment S23: Switch Vehicles - Reverse Route - Move - Swap
Experiment S24: Switch Vehicles - Reverse Route - Swap - Move

Table 14. The results of the experiments S. In bold the best improvement, percentage wise, per data instance is given.

Experiment	DI1		DI3		DI6	
	Improvement(%)	Time	Improvement(%)	Time	Improvement(%)	Time
S1	0.0	0.0252	3.71	0.7526	2.39	2.5264
S2	0.0	0.0265	0.44	0.4246	7.77	2.4127
S3	0.0	0.0446	5.95	0.8650	2.48	2.3472
S4	0.0	0.0198	4.36	0.4568	0.83	1.9572
S5	0.0	0.0448	4.57	0.4807	13.06	4.7089
S6	0.0	0.0200	5.89	0.7197	7.03	3.3824
S7	0.0	0.0261	2.83	0.4697	**16.70**	3.6735
S8	0.0	0.0290	0.0	0.4101	2.39	2.2273
S9	0.0	0.0269	6.32	0.8497	10.75	5.2270
S10	0.0	0.0259	**7.22**	0.6365	2.59	2.1587
S11	0.0	0.0295	2.40	0.4618	5.30	2.3858
S12	0.0	0.0259	6.32	0.6896	10.20	4.4204
S13	0.0	0.0299	0.0	0.4449	1.92	2.1178
S14	0.0	0.0300	6.03	1.2101	7.86	4.7957
S15	0.0	0.0298	1.62	0.4396	2.82	2.2399
S16	0.0	0.0250	7.11	0.4447	7.69	3.2698
S17	0.0	0.0299	4.12	0.4630	6.05	5.5401
S18	0.0	0.0268	6.30	0.6751	9.72	3.8408
S19	0.0	0.0246	1.92	0.5097	0.0	1.9899
S20	0.0	0.0350	6.06	1.1980	14.35	5.5087
S21	0.0	0.0237	5.77	0.4099	8.01	2.7570
S22	0.0	0.0557	6.48	0.8871	7.56	3.5351
S23	0.0	0.0221	6.35	0.5097	7.56	4.0452
S24	0.0	0.0249	6.04	0.9197	8.00	12.6570

As could be expected following the results of Experiment I and NI, there is no improvement found for Data Instance $D1$. Therefore, in the case of this data instance, the order of operators does not matter. From the other results in Table 14 it follows that Experiment $S7$ is best for Data Instance $D6$, Experiment $S10$ is best for Data Instance $D3$, and Experiment $S20$ is close to the best result for both instances $D3$ and $D6$. This is also the case for Experiments $S9$ and $S5$. Unfortunately, each of these orders has one of the highest computation times, however it below six seconds for Data Instance $D6$. Thus, even for larger data sets than Data Instance $D6$, it should be able to compute within a reasonable time. In general, if the computation time increases, the result seem to get better too. With not all data instances used, and only one run per data instance, these results are limited. By using more data instances, more runs, or even a different initial solution, the results can be different.

5.3.4 Adaptiveness of the Variable Neighborhood Search

In all the previous experiments, an Adaptive VNS was used. This means that when the number of iterations without an improvement increases, the shaking phase differs. The adaptiveness is of influence for the final result, as the shaking influences which routes go through the local search phase. Therefore, choosing right shaking method is important. This experiment serves to show the usefulness of adaptive behavior, compared to a single type of shaking. Therefore, Data Instances $D3$, $D5$, and $D6$ will be run, with random initialization, 25 iterations without improvement, and the order of local search as determined using Experiment S. Data Instances $D1$, $D2$, and $D4$ are not used as there were no improvements found during Experiment NI. The experiments will be run with each of the shaking operators, and with the adaption, where every 6 or 7 (25% of the) iterations without improvement the shaking changes to a different operator. In this first Adaptive VNS, the order of shaking operators is Swap - Move - Switch Vehicles - Reverse Route.

Experiment AVNS1.1: Adaptive shaking phase; Swap - Move - Switch Vehicles - Reverse Route
Experiment AVNS1.2: Only using Swap
Experiment AVNS1.3: Only using Move
Experiment AVNS1.4: Only using Switch Vehicles
Experiment AVNS1.5: Only using Reverse Route

Table 15. The average improvement for each of the kinds of shaking is given for each of the data instances. In bold is the best improvement per data instance. The last row is the average of improvements of all the data instances.

Data instance	AVNS1.1	AVNS1.2	AVNS1.3	AVNS1.4	AVNS1.5
D3	9.17	**13.15**	8.19	9.93	11.46
D5	5.13	4.06	**5.70**	5.11	5.38
D6	11.56	10.89	9.90	11.97	**13.39**
average	8.62	9.37	7.93	9.00	**10.23**

The results in Table 15 show that for each of the data instances, Reverse Route improved the solution. However, for two of the data instances a different operator resulted in even better results. In general, from the overall results, it can be concluded that a neighbor generated during shaking that lies further away rather than closer by results in better improvements. This is logical, as the change of leaving a local optimum increases if the neighbor is further away. To determine if the order of the adaptive shaking is of influence Experiment $AVNS2$ will repeat the Adaptive VNS with two different orders. These orders are the increasing and decreasing percentage of improvement from Experiment $AVNS1$.

Experiment AVNS2.1: Increase improvement; Move - Switch Vehicles - Swap - Reverse Route
Experiment AVNS2.2: Decreased improvement; Reverse Route - Swap - Switch Vehicles - Move

Table 16. The average improvement for each of the kinds of shaking is given for each of the data instances. In bold is the best improvement per data instance. The last row is the average of improvements of all the data instances.

Data instance	AVNS2.1	AVNS2.2
D3	**14.11**	9.24
D5	**4.95**	3.56
D6	8.50	**12.27**
average	**9.19**	8.36

The results from Experiment $AVNS2$ are given in Table 16. There is no conclusive result, as it depends on the data set whether and increased or decreased operator improvement strategy results in the biggest improvements, besides the number of experiments is small and different orders could lead to better solutions. Furthermore, these improvements are less than most of the improvements by either just using the Reverse Route operator, or the Adapted Shaking from Experiment $AVNS1$. A second possibility for the adaptive VNS is to use a different number of shakes. As the Reverse Route operator is deemed the best, it will be tested if it matters how many routes are reversed. With the randomness involved, it may be that these shakes lead to infeasible solutions which are not accepted. There is therefore no guarantee that the amount of shakes lead to the same amount of vehicles whose routes are reversed. To ensure that the reversals are not undone due to an even number of reversals Experiment $AVNS3$ will do 1, 3, 7 and 11 route reversals dependent on which iteration it is in. As well as just these amount of route reversals independent of the iteration.

Experiment AVNS3.1: 1 shake
Experiment AVNS3.2: 3 shakes
Experiment AVNS3.3: 7 shakes
Experiment AVNS3.4: 11 shakes
Experiment AVNS3.5: Adapted shaking; 1 - 3 - 7 - 11

Table 17. The average improvement for each of number of shakes is given for each of the data instances. In bold is the best improvement per data instance. The last row is the average of improvements of all the data instances.

Data instance	Amount of shakes				
	AVNS3.1	AVNS3.2	AVNS3.3	AVNS3.4	AVNS3.5
D3	10.31	**13.29**	6.81	11.79	9.89
D5	4.28	2.82	**8.26**	2.93	3.72
D6	14.63	10.93	11.58	**14.74**	10.28
average	9.74	9.01	8.88	**9.82**	7.96

From the results for Experiment $AVNS3$, presented in Table 17, it can be concluded that on average, the bigger the shaking, the better the improvement. Data instance 5 is clearly different in this, but for data instance this also results in the best improvement, and for data instance 3 in the second best improvement. Surprisingly, the adapted version does not perform well. On average it has the worst improvements, which is also the case for data instance 6. For data instance 3, it is the second worst improvement. Once more, data instance 5 is an exception as it performs averagely for this data instance. Combining the results from Experiments $AVNS1, AVNS2 \& AVNS3$ it is decided that during the shaking phase, operator Reverse Route will be called 11 times, regardless of the number of iteration the algorithm finds itself in.

5.3.5 MILP maximum running time

Like the number of iterations for the VNS, the maximum run time of the MILP helps in finding a (near) optimal solution without extensive computational time. Like Experiments NI this experiment serves to find the trade off between result and computational time.

Experiment M1: This experiment will have a short allowable computation time of 60 seconds (1 minute).
Experiment M2: This experiment will have a medium allowable computation time of 600 seconds (10 minutes).
Experiment M3: This experiment will have a large allowable computation time of 3600 seconds (1 hour).

Table 18. The results of the experiments M. In bold the smallest gap per data instance is given.

Data instance	M1			M2			M3		
	Result	Time	Gap	Result	Time	Gap	Result	Time	Gap
D1	183.00	0.4441	0.0	183.00	1.0799	0.0	183.00	0.8848	0.0
D2	283.44	2.0342	0.0	283.44	2.1501	0.0	283.44	2.0697	0.0
D3	311.76	61.5100	0.1088	306.35	600.6143	0.0810	306.35	524.1762	0.0
D4	482.83	27.9464	$2.4*10^{-8}$	482.83	33.2214	$2.4*10^{-8}$	482.83	33.1977	$2.4*10^{-8}$
D5	494.73	61.4339	0.2074	477.68	406.8332	$1.3*10^{-5}$	477.68	463.6435	0.0
D6	-	63.6249	∞	-	603.8501	∞	686.98	3603.7770	0.0645

The results shown in Table 18 indicate that a 600 second maximum already allows the MILP to find (near) optimal solutions for most of the data instances. Data Instance $D6$ is the exception. One of the reasons for this could be that a limited number of feasible solutions. Unlike the VNS, the MILP limits the vehicle use at once, without exception. Therefore, many combinations between locations and vehicles will lead to a result where not all locations can be given their demand with the vehicles used. It is unwanted to wait for the routes for an hour, as the time of volunteers is precious, and they want to know their routes quickly. Thus, the ideal maximum time is 600 seconds (10 minutes). It guarantees a relatively quick and (near) optimal solution in most of the cases.

5.3.6 Final parameters

From these experiments, the final tuning of the VNS and MILP have been determined. Each of the parameter values found from the experiments were applied for the tuning of the next parameter. In order to get results with the data from the food banks, there will be a randomized initial solution, and thus run multiple times. It stops the VNS after 25 iterations without improvement, as this has both a short computation time, and overall found the best improvements. As the order of Switch Vehicles - Swap - Reverse Route - Move seemed to be the order of operators to perform best overall, this order will be used. Furthermore, it is decided not use an adaptive VNS as the results are better using multiple Reverse Route shakes. Therefore, the shaking phase will consist of calling the Reverse Route operator 11 times. For the MILP, the maximum time will be set at 600 seconds, for most cases this will not only lead to a result, it will already be (near) optimal.

For 4 out of the 5 data instances solved to optimality, the VNS finds better or equal results based on the random initialization (Experiment I) alone. For Data Instance $D6$, which is only solved to near optimality, it's result is better than the average of the randomized initialization and nears the minimum found. One of the reasons for this can be that the distances between locations are quite large and thus add a lot of extra costs in the MILP, while the VNS goes into the exception and has a vehicle go from its location to the RDC and back. As there is no penalty counted for this, and this is possible for each vehicle, even if it drove a route already, this can result into significantly less distance.

5.4 Numerical Experiments

In this section the results of the MILP and VNS for the different scenarios are given. The section starts with the MILP results, followed by the VNS results and lastly an overview of the results that are comparable against the current situation.

5.4.1 MILP Results

The MILP was ran with the different scenarios from Table 10. These results are given in Table 19.

Table 19. Results following from the MILP

Scenario	Cost	Time	Gap	Distance	Number of vehicles
2	237.11	3.0893	0.0	823	7
3	259.09	13.1738	$2.19 * 10^{-16}$	866	8
4	369.30	2.1191	0.0	518	4
6	-	163.97	∞	-	-
7	186.68	8.1004	$1.03 * 10^{-7}$	627	6
8	203.20	600.61	0.08	739	6
9	366.07	3.4833	0.0	499	4
11	368.28	2.0804	0.0	512	4
12	-	0.5633	∞	-	-
13	363.69	0.2536	0.0	485	4

The MILP managed to find better results than the current situation for Scenario 7, and 8. For Scenario 2 and 3 there was an optimal result, but this is against more costs than the current situation. This is possible, because in the current situation each food bank combines the different commodities as they only drive a small distance, while this is strictly not allowed in the MILP. More surprisingly is that the MILP cannot find a feasible solution at all for Scenario 12. The resulting routes, for commodity 1 in Scenario 7 are given in Figure 15. As the figures show, the food bank in Raalte will need to travel to the RDC themselves for one of the commodities. The results will change when they are not taken into consideration, as their full demand fits in their vehicle and can transport all at once. However, as they are supposed to bring around commodity 1 to multiple food banks, the resulting routes would change without taking them into consideration. This shows that the routes will have to be carefully analysed, to determine the best possible collaborations. In Appendix D the routes for Scenario 2 and the other commodities for Scenario 7 are given.

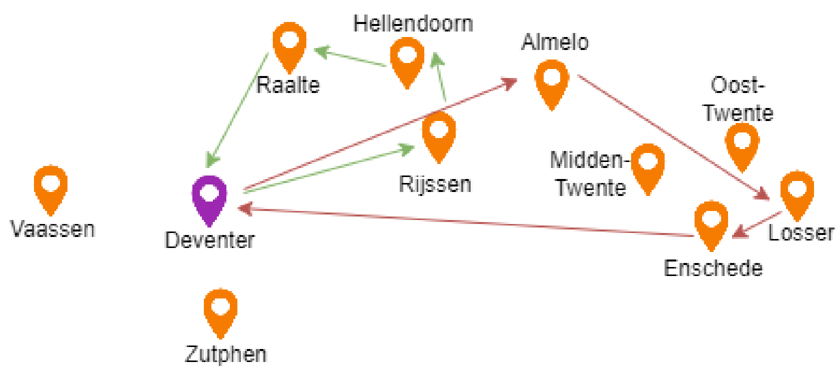

Figure 13. The routes for transporting commodity 1 on Thursday as determined by the MILP.

The scenarios with centralized transportation from the RDC all lead to much higher costs than by collaborating with the current fleet of vehicles. This is due to the large difference in daily costs per vehicle (€70.31 vs €17.66), as the distance driven does become less. If the costs per vehicle can be brought down, e.g. due to sponsoring, this becomes an interesting option. Lastly, for the scenarios where the new vehicles are located at the four biggest food banks it often is not possible to find a feasible solution. The resulting routes for the Thursday is given in Figure 14. The figure shows that the routes are the same as long as the demand fits within the vehicles.

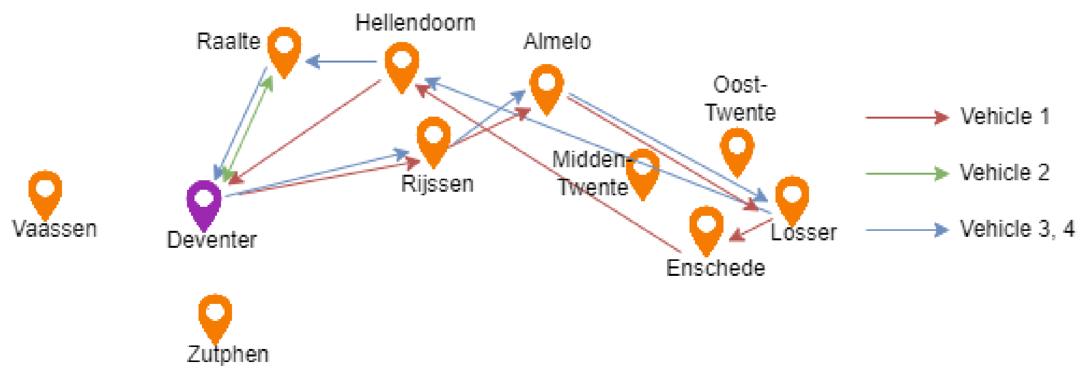

Figure 14. The routes for transport for Scenario 9 generated by the MILP.

The time it takes for the MILP to find a solution, or not, is less than 15 seconds, all scenarios, except Scenario 8. This means that the computation time of the MILP did not form a limitation. The MILP is thus, based on the ability to generate result and the computation time, a valid option.

5.4.2 VNS results

Next, the VNS is ran for each of scenarios, as indicated by Table 10. The results are given in Table 20.

Table 20. Results following from the VNS. The time is the time it takes to perform the VNS. The distance and number of vehicles used are given for the final solution found by the VNS.

Scenario	Initial cost(€)	Result(€)	Improvement(%)	Time(s)	Distance(KM)	Vehicles
2.1	236.99	232.21	2.02	0.1052	743	7
2.2	195.51	190.36	2.63	0.0642	735	6
2.3	179.23	**173.58**	3.16	0.0813	753	5
2.4	210.58	206.88	1.75	0.0797	690	6
2.5	218.06	211.77	2.89	0.0947	724	6
3.1	192.26	185.35	3.60	0.1153	810	6
3.2	189.79	**180.41**	4.94	0.0951	892	6
3.3	191.30	182.92	4.38	0.0844	1027	6
3.4	196.10	182.25	7.06	0.1059	794	6
3.5	231.59	227.31	1.85	0.1238	848	7
6	281.31	**278.42**	1.03	0.0526	550	3
7.1	239.08	239.08	0.0	0.0978	999	8
7.2	239.36	**223.14**	6.77	0.1290	887	7
7.3	228.17	227.61	0.25	0.1343	902	7
7.4	239.45	225.66	5.76	0.1338	891	7
7.5	230.52	228.79	0.75	0.1379	875	7
8.1	243.59	**238.55**	2.07	0.3508	1043	7
8.2	292.48	276.40	5.50	0.2928	1098	8
8.3	252.23	244.67	3.00	0.2590	1073	8
8.4	279.62	276.54	1.10	0.2454	1110	9
8.5	247.00	240.80	2.51	0.2535	1183	8
11	-	-	-	-	-	-
12.1	104.99	**104.99**	0	0.0350	736	3
12.2	140.63	139.37	0.90	0.03847	822	4
12.3	104.99	104.99	0	0.0312	836	3
12.4	104.99	104.99	0	0.0361	831	3
12.5	140.63	139.37	0.90	0.0394	722	4

The VNS is able to find a better result than the current situation for every scenario, except Scenario 7. This is due to needing less vehicles, as the distances driven do increase in these cases. Furthermore, only for the scenarios with the new vehicles, there is only one scenario (Scenario 6) for which a result is found. It is surprising that for Scenario 11 no feasible solution was found. Therefore it was checked if ordered initial solution was able to find a feasible solution. This was not the case, and thus it was concluded that the VNS is limiting. Part of this limitation is not allowing a vehicle belonging to another food bank to drive solely to satisfy the demand of another food bank. This limitation is playing its part in not being able to find a feasible solution in Scenario 11. Lastly, each of the scenarios for which a result can be found is solved within a second. Therefore the time did not form a limitation.

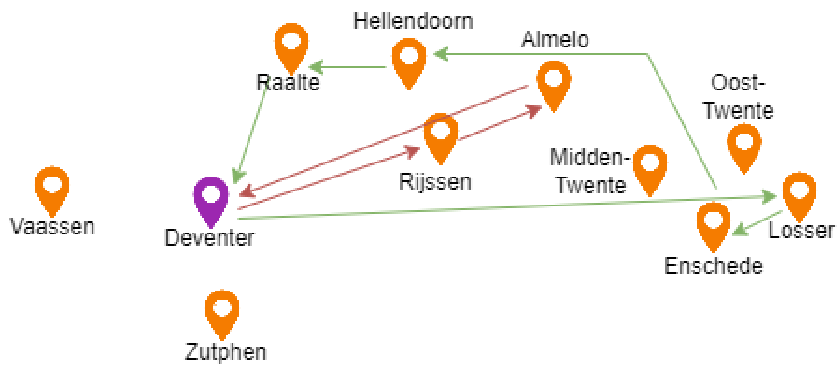

Figure 15. The routes for transporting commodity 1 on Thursday as determined by the VNS.

Figure 15 shows the routes as determined by the VNS for commodity 1 on the Thursday, the other routes are given in Appendix D. These show that a combination of the current situation and a collaboration may be an even better solution. This is because in this case the food bank in Losser has to drive for commodity 2 itself. As their vehicle is large enough to get their full demand in one trip, it may be even cheaper for them to go by themselves, while the other food banks collaborate. It will need to be checked how feasible this is with regards to the vehicles used.

5.4.3 Overview of the results

Table 21. Overview of the results for the scenarios comparable to the current situation.

Scenario	Original(€)	MILP (€)	Improvement (%)	VNS (€)	Improvement (%)
2	226.09	237.11	-4.87	173.58	23.23
3	243.41	259.09	-6.44	180.41	25.88
7.1	203.35	186.68	8.20	223.14	-9.73
7.2	203.35	186.68	8.20	186.14	8.46
8	246.10	203.20	17.43	238.55	3.07
12	137.97	-	-	104.99	23.90

In Table 21 the results have been summarized by giving the original cost, as well as the costs determined by the MILP and VNS. Besides, the improvement, in percentage, has been given. For Scenario 7 there are two results given. The first one follows from the results in Table 20. In the second, the VNS was given more iterations without improvement. This resulted in a lower cost than the MILP was able to find. For Scenario 8 the number of iterations were increased too, however, this did not result in a better possible result than the MILP gave. The MILP and VNS both deal with their own limitations in finding the most cost-efficient routing. Thus it occasionally happens that they do not perform better than the original situation, or even have the VNS outperform the MILP which is supposed to solve to optimality. For example, the MILP is strict in the usage of vehicles, which might result in using an extra vehicle against the cost of the vehicle, while the VNS repeats the use of a vehicle for a single drop off, adding only some distance. The other way around the VNS uses a vehicle to supply the RDC, while this is not taken into consideration in the MILP. The results thus cannot be compared one on one, but only in the general picture. This also means that it is indeed possible find a better result with the use of the VNS than the MILP which solved to optimality. In general, from these results it can be said that it would be financially more interesting to find a collaboration using the existing fleet of vehicles than to invest in larger, more expensive vehicles. Especially since the vehicles are also used for the local trips to supermarkets, bakeries and local food producers and therefore the non-variable costs remain. Furthermore, the VNS is not able to find a solution for the scenarios with centralized transportation. However, the MILP is able to find results for each of the scenarios with the transport being centrally organized. Besides, most scenarios use all vehicles, or include trips for single food banks as they could not be added to one of the routes without exceeding capacity and/or time window constraints. This means that certain vehicles have to make multiple trips within the same day, which is unwanted. Besides, the costs of the current vehicles are expected to rise significantly, and thus it is questionable whether the new vehicles continue to be that much more expensive. All-in-all it is clear that there is room to improve the transportation between the RDC and the food banks, but there is not one definite method to find a better routing, due to the simplification of the MILP and limitations of the VNS.

5.5 Conclusion

This chapter began with the fine tuning of the parameters of both the MILP and VNS, based on six different data instances. These data instances were smaller, equal in size, and larger than the real world scenarios. Based on the results from the parameter fine tuning, the parameter settings for the VNS and MILP were determined. For instance, it is determined to use the regular VNS and not an adaptive version. These settings were used to find the results in the numerical experiments. Next, the first sub question *Which are the different scenarios under which we test the solution approach?* was answered by determining the scenarios and data used. There were a total of 13 scenarios, of which 11 were run by the MILP and/or VNS. The scenarios consisted of a collaboration with the use of the existing vehicle fleet, with or without the food banks of Vaassen and Zutphen. Besides, there were scenarios with the transport centralized where the start and end points of the vehicles is at the RDC or where the new vehicles are located at the four largest food banks. These latter scenarios did not include the food banks in Vaassen and Zutphen as they have notified the rest of the region of their lack of interest in collaborating. With the scenarios and data ready, the MILP and VNS were used to find solutions to the transportation problem of the food banks. The results of the numerical experiments of these scenarios are summarized in Table 21. With these results *How does the solution approach perform under the different scenarios considered?* is answered. The sub questions together can answer *How does the solution approach compare to the current situation?*. For each of the scenarios either the MILP or VNS finds a better result than the current solution. Thus, it became clear that a collaboration between the food banks with regard to transportation is wanted. However, it cannot be said that either the MILP or the VNS is better than the other, as it was dependent on the scenario. Therefore it is up to the food banks to make a decision on how to arrange such a collaboration.

6 Conclusions and Recommendations

In this chapter the research findings are summarized and recommendations are given regarding the outcome of the research as well as future research.

6.1 Conclusion

For this research, the transport of food between the RDC and food banks in the region of Twente-Salland was explored. The aim was to find a way to reduce the costs of this transport, as it makes up a sixth of the total budget. For that reason a MILP and VNS were developed to optimize the routing between the food banks. Both these methods took several constraints into consideration, such as the opening hours of the food banks, the capacity of the vehicles, and the different products (commodities) to be transported. The data has been provided by the local food banks, and created with the use of Google maps.

To ensure the generality, and applicability in other regions, the MILP and VNS are fine tuned with the use of six data instances of smaller, similar and larger number of locations compared to the region of Twente-Salland. This helps determine if larger regions will also be able to generate a collaborative routing. These data instances underwent experiments to determine the maximum allowed computation time (for the MILP) or the method of initialization, number of iterations without improvement, order of operators in the local search, and the amount of shaking for the VNS. With the best combination of settings found, the numerical experiments could take place. These results show that using the MILP or VNS lowers the costs up to 25.88%. Furthermore, it was checked whether it would be feasible to centralize the transport, with vehicles situated at the RDC or new vehicles at the four largest food banks would be useful. The VNS was most often not able to generate any feasible routing, while the MILP only found results for the centralized transport that were higher than a collaboration with the existing vehicles, or current situation. To conclude, the food banks can optimize their transportation between them and the RDC, however there is not one clear method to do so.

6.2 Recommendations

Based on the results from the MILP and VNS it can be recommended that the food banks should work together with the vehicles they currently own. These costs can, with the current lease and fuel costs, be reduced by up to 25.88%. The costs of the new vehicles are that much higher than their current vehicles, that the decrease in distance travelled is not compensated. The costs increase between 9.73% and 163.60%. With the exception of one scenario where a decrease in costs was found. By having timely communication with the RDC and amongst one another, the food banks should be able to plan suitable routing that leads to lower costs. However, it needs to be considered that the cost of fuel is increasing significantly, and the lease (or maintenance) of the current vehicles is likely to increase too. It could therefore be, that the difference in costs as currently found will be significantly less. Furthermore, taking into consideration the KPIs of distance driven and number of vehicles used, the scenarios with the new vehicles outperform a collaboration with the existing fleet. However, the smaller vehicles are still to be used for the local trips to supermarkets, bakeries and local producers. Thus their lease/standard costs will remain even when new vehicles are used for the trips to and from the RDC. It is thus up to the food banks themselves to consider this tradeoff, also taking the limitations of the MILP and VNS into consideration.

6.2.1 Limitations and Future Work

As the numerical experiments only use the average amount of demand on specific week days, and the fraction of demand for each commodity has been assumed, it may be the case the there are days where these numbers are completely different. The found routes may then be infeasible, and thus it remains needed for the food banks to determine the right routing for every day. Moreover, the MILP does not take the supply into consideration, based on the assumption that this is less than the demand and thus will fit the vehicle. By including this the MILP will give a better impression of the possible routes and related costs. Within the VNS it would be better to include the supply as part of the routes instead of a separate route which takes a vehicle that cannot be used for the demand. Thus, the data, the MILP and VNS can all be improved to take the supply better into consideration. For the food banks this means it is important to start keeping track of the supply they deliver to the RDC, which should also track this. This makes it comparable and less sensitive to errors. Furthermore, the MILP and VNS should be updated such that these limitations no longer exist, by extending and updating of route creation respectively. Updating the route creation of the VNS will also help improve the allocation of the vehicles. Possibly this leads to the ordered initialization outperforming the random initialization. The experiments have shown that the initial solution limits the possible outcomes of the VNS and thus also the possible results. This does not directly solve the issue of average demand being used. This can be solved by altering the data files, and have the demand correspond with known data from different weeks, such that more scenarios are covered.

Furthermore, the test data instances were only used to test certain types of scenarios. Had the test data instances been used to cover all types of scenarios tested with the actual data could have led to changes in the VNS and MILP allowing more scenarios to be able to find a feasible solution. This must be taken into consideration when updating the VNS and MILP. Not only could the testing have been more extensive, the analysis of the results from these experiments was limited. No statistical analysis was performed, even though randomness was involved. There is no guarantee that the five results found are covering the full solution space. Adding a statistical analysis will give even more insights into the found results.

Besides the limitations in the MILP and VNS, there are limitations in the data. The costs for driving a kilometer are based on the costs and travelled distances from 2021. However, since then the average price for a liter of diesel has increased by 26,5% (Shell, 2022). If the cost of fuel continues to increase, the extra distance travelled may come at such extensive costs that the lease of an extra, or more expensive, vehicle may be worth it. For fairer comparisons, the data used should all be from the same year, such that, amongst others, inflation does not influence the results. Another, preferable, option is to normalize the known data.

As hinted in Section 2.3, there should be a fair cost division between the food banks. This has not been taken into consideration, so it may happen that all vehicles from a food bank are being used, while others are not. It is important for the food banks to find a way to divide the costs, such that a collaboration is fair to all of them. It is needed to look into this, as getting rid of costs is one of the incentives from the food banks to consider a collaboration. For instance, the Shapley value can be used to determine whether this collaboration is advantageous for all. In general, game theory can be used to help with the forming and maintaining of the collaboration.

Lastly, the MILP and VNS created are not user friendly. The input data is time consuming to change, especially for the MILP where the locations are copied multiple times to correspond to each commodity, and prevent different commodities within the same vehicle. It is therefore advised against using the codes for the MILP and VNS as they are. In the future a tool, which is user friendly, can be developed such that routes can be generated daily. Thus being able to generate routes that are feasible regardless of the demand that day. Furthermore, by having such a tool, the planning can be made centrally. This is advantageous as the RDC has the full information regarding demand, where the food banks themselves simply do not have this information available at all times.

6.3 Contribution to Theory and Practice

This research has several contributions to both the theory and the practice. First, the contribution to theory is evaluated, after which the evaluation of the practice takes place.

6.3.1 Contribution to theory

Research in the field of transportation of goods for the food bank has been limited. There has been research regarding the supply chain of food banks in other countries, but these food banks have a different method of working than food banks in the Netherlands. In the Netherlands the food banks collect food to give to their clients directly, while in the found literature the food banks collect food as an intermediary and forward the food to other charitable organizations. This research took an in-depth look into the transport between depot and customer in a network of food banks in the region of Twente-Salland. Furthermore, a MILP has been introduced to solve the multi-commodity VRP with time windows and heterogeneous fleet of vehicles. This variant in itself is not new. However, specifically noticeable is the fact that the vehicles are not located at the depot, but at the customer. This, to the best of my knowledge, had not been done before. Besides not all of the vehicles have to be used.

6.3.2 Contribution to practice

In practice this research contributed valuable insights into the transportation flows between the RDC and food banks. Using different methods, it became clear that there is room for improvement in these transportation flows, when the food banks collaborate on transport.

These insights can help the logistic coordinators of the food banks in the region of Twente-Salland with the decision making surrounding the transport of goods. The set up of this research is general, so it can also be used by other regions in the Netherlands.

References

Aksoy, Ö., & Kapanoglu, M. (2012, 07). Multi-commodity, multi-depot, heterogenous vehicle pickup and delivery problem for air transportation in the turkish air force. , 53-57.

Battarra, M., Monaci, M., & Vigo, D. (2009). An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research*, *36*(11), 3041-3050. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0305054809000410` doi: https://doi.org/10.1016/j.cor.2009.02.008

Bazerghi, C., McKay, F. H., & Dunn, M. (2016). The role of food banks in addressing food insecurity: A systematic review. *J Community Health*, *41*, 732-740. doi: 10.1007/s10900-015-0147-5

Belgin, O., Karaoglan, I., & Altiparmak, F. (2018). Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, *115*, 1-16. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0360835217305193` doi: https://doi.org/10.1016/j.cie.2017.10.032

Brock, L. G., & Davis, L. B. (2015). Estimating available supermarket commodities for food bank collection in the absence of information. *Expert Systems with Applications*, *42*(7), 3450-3461. Retrieved from `https://www.sciencedirect.com/science/article/pii/S095741741400774X` doi: https://doi.org/10.1016/j.eswa.2014.11.068

Campbell, C. C. (1991). Food insecurity: A nutritional outcome or a predictor variable? *The Journal of Nutrition*, *121*(3), 408-415. doi: https://doi.org/10.1093/jn/121.3.408

Cattaruzza, D., Absi, N., Feillet, D., & Vidal, T. (2014). A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research*, *236*(3), 833-848. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0377221713005006` doi: https://doi.org/10.1016/j.ejor.2013.06.012

Cattaruzza, D., Absi, N., Feillet, D., & Vigo, D. (2014). An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. *Computers & Operations Research*, *51*, 257-267. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0305054814001646` doi: https://doi.org/10.1016/j.cor.2014.06.006

Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, *20*(1), 255-282.

Crainic, T. G., Ricciardi, N., & Storchi, G. (2004). Advanced freight transportation systems for congested urban areas. *Transportation Research Part C: Emerging Technologies*, *12*(2), 119-137. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0968090X04000142` doi: https://doi.org/10.1016/j.trc.2004.07.002

Crainic, T. G., Ricciardi, N., & Storchi, G. (2009). Models for evaluating and planning city logistics systems. *Transportation Science*, *43*(4), 432-454. Retrieved from `https://doi.org/10.1287/trsc.1090.0279` doi: 10.1287/trsc.1090.0279

Crevier, B., Cordeau, J.-F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*(2), 756-773. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221705006983 doi: https://doi.org/10.1016/j.ejor.2005.08.015

Cuda, R., Guastaroba, G., & Speranza, M. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, *55*, 185-199. Retrieved from https://www.sciencedirect.com/science/article/pii/S030505481400166X doi: https://doi.org/10.1016/j.cor.2014.06.008

Dantzig, G., & Ramser, J. (1959). The truck dispatching problem. *Management Science*, *6*, 80-91.

Davis, L. B., Sengul, I., Ivy, J. S., Brock, L. G., & Miles, L. (2014). Scheduling food bank collections and deliveries to ensure food safety and improve access. *Socio-Economic Planning Sciences*, *48*(3), 175-188. Retrieved from https://www.sciencedirect.com/science/article/pii/S0038012114000172 doi: https://doi.org/10.1016/j.seps.2014.04.001

den Hurk, V. (n.d.). *Opel movano 2.3 cdti 136 pk bakwagen.* Retrieved 2022-04-28, from https://www.vdhurk.nl/voorraad/20904751-opel-movano-23-cdti-136-pk-bakwagen

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, *40*(2), 342-354.

Distrifood. (2021). *Marktaandelen 2008-2020.* Retrieved 08-12-2021, from https://www.distrifood.nl/food-data/marktaandelen

Duhamel, C., Potvin, J.-Y., & Rousseau, J.-M. (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science*, *31*(1), 49-59. doi: http://dx.doi.org/10.1287/trsc.31.1.49

Duk Song, B., & Dae Ko, Y. (2016). A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery. *Journal of Food Engineering*, *169*, 61-71. Retrieved from https://www.sciencedirect.com/science/article/pii/S0260877415003805 doi: https://doi.org/10.1016/j.jfoodeng.2015.08.027

Eisenhandler, O., & Tzur, M. (2019). The humanitarian pickup and distribution problem. *Operations Research*, *67*(1), 10-32.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, *57*(4), 1472-1483. doi: 10.1016/j.cie.2009.05.009

Euchi, J. (2020). Hybrid adaptive memory programming to optimise the multi-commodity many to many vehicle routing problem. *International Journal of Mathematics in Operational Research*, *17*(4), 492-513. Retrieved from https://www.inderscienceonline.com/doi/abs/10.1504/IJMOR.2020.110840 doi: 10.1504/IJMOR.2020.110840

Fallahi, A. E., Prins, C., & Wolfler Calvo, R. (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, *35*(5), 1725-1741. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054806002516 doi: https://doi.org/10.1016/j.cor.2006.10.006

Feo, T., & Resende, M. (1995). Greedy randomized adaptive search procedures. *J Glob Optim*, *6*, 109-133. Retrieved from https://link.springer.com/article/10.1007/BF01096763 doi: https://doi.org/10.1007/BF01096763

Gjenstø, A., & Vaksvik, S. (2008). *Single vehicle pickup and delivery problem with multiple commodities and visit windows* . Molde University College.

Goeke, D. (2018). *goeke 2018.* VRP-REP.

Goetschalkx, M., & Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls. *European Journal of Operational Research*, *42*(1). doi: 10.1016/0377-2217(89)90057-x

Hamelin, A.-M., Habicht, J.-P., & Beaudry, M. (1999). Food insecurity: Consequences for the household and broader social implications. *American Society for Nutritional Sciences*.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, *130*(3), 449-467. doi: https://doi.org/10.1016/S0377-2217(00)00100-4

Hansen, P., & Mladenović, N. (2005). Variable neighborhood search. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: Introductory tutorials in optimization and decision support techniques* (pp. 211–238). Boston, MA: Springer US. Retrieved from `https://doi.org/10.1007/0-387-28356-0_8` doi: 10.1007/0-387-28356-0_8

Hau, A., Fang, F., & Shi, Z. R. (2021, June). Poster: Pallet estimation for food bank logistics management. In *Compass '21: Acm sigcas conference on computing and sustainable societies* (p. 385-388).

Ipskamp, M. (2021). *Centralizing the logistics of de voedselbank* . University of Twente.

Irnich, S. (2000). A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. *European Journal of Operational Research*, *122*(2), 310-328. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0377221799002350` doi: https://doi.org/10.1016/S0377-2217(99)00235-0

Jayarathna, N., Lanel, J., & Juman, Z. (2020). Five years of multi-depot vehicle routing problems. , *5*(2), 109-123. doi: http://dx.doi.org/10.14254/jsdtl.2020.5-2.10

Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). Vehicle routing problem with time windows.
doi: 10.1007/0-387-25486-2_3

Kevin, K., Aritonang, Y. K., & Lesmono, J. D. (2019). A model to choose hub and route determination. *Jurnal Teknik Industri*, *20*(2), 182-190. doi: https://doi.org/10.22219/JTIUMM

Kramer, G. (n.d.). *Renault master.* Retrieved 2022-04-28, from `https://garagekramer.nl/occasion/renault-master-t35-2-3-dci-l3h2/`

Kuo, Y., & Wang, C.-C. (2012). A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications*, *39*(8), 6949-6954. doi: https://doi.org/10.1016/j.eswa.2012.01.024.

Lahyani, R., Khemakhem, M., & Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, *241*(1), 1-14. doi: 10.1016/j.ejor.2014.07.048

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics (NRL)*, *54*(8), 811-819. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20261` doi: https://doi.org/10.1002/nav.20261

Lin, S.-W., Yu, V. F., & Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, *36*(5), 1683-1692. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054808000762 doi: https://doi.org/10.1016/j.cor.2008.04.005

Martins, C., Melo, M., & Pato, M. (2019). Redesigning a food bank supply chain network in a triple bottom line context. *International Journal of Production Economics*, *214*, 234-247. Retrieved from https://www.sciencedirect.com/science/article/pii/S092552731830450X doi: https://doi.org/10.1016/j.ijpe.2018.11.011

Moshref-Javadi, M., & Lee, S. (2016). The customer-centric, multi-commodity vehicle routing problem with split delivery. *Expert Systems with Applications*, *56*, 335-348. Retrieved from https://www.sciencedirect.com/science/article/pii/S095741741630121X doi: https://doi.org/10.1016/j.eswa.2016.03.030

Naccache, S., Côté, J.-F., & Coelho, L. C. (2018). The multi-pickup and delivery problem with time windows. *European Journal of Operational Research*, *269*(1), 353-362. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221718300717 doi: https://doi.org/10.1016/j.ejor.2018.01.035

Nair, D., Grzybowska, H., Fu, Y., & Dixit, V. (2018). Scheduling and routing models for food rescue and delivery operations. *Socio-Economic Planning Sciences*, *63*, 18-32. Retrieved from https://www.sciencedirect.com/science/article/pii/S0038012116302294 doi: https://doi.org/10.1016/j.seps.2017.06.003

Neter, J. E., Dijkstra, S. C., Visser, M., & Brouwer, I. A. (2014). Food insecurity among dutch food bank recipients: a cross-sectional study. *BMJ Open*. doi: 10.1136/bmjopen-2013-004657

Paul, A., Kumar, R. S., Rout, C., & Goswami, A. (2021). A bi-objective two-echelon pollution routing problem with simultaneous pickup and delivery under multiple time windows constraint. *OPSEARCH*, *58*, 962-993.

Rafiei, H., Rabbani, M., Vafa-Arani, H., & Khoshnudi, N. (2013). A heterogeneous fleet vehicle waste collection problem with various zones and intermediate facilities. *International Journal of Engineering Sciences and Research Technology*, *2*(5), 1251-1259.

Ramirez-Villamil, A., Jaegler, A., & Montoya-Torres, J. R. (2021). Sustainable local pickup and delivery: The case of paris. *Research in Transportation Business & Management*, 100692. Retrieved from https://www.sciencedirect.com/science/article/pii/S2210539521000754 doi: https://doi.org/10.1016/j.rtbm.2021.100692

Ramos, T. R. P., Gomes, M. I., & Póvoa, A. P. B. (2020). Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*, *23*(2), 103-120. doi: 10.1080/13675567.2019.1630374

Rieck, J., Ehrenberg, C., & Zimmermann, J. (2014). Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. *European Journal of Operational Research*, *236*(3), 863-878. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221713010059 (Vehicle Routing and Distribution Logistics) doi: https://doi.org/10.1016/j.ejor.2013.12.021

Rienstra, W. (2021). *Choosing between centralized and decentralized logistics for food banks in the region twente-salland* . University of Twente.

Rijksoverheid. (2021). *Welk rijbewijs heb ik nodig om in een auto te rijden?* Retrieved 15-11-2021, from https://www.rijksoverheid.nl/onderwerpen/rijbewijs/vraag-en-antwoord/rijbewijs-auto

Ríos-Mercado, R., López-Pérez, F., & Castrillón-Escobar, A. (2013, 09). A grasp for a multi-depot multi-commodity pickup and delivery problem with time windows and heterogeneous fleet in the bottled beverage industry. In (Vol. 8197, p. 143-157). doi: 10.1007/978-3-642-41019-2_11

Savelsbergh, M., & M.Sol. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17-29. doi: https://doi.org/10.1287/trsc.29.1.17

Sharma, R., & Saini, S. (2020). Heuristics and meta-heuristics based multiple depot vehicle routing problem: A review. In *2020 international conference on electronics and sustainable communication systems (icesc)* (p. 683-689). doi: 10.1109/ICESC48915.2020.9155814

Shell. (2022). *Historisch prijzenoverzicht.* Retrieved 2022-04-28, from https://www.shell.nl/consumenten/shell-fuels/historisch-prijzenoverzicht.html

Tansini, L., Urquhart, M., & Viera, O. (2001). Comparing assignment algorithms for the multi-depot vrp. *Reportes Técnicos*, *01*(08).

Toth, P., & Vigo, D. (2002). *The vehicle routing problem.* Monographs on Discrete Mathematics and Applications 9, Siam.

Voedselbank Almelo. (2021). *Statistieken, de cijfers van de voedselbank almelo.* Retrieved 01-12-2021, from https://www.voedselbankalmelo.nl/

Voedselbanken Nederland. (2021a). *Armoede in nederland.* Retrieved 08-11-2021, from https://voedselbankennederland.nl/artikelen/armoede-in-nederland/

Voedselbanken Nederland. (2021b). *Feiten en cijfers voedselbanken nederland - 2020.* Retrieved 10-11-2021, from https://voedselbankennederland.nl/wp-content/uploads/2021/03/DEF_Feiten_en_Cijfers_per_31-12-2020.pdf

Voedselbanken Nederland. (2021c). *Missie 538 voor de voedselbank.* Retrieved 20-12-2021, from https://voedselbankennederland.nl/missie-538/

Voedselbanken Nederland. (2021d). *Missie en visie voedselbanken.* Retrieved 08-11-2021, from https://voedselbankennederland.nl/wat-we-doen/missie-en-visie-voedselbanken/

Voedselbanken Nederland. (2022). *Zijn giften fiscaal aftrekbaar?* Retrieved from https://voedselbankennederland.nl/ufaqs/zijn-giften-fiscaal-aftrekbaar/

Wasner, M., & Zäpfel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, *90*(3), 403-419. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925527303003906 (Production Control and Scheduling) doi: https://doi.org/10.1016/j.ijpe.2003.12.002

Yang, X., Bian, Y., Bostel, N., & Dejax, P. (2019). A two-phase method for solving the hub location and routing problem. *4th IEEE International Conference on Cybermetics (Cybconf)*, 1-6. doi: 10.1109/CYBCONF47073.2019.9436586

Yazgı Tütüncü, G., Carreto, C. A., & Baker, B. M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, *37*(1), 138-154. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305048306001356 doi: https://doi.org/10.1016/j.omega.2006.11.001

Zhang, G., Habenicht, W., & Spieß, W. E. L. (2003). Improving the structure of deep frozen and chilled food chain with tabu search procedure. *Journal of Food Engineering*, *60*, 67-79. doi: 10.1016/S0260-8774(03)00019-0

Zhou, H., Qin, H., Zhang, z., et al. (2022). Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery. *Soft Comput*, *26*, 3345-3360. Retrieved from https://doi.org/10.1007/s00500-021-06712-2

Çağrı Koç, Laporte, G., & İlknur Tükenmez. (2020). A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, *122*, 104987. Retrieved from https://www.sciencedirect.com/science/article/pii/S0305054820301040 doi: https://doi.org/10.1016/j.cor.2020.104987

Appendices

A Experiment NI results

Tables 22, 23 and 24 give the results, per data set, per run, for Experiments $NI1$, $NI2$ and $NI3$. These were used to determine the number of iterations without improvement.

Table 22. The results of the experiment NI1

Dataset	Initial cost	Cost			Time(s)		
		Minimum	Average	Maximum	Minimum	Average	Maximum
1.1	183.01	183.01	183.01	183.01	0.0060	0.0013	0.0117
1.2	183.01	183.01	183.01	183.01	0.0080	0.0016	0.0094
1.3	183.01	183.01	183.01	183.01	0.0015	0.0016	0.0081
1.4	147.47	147.47	147.47	147.47	0.0036	0.0030	0.0152
1.5	183.01	183.01	183.01	183.01	0.0065	0.0016	0.0096
2.1	210.52	210.52	210.52	210.52	0.0221	0.0254	0.0301
2.3	210.52	210.52	210.52	210.52	0.0210	0.0220	0.0231
2.3	210.52	210.52	210.52	210.52	0.0213	0.0226	0.0239
2.4	210.52	210.52	210.52	210.52	0.0230	0.0312	0.0418
2.5	210.52	210.52	210.52	210.52	0.0240	0.0260	0.0286
3.1	404.58	370.78	370.78	370.78	0.1292	0.1606	0.2687
3.2	371.40	340.74	340.74	340.74	0.1407	0.1650	0.2364
3.3	357.07	323.63	323.63	323.63	0.1131	0.1406	0.2023
3.4	384.21	316.67	316.67	316.67	0.1155	0.1355	0.1717
3.5	331.27	263.93	263.93	263.93	0.1120	0.1545	0.3199
4.1	513.45	513.45	513.45	513.45	0.1219	0.1311	0.1407
4.2	505.10	505.10	505.10	505.10	0.1105	0.1272	0.1772
4.3	467.18	467.18	467.18	467.18	0.1131	0.1187	0.1251
4.4	519.92	519.92	519.92	519.92	0.1160	0.1228	0.1326
4.5	519.92	519.92	519.92	519.92	0.1156	0.1245	0.1423
5.1	537.05	502.89	511.80	514.05	0.2470	0.3022	0.4005
5.2	537.71	520.57	520.57	520.57	0.2522	0.3014	0.4267
5.3	572.38	543.49	543.49	543.49	0.2110	0.2268	0.2767
5.4	619.60	583.60	583.60	583.60	0.2617	0.3046	0.4383
5.5	459.81	448.56	453.03	459.72	0.2245	0.3161	0.4868
6.1	705.93	545.01	545.01	545.01	0.5580	0.7170	1.3168
6.2	864.09	678.97	678.97	678.97	0.6656	0.8986	1.8275
6.3	688.27	557.35	557.35	557.35	0.6700	0.9114	1.8502
6.4	774.27	654.26	654.26	654.26	0.6783	0.7342	1.3671
6.5	737.64	555.59	555.59	555.59	0.5937	0.7342	1.2665

Table 23. The results of the experiment NI2

Dataset	Initial cost	Cost			Time		
		Minimum	Average	Maximum	Minimum	Average	Maximum
1.1	167.70	167.70	167.70	167.70	0.0189	0.0194	0.0208
1.2	183.01	183.01	183.01	183.01	0.0180	0.0182	0.0190
1.3	183.01	183.01	183.01	183.01	0.0180	0.0192	0.0202
1.4	183.01	183.01	183.01	183.01	0.0200	0.0209	0.0224
1.5	183.01	183.01	183.01	183.01	0.0200	0.0238	0.0335
2.1	210.52	210.52	210.52	210.52	0.0513	0.0579	0.0708
2.2	210.52	210.52	210.52	210.52	0.0552	0.0633	0.0770
2.3	210.52	210.52	210.52	210.52	0.0540	0.0558	0.0592
2.4	210.52	210.52	210.52	210.52	0.0545	0.0562	0.0578
2.5	194.82	194.82	194.82	194.82	0.0512	0.0528	0.0550
3.1	305.06	264.07	264.07	264.07	0.2590	0.2733	0.3243
3.2	394.60	346.18	347.26	348.29	0.3211	0.4627	0.7407
3.3	273.32	223.50	223.50	223.50	0.2882	0.3078	0.3395
3.4	373.21	267.97	271.86	277.70	0.2816	0.3911	0.5776
3.5	366.63	293.60	293.60	293.60	0.3247	0.4218	0.6739
4.1	517.51	517.51	517.51	517.51	0.2801	0.2915	0.3030
4.2	517.55	517.55	517.55	517.55	0.2601	0.2694	0.2955
4.3	467.21	467.21	467.21	467.21	0.2741	0.2782	0.2819
4.4	513.42	513.42	513.42	513.42	0.2761	0.2799	0.2830
4.5	519.92	519.92	519.92	519.92	0.2741	0.2788	0.2865
5.1	600.28	566.75	566.75	566.75	0.5314	0.5939	0.7863
5.2	546.87	504.04	519.83	523.78	0.5285	0.6336	0.9214
5.3	535.16	489.69	489.69	489.9	0.6904	0.8558	1.3948
5.4	491.98	472.22	475.68	480.91	0.5696	0.7646	1.1249
5.5	627.27	601.73	601.73	601.73	0.6178	0.6510	0.6795
6.1	660.14	553.23	553.23	553.23	1.3017	1.5404	1.9923
6.2	772.87	587.53	587.53	587.53	1.7885	2.0837	2.9277
6.3	804.82	647.79	647.79	647.79	1.6138	2.4381	4.0325
6.4	741.34	614.11	614.11	614.11	1.8992	2.2958	3.5940
6.5	778.34	738.61	738.61	738.61	1.8391	2.2310	3.5152

Table 24. The results of the experiment NI3

Dataset	Initial cost	Cost			Time		
		Minimum	Average	Maximum	Minimum	Average	Maximum
1.1	183.01	183.01	183.01	183.01	0.0290	0.0292	0.0300
1.2	183.01	183.01	183.01	183.01	0.0300	0.0303	0.0311
1.3	110.78	110.78	110.78	110.78	0.0290	0.0300	0.0309
1.4	147.47	147.47	147.47	147.47	0.0301	0.0328	0.0380
1.5	183.01	183.01	183.01	183.01	0.0280	0.0286	0.0290
2.1	202.79	202.79	202.79	202.79	0.0808	0.0830	0.0865
2.2	210.52	210.52	210.52	210.52	0.0823	0.0872	0.0900
2.3	210.52	210.52	210.52	210.52	0.0893	0.0946	0.1104
2.4	210.52	210.52	210.52	210.52	0.0810	0.0858	0.0899
2.5	210.52	210.52	210.52	210.52	0.0790	0.0801	0.0827
3.1	410.19	363.27	363.27	363.27	0.4639	0.5687	0.9768
3.2	381.16	345.25	345.25	345.25	0.4933	0.5293	0.6377
3.3	281.28	192.34	192.34	192.34	0.3322	0.4896	0.8331
3.4	366.21	333.29	333.29	333.29	0.5883	0.6412	0.7542
3.5	340.15	290.78	290.78	290.78	0.4318	0.4688	0.4961
4.1	477.42	477.42	477.42	477.42	0.4494	0.4654	0.4896
4.2	568.36	568.36	568.36	568.36	0.4380	0.4541	0.4814
4.3	496.43	496.43	496.43	496.43	0.4218	0.4324	0.4500
4.4	515.31	515.31	515.31	515.31	0.5256	0.5392	0.5536
4.5	467.21	467.21	467.21	467.21	0.4806	0.5380	0.5727
5.1	569.63	540.61	540.61	540.61	0.7486	0.7883	0.8237
5.2	643.96	604.42	604.42	604.42	0.9069	1.0444	1.1263
5.3	534.30	517.94	521.21	534.30	0.9366	1.0836	1.5498
5.4	535.13	506.11	506.11	506.11	0.9671	1.0777	1.2911
5.5	546.87	507.41	510.69	523.78	0.8318	0.9909	1.6008
6.1	794.24	661.62	661.62	661.62	2.6513	3.2699	3.8769
6.2	704.27	613.52	618.95	631.10	2.6116	3.5762	4.7164
6.3	698.89	566.72	566.72	566.72	2.7679	4.0688	8.8030
6.4	840.96	606.17	606.17	606.17	3.0792	3.5462	4.8627
6.5	726.3038	569.61	569.61	569.61	2.2897	2.6890	3.7170

B Experiment AVNS results

Tables 25, 26 and 27 contain the full results of Experiments determining the adaptiveness of the VNS. The experiments are explained and summarized results are given in Section 5.3.4. As each data set was ran 5 times, there are 5 results per data set.

Table 25. The results of the experiments AVNS for each of the separate runs.

Dataset	Adapted shaking		Swap		Move		Switch Vehicles		Reverse Routes	
	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)
3.1	5.06	0.2949	10.06	0.6851	8.32	0.7451	6.80	0.3705	11.72	0.7751
3.2	3.84	0.5950	11.73	0.5850	5.23	0.5052	9.07	0.4769	7.21	0.3338
3.3	6.15	0.3109	12.84	0.5902	7.56	0.5449	18.12	0.4926	18.33	0.3813
3.4	12.92	0.3692	10.12	0.4900	11.40	0.3897	7.11	0.3301	8.66	0.6803
3.5	17.89	0.3494	20.98	0.3105	8.45	0.7063	8.54	0.3500	11.40	0.4199
5.1	9.65	2.4344	2.77	0.5892	7.30	0.6702	7.98	0.7398	8.82	1.3304
5.2	5.39	2.4193	3.01	0.5906	5.67	0.6446	3.64	0.7600	3.54	0.7146
5.3	2.43	0.5408	3.47	0.7298	4.15	0.7900	3.21	0.6305	3.10	0.5853
5.4	2.38	0.5759	7.03	0.4901	6.22	0.5412	6.02	0.6445	5.51	0.7346
5.5	5.82	0.6692	4.02	0.7001	5.14	0.8090	4.68	0.5269	5.93	0.6924
6.1	8.37	1.6135	14.77	2.3898	13.28	2.1635	9.77	2.5199	9.78	2.0616
6.2	11.66	1.5665	8.33	1.9659	9.90	2.1455	9.86	2.3313	11.67	1.9581
6.3	10.27	2.8294	14.08	2.0144	7.20	1.4428	11.92	3.4509	14.93	1.8902
6.4	12.70	1.8182	10.29	1.6799	6.19	1.7478	14.12	2.5316	18.38	3.5989
6.5	14.82	2.5529	6.99	1.7098	12.93	2.8575	14.18	1.7363	12.17	2.9768

Table 26. The results of the experiments AVNS for each of the separate runs.

Dataset	Increased improvement		Decreased improvement	
	(%)	Time(s)	(%)	Time(s)
3.1	10.74	0.6862	7.14	0.5600
3.2	16.30	0.4443	19.28	0.4703
3.3	8.31	0.4894	7.69	0.6107
3.4	26.18	0.3401	3.68	0.3722
3.5	9.01	0.3318	8.39	0.5811
5.1	2.09	0.4762	2.65	0.6302
5.2	5.78	0.6760	3.56	0.6690
5.3	5.42	0.6385	3.01	0.6467
5.4	6.23	0.6414	5.06	0.5493
5.5	5.25	0.7737	3.49	0.8828
6.1	7.22	1.3963	13.58	2.880
6.2	9.55	2.0449	10.78	1.9729
6.3	10.34	2.1322	13.88	3.6335
6.4	4.18	1.8229	13.89	1.7995
6.5	11.23	2.3942	9.20	2.0043

Table 27. The improvements

Data set	Amount of shakes									
	1		3		7		11		Adapted	
	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)	(%)	Time(s)
3.1	9.62	0.3765	15.60	0.3429	1.28	0.5895	9.53	0.3878	10.21	0.4253
3.2	12.11	0.5347	14.01	0.6796	7.56	0.3900	7.09	1.0249	11.25	0.6268
3.3	9.93	0.3644	19.80	0.4280	17.37	0.4001	17.94	0.4872	10.31	0.5984
3.4	10.43	0.3363	13.96	0.3860	4.82	0.5689	16.26	0.5610	10.24	0.5146
3.5	9.44	0.5760	3.07	0.4925	3.01	1.0481	8.11	0.9794	7.46	0.4976
5.1	5.97	0.7707	2.31	0.5231	10.92	0.5852	2.04	0.7482	4.01	0.7874
5.2	2.94	0.6748	3.60	0.5899	6.56	0.6951	3.26	0.7312	4.94	0.5510
5.3	6.40	0.5240	0.02	0.5672	10.35	0.7003	2.04	0.6088	4.61	0.8169
5.4	2.85	0.5717	2.20	0.5396	5.87	0.6116	3.54	0.5501	2.87	0.5729
5.5	3.21	0.4560	5.97	0.7357	7.58	0.6281	3.75	0.6567	2.16	0.7699
6.1	18.10	3.0023	11.00	5.0990	19.12	4.1748	13.80	3.0795	12.23	1.9741
6.2	13.77	2.6438	10.57	2.8510	8.61	2.0795	17.15	2.4881	8.21	2.5406
6.3	17.38	2.4410	6.80	2.7334	7.30	3.9600	11.43	2.1382	14.57	2.7888
6.4	14.33	2.8324	13.51	10.6761	9.16	9.2103	14.92	1.9420	7.23	5.2803
6.5	9.58	1.6180	12.79	7.7842	13.69	9.3853	16.40	2.5221	9.13	8.9707

C Real world data

The lease for two of the vehicles (Renaults) the food bank in Almelo uses is €6.000 per year. This is equal to €500 per month or €16,44 per day. The other vehicle (Opel) used by the food bank in Almelo has a combined cost, consisting of taxes, maintenance and insurance, of €6.000 per year. This is equal to €500 per month or €16,44 per day. It can be assumed that these costs are similar for the vehicles used by the other food banks in the region. Regarding the fuel costs, the Renaults cost €0,143 per kilometer, and the Opel €0,125. This also means that the Renaults have a usage of 9,75 liter fuel/100km, and the Opel has a usage of 8,5 liter fuel/100km. This is more than expected (Kramer, n.d.; den Hurk, n.d.). Therefore, when considering new vehicles, the usage should be considered to be more than given by manufacturers and/or salesman. This is particularly relevant for determining the costs for the vehicles expected to be used for the pilot.

D Route Figures

Figures 16, 17 and 18 are the routes for commodity 1, 2 and 3 respectively on the Wednesday. These are the results from the MILP. The different colors represent the different routes.

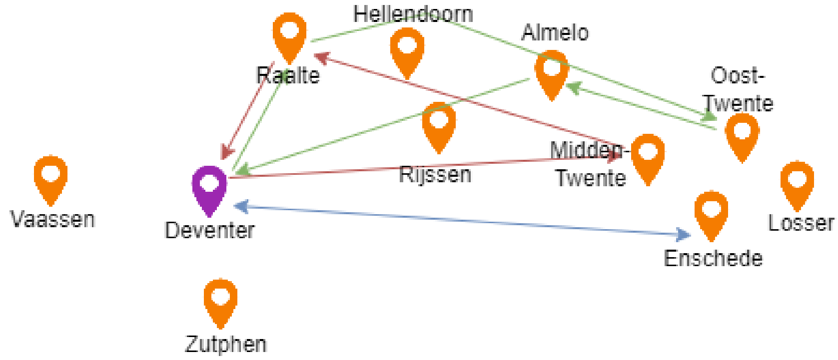

Figure 16. The routes for transporting commodity 1 on Wednesday as determined by the MILP.

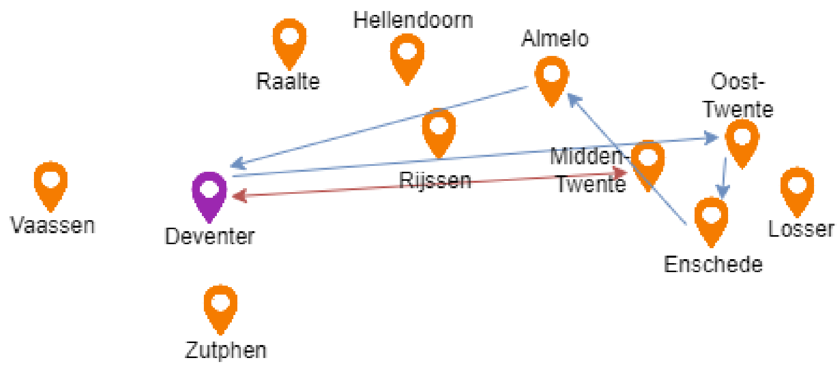

Figure 17. The routes for transporting commodity 2 on Wednesday as determined by the MILP.

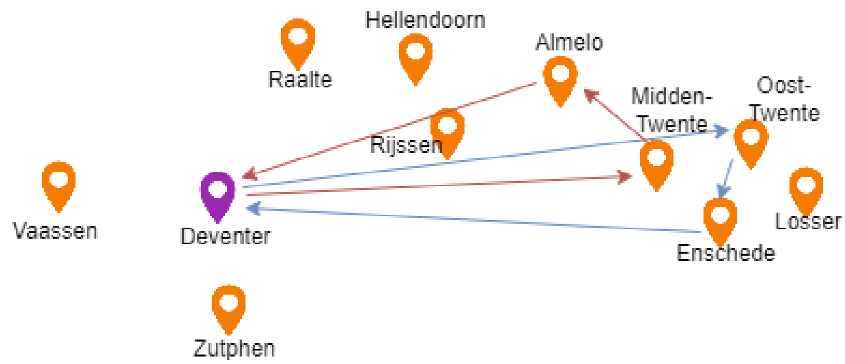

Figure 18. The routes for transporting commodity 3 on Wednesday as determined by the MILP.

The next two figures, Figure 19 and Figure 20 are the routes for commodities 2 and 3 on a Thursday respectively. There are no results for the Firday, as it was unable to find a feasible solution.

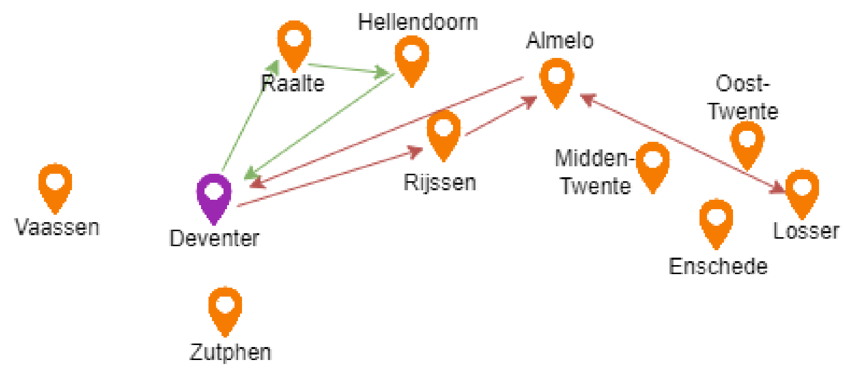

Figure 19. The routes for transporting commodity 2 on Thursday as determined by the MILP.

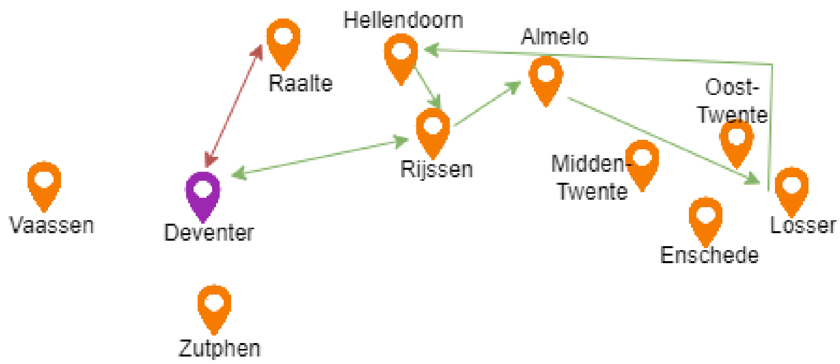

Figure 20. The routes for transporting commodity 3 on Thursday as determined by the MILP.

Next the results from the VNS are given for all commodities on the Wednesday in Figures 21, 22 and 23. Followed by the results from the Thursday for commodities 0, 2 and 3 in Figures 24, 25 and 26.

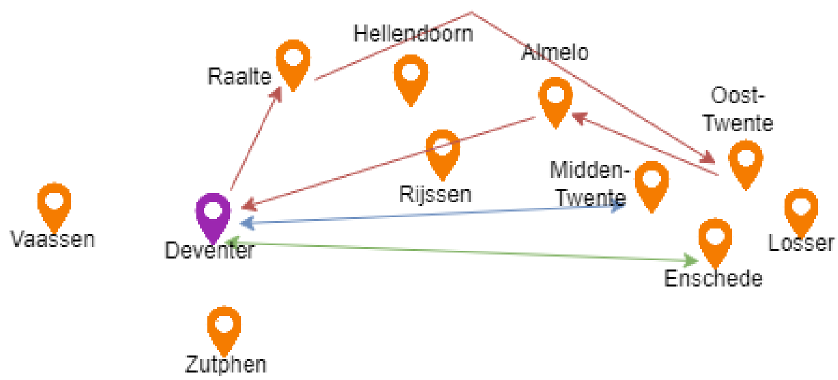

Figure 21. The routes for transporting commodity 1 on Wednesday as determined by the VNS.

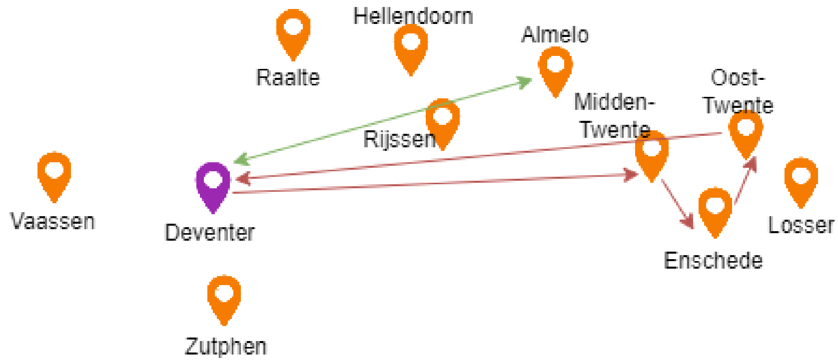

Figure 22. The routes for transporting commodity 2 on Wednesday as determined by the VNS.

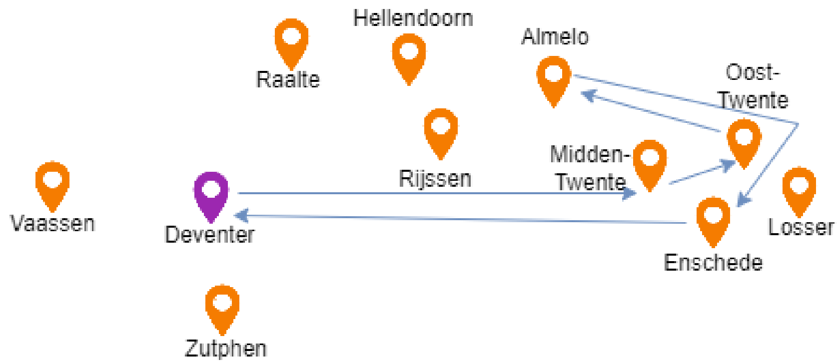

Figure 23. The routes for transporting commodity 3 on Wednesday as determined by the VNS.

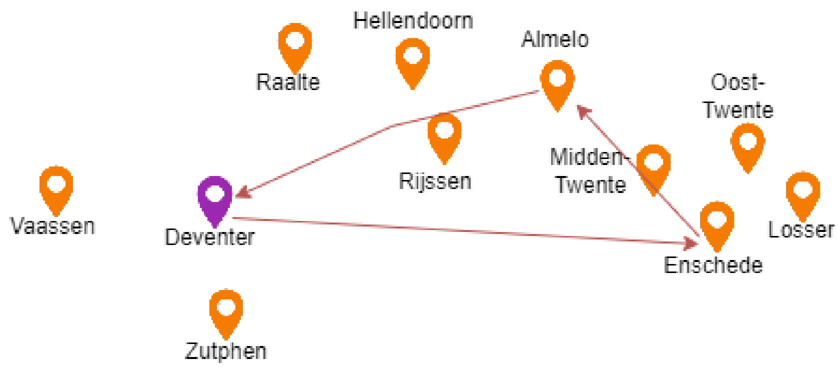

Figure 24. The routes for transporting commodity 0 on Thursday as determined by the VNS.

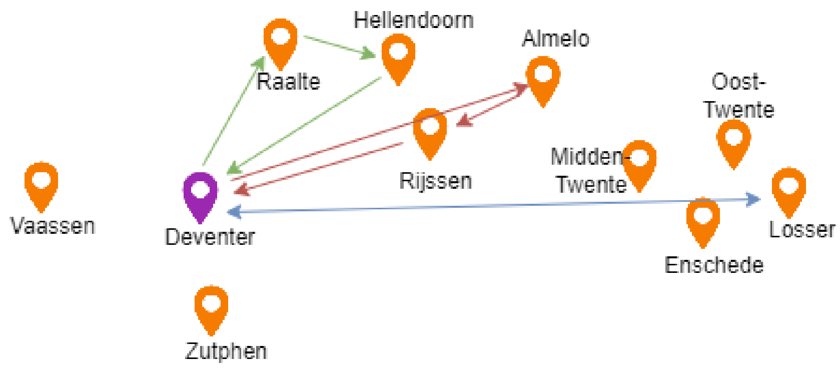

Figure 25. The routes for transporting commodity 2 on Thursday as determined by the VNS.

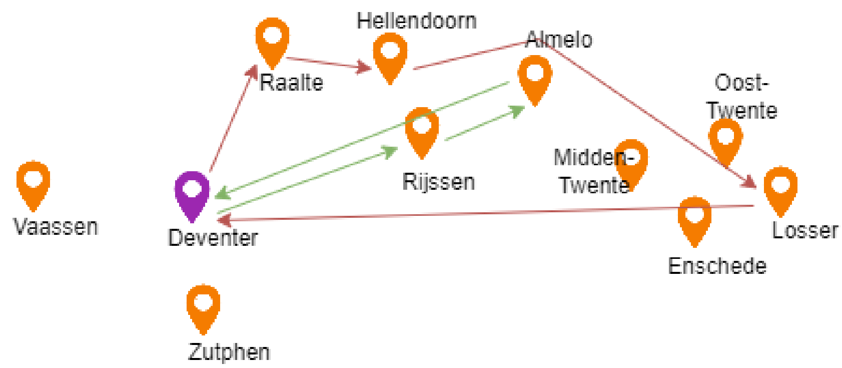

Figure 26. The routes for transporting commodity 3 on Thursday as determined by the VNS.

The results from the VNS for the Friday are given in Figures 27, 28, 29 and 30.

Figure 27. The routes for transporting commodity 0 on Friday as determined by the VNS.

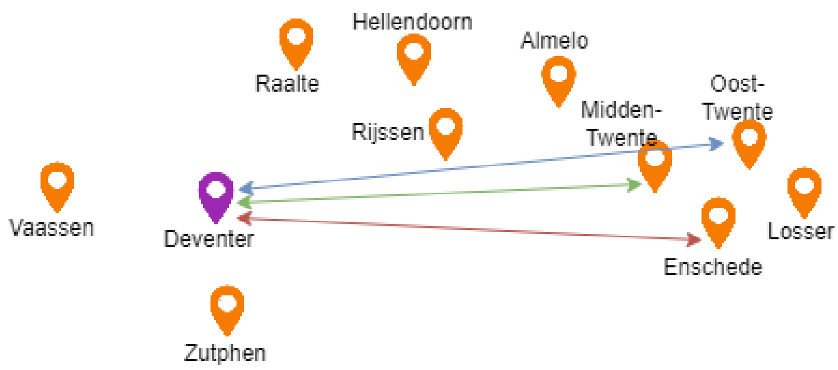

Figure 28. The routes for transporting commodity 1 on Friday as determined by the VNS.

Figure 29. The routes for transporting commodity 2 on Friday as determined by the VNS.

Figure 30. The routes for transporting commodity 3 on Friday as determined by the VNS.

Finally, the routes of the MILP with centralized transportation for the Wednesday and Friday are given in Figures 31 and 32.

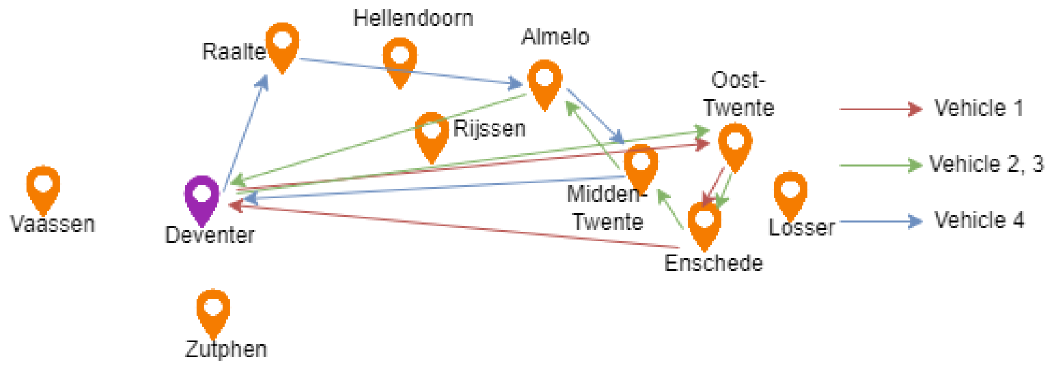

Figure 31. The routes for transport for Scenario 4 generated by the MILP.

Figure 32. The routes for transport for Scenario 13 generated by the MILP.