Master's Thesis

Optimizing the moment of customer delivery in ORTEC Inventory Routing

Loes A.M. Knoben

May 2016

UNIVERSITY OF TWENTE.

ORTEC

Author

Loes A.M. Knoben

University

University of Twente

Faculty

Electrical Engineering, Mathematics and Computer Science

Master Programme

Applied Mathematics

Specialization

Operations Research

Chair

Discrete Mathematics and Mathematical Programming

Graduation Date

May 25, 2016

Assessment Committee

University of Twente

Prof. dr. M. Uetz

Dr. ir. G. F. Post

Dr. J.C.W. van Ommeren

ORTEC

Prof. dr. J. A. dos Santos Gromicho

J. C. Kooijman, MSc.

Abstract

ORTEC Inventory Routing is a Vendor Managed Inventory solution that minimizes the long-term costs involved in distributing a product to multiple customers, while preventing stock-outs at those customers. In this thesis we aim to identify opportunities to improve the selection criteria used by OIR to decide on a day by day basis which customers to deliver to. Currently this decision is integrated in the construction of the routes by using the daily cost per volume as a short-term objective to minimize the total cost per volume over the horizon. First, we design two short-term solution approaches that aim to minimize the daily cost per volume without constructing the delivery routes. We find that the daily cost per volume is not always a good indicator of the total cost per volume, but that a method incorporating a capacitated minimum spanning tree nevertheless shows promising results. Second, we design alternative short-term objectives and examine their effect on the long-term objective. In doing so we find that optimizing the cost per volume of each delivery to a customer gives better results than optimizing the cost per volume of each day or each route.

Contents

1 Introduction

Optimizing distribution and transportation costs is of great importance in the Oil and Gas industry. Due to the growing complexity of business processes and the increasing amounts of data at our disposal, expectations rise, while making good decisions is getting increasingly difficult. One of the recent trends is to integrate different supply chain aspects. This happens for example with Vendor Managed Inventory (VMI), where the vendor takes the responsibility for the inventory of its customers by combining inventory management and route scheduling. In this concept, the customer does not have to monitor its inventory levels or place orders, but instead the vendor decides when, how much and how to deliver. By using VMI, the vendor gets more freedom and can choose the delivery moments such that suitable deliveries can be combined better. As a result, the vendor can save on transportation costs and the customer can save on resources for inventory management at the same time.

ORTEC is a company that is specialized in advanced planning software to support the customers' decision making, in order to optimize business processes. The company follows the recent trend of VMI by developing ORTEC Inventory Routing (OIR), a software product that integrates demand forecasting, stock replenishment, route planning and route execution. The goal of OIR is to minimize the costs of distributing a certain product over a long horizon, while preventing stock-outs at customers. Three decisions have to be made to achieve this: when to deliver to a customer, how much to deliver to a customer and which routes to use to fulfill these deliveries. Answering these three questions is referred to as the Inventory Routing Problem (IRP). OIR can be used in a wide range of transport and distribution planning sectors, for the delivery of products like gas, oils and chemicals. The product is especially successful when the amounts to deliver are limited, while the costs involved are high.

Most research into improving the performance of OIR has focused on the forecasting, the routing or on the delivery volumes, however the aspect of optimizing the moment of customer delivery has been underexposed. In order to deal with the high complexity of the IRP and the large amount of stochasticity involved, OIR solves the problem on a day to day basis. Optimizing the moment of delivery therefore translates to deciding to which customers to deliver on the upcoming day and to which not. We will refer to this problem as order selection.

When deciding to which customers to deliver, both short-term and long-term effects should be considered, however these might be conflicting. Because decisions in OIR are made on a day by day basis, a short-term objective is used that should reflect the goal of minimizing the long-term costs. A heuristic is used to minimize this short-term objective, hence a different approach can result in better solutions. Besides this, using a short-term objective to minimize the long-term objective is a heuristic on its own, indicating that the use of a different short-term objective might results in a better solution. This research will examine these two lines of improvement.

This thesis is structured as follows. In Chapter 2 we introduce the problem in more detail and formulate the scope and goal of this research. In Chapter 3 we provide an overview of the literature that is relevant to this research. In Chapter 4 we explain a short-term solution approach that aims to improve the minimization of the given short-term objective, by one that improves the long-term performance. In Chapter 5 we discuss a long-term solution approach that aims to replace the given short-term objective. In Chapter 6 we present the design of our experimental environment and the setup of the experiments. In Chapter 7 we analyze the results of the experiments. In Chapter 8 we close with the conclusions and recommendations of our research.

2 Problem Description

This chapter introduces the problem that is examined in this research. First Section 2.1 explains the terminology and Section 2.2 provides some background on the problem and describes the current situation at ORTEC. Then the research motivation, scope and goal are given in Sections 2.3, 2.4 and 2.5. The chapter closes with the mathematical problem formulation in Section 2.6.

2.1 Terminology

This report contains many definitions related to the Inventory Routing Problem (IRP) that might be ambiguous. This section introduces the most important definitions as used throughout the research. A visualization of some of the definitions can be found in Figure 2.1. An overview of the used abbreviations can be found in Appendix A.

Planning horizon: The set of days for which the IRP is solved and routes need to be constructed. This can for example be a few months or a year.

Planning window: A smaller set of days that is considered when deciding which customers to deliver to on the upcoming day. The standard length is one day, but this can be extended to a longer period of time.

Safety stock level: The inventory level at a customer should always be above the safety stock level to guard against stock out. This level is defined as a fraction of the total capacity of a customer.

Earliest delivery level: Delivery to a customer is allowed from the moment the inventory level reaches the earliest delivery level, which is again a fraction of the customers capacity. Before this moment, the volume that can be delivered to the customer is too small to consider it for delivery. This level is introduced to prevent that a customer is delivered too often, which might be bad for safety issues and long-term performance.

Must-go customer: A customer whose inventory will reach the safety stock level in the planning window. This means that a delivery to this customer is required.

May-go customer: A customer whose inventory is below the earliest delivery level but will not reach the safety stock level in the planning window. In this case a delivery to the customer may be made, but it is not necessary.

Order: A customer that is selected for delivery in the planning window.

Delivery window: Each order has a delivery window that specifies when delivery can take place. It depends on aspects like the opening hours of a customer and the moment the earliest delivery level and safety stock level are reached.

Logistics ratio: The cost per volume corresponding to delivering a certain set of orders. The long-term logistics ratio (LTLR) is the cost per volume over the entire planning horizon and the short-term logistics ratio (STLR) is the cost per volume over the planning window.

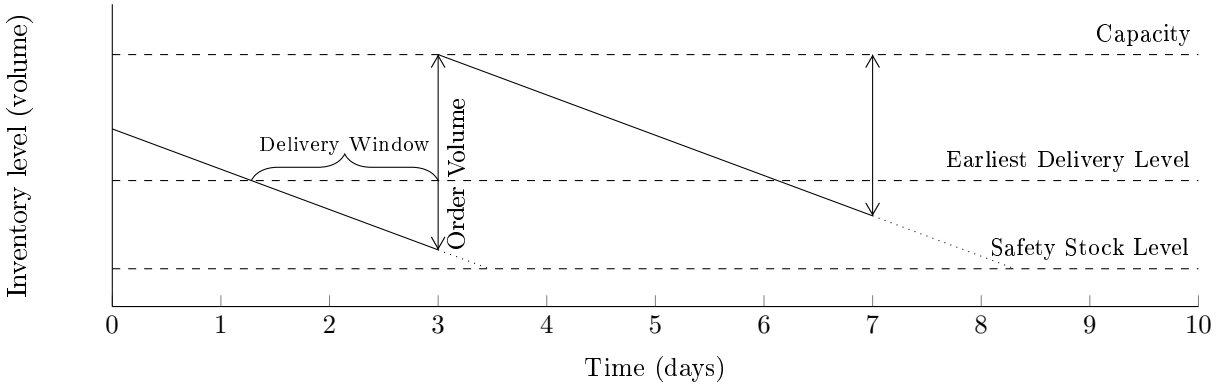


FIGURE 2.1: Visualization of the inventory level of a customer. On day 3 the customer is a must-go customer, since its inventory level will reach the safety stock level during the day. On day 7 it is a may-go customer, since this delivery can also be postponed to day 8.

2.2 Background

To have a clear understanding of the background of the problem being investigated, the upcoming sections provide extra information about the Inventory Routing Problem and the way that it is solved by ORTEC Inventory Routing.

2.2.1 Inventory Routing Problem

Due to aspects like rising competition and reduced profit margins, the need to increase efficiency and decrease costs keeps growing. Companies deal with this by introducing new technologies and innovations, for example organizing the supply chain differently [1]. In Vendor Managed Inventory (VMI) the vendor gets more freedom by taking the responsibility for the inventory of its customers. In order to translate the gained freedom into benefits, the vendor has to find an efficient delivery schedule that satisfies all customers. Solving this problem, means the supplier has to combine inventory management and fleet management, which means taking decisions on replenishment of customers and finding optimal delivery routes respectively. This results in the Inventory Routing Problem (IRP), where both aspects are integrated into one framework in order to take the dependencies between these components into account. This problem was first introduced by Bell et al. [2] more than thirty years ago and a considerable amount of research has been done into different variants of the problem since. We describe the basic deterministic version of the problem that most resembles the IRP as solved in ORTEC Inventory Routing.

The basic IRP considers a set V of n customers, where every customer $i \in V$ has a deterministic product consumption of u_i per day, an initial inventory of $I_i(0)$ at time 0 and a capacity of C_i. Inventory holding

costs are not taken into account and time is discretized into days. The IRP deals with distributing one product from a single depot, which has unlimited product availability, over this set of customers during a given planning horizon of T days. To make sure customers do not experience stock-out, we require that the inventory level at a customer is bigger than the safety stock level at every moment in time. A solution to the IRP includes the following three aspects:

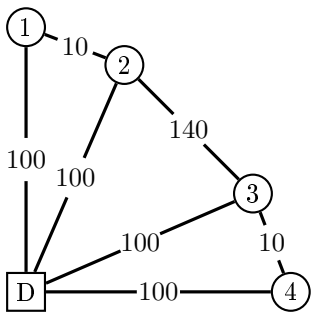
- When to deliver to a customer
- How much to deliver to a customer
- Which delivery routes to use

All routes are required to start and end at the depot and there is an unlimited set of homogeneous vehicles with capacity Q available to execute these routes. Fixed costs f are charged when using a vehicle for a route and transportation costs c_{ij} are involved in driving from one customer i to another customer j. These transportation costs depend on the distance and the travel time between the customers. The delivery routes specify which customers are visited on every route and in what order they are visited. Based on this information, the cost of a route can be calculated. The total distribution cost of a day in the planning horizon is the sum of the costs of all routes executed during this day, plus the fixed costs based on the number of routes. The goal is to minimize the distribution costs summed over the planning horizon under the constraint that no customers experience stock-outs [3].

Example 2.2.1. To illustrate the dynamics of the IRP, Bell et al. [2] present a simple example with four customers. The capacities and usage rates of all customers can be found in Figure 2.2a and all possible roads between the customers are displayed and labeled with the corresponding distances in Figure 2.2b. Suppose our vehicle capacity is 5000 units and all customers have a starting inventory equal to their

	Customers			
	1	2	3	4
Capacity	5000	3000	2000	4000
Usage rate	1000	3000	2000	1500

(a) Capacities and usage rates.

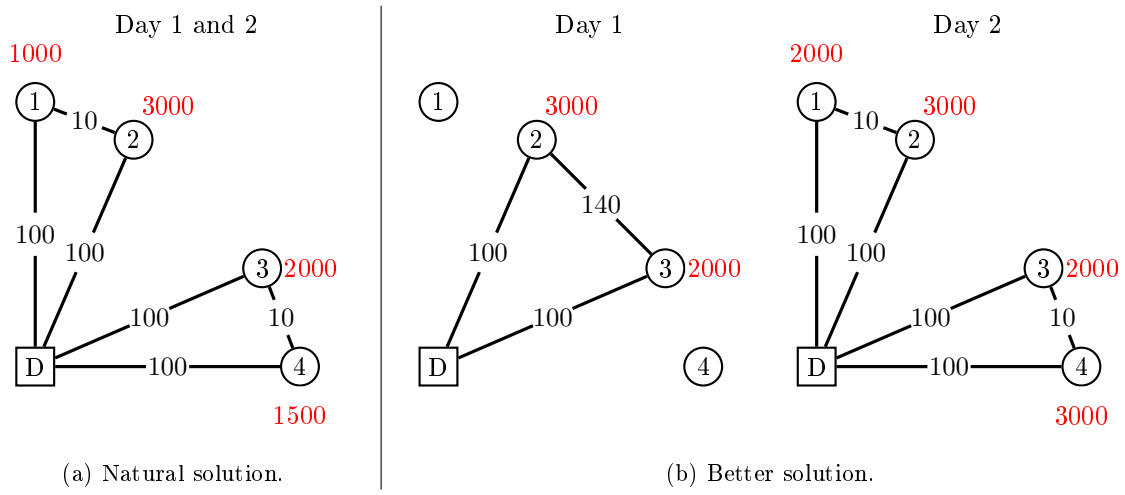


(b) Distances.

FIGURE 2.2: Properties of the customers in Examples 2.2.1 and 2.2.2. Figure 2.2a shows their capacities and usage rates and Figure 2.2b shows all roads between the customers and the corresponding distances.

capacity. We are interested in finding a schedule for the next two days that accomplishes the required deliveries while traveling minimum distance. Suppose that we always deliver the maximum amount possible to a customer, given its current inventory level and its capacity. The most obvious delivery schedule in this example would have two routes for every day of the horizon, delivering all four customers as displayed in Figure 2.3a. This schedule results in a total distance of 840, while delivering a volume of 15000 units. A better solution however, would be to deliver only customers 2 and 3 on the first day and all customers on the second day and repeat this periodic solution for every day of the horizon as is visualized in Figure 2.3b. This way the capacities of the vehicles can be used optimally. In this solution,

the vehicles deliver the same volume in two days, but only drive a total distance of 760. Moreover there are only three vehicles needed instead of four. Both solutions are displayed in Figure 2.3.



(a) Natural solution. (b) Better solution.

FIGURE 2.3: Illustration belonging to Example 2.2.1. The natural solution in Figure 2.3a delivers a volume of 15000 in two days, by driving a distance of 840. The better solution in Figure 2.3b delivers the same volume in two days, by driving a distance of only 760.

2.2.2 ORTEC Inventory Routing

ORTEC Inventory Routing (OIR) solves the Inventory Routing Problem day by day, where the solution approach for each day is divided into multiple phases. The first phase is the forecasting phase, where the usage rate of each customer is determined. The forecasted usage rate is then used in the order generation phase to calculate delivery windows and volumes for all customers. Based on the delivery windows the orders are determined, which are the customers that will get a delivery today. Next the routing phase uses these orders as input to construct the delivery routes for the vehicles. In the execution phase these vehicle routes are used to deliver the orders and real-time adjustments can be made when there are delays or the inventory levels at customers are not as expected. Information on inventory levels and deliveries, acquired in the execution phase, is used as input for the forecasting of the next day. The structure of the phases is shown in Figure 2.4. All these phases are performed subsequently for each day of the planning horizon. We discuss them in more detail in the next paragraphs.

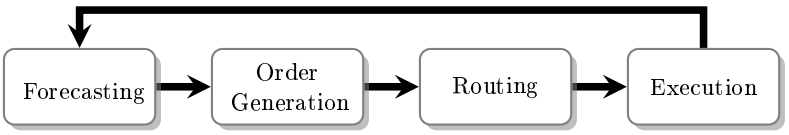


FIGURE 2.4: Phases of ORTEC Inventory Routing

Forecasting phase

When executing OIR, the first phase is the forecasting phase, where the demand of a customer is determined with the help of historical data. As input we have the customer history, consisting of data on delivered volumes, stock measurements and sales data. Various forecasting parameters are used that help determine the usage. Daily, weekly and yearly profiles are used to specify patterns in time, weekday

and season respectively. Besides this there is a set of special days to indicate exceptions like holidays and there is a correction for exceptional usage during such a day. Last there is the customer data, like opening hours. The forecasting phase uses all this data to predict the usage for the current week and specifies this as an hourly usage rate based on the weekly and daily patterns.

The output of the forecasting phase is an hourly usage rate for every customer.

Order generation phase

The second phase in OIR is the order generation. In the order generation phase delivery windows and volumes are computed for all customers. The usage rate generated by the forecasting is the most important input for this phase. The first step is to determine the moment that the customer reaches the safety stock, which will give the latest possible moment for delivery. The earliest possible moment for delivery is determined in the same way, based on the earliest delivery level. Combining these two values with the opening hours and a minimal delivery window gives the order delivery window for the customer. Based on the right hand side of this delivery window the customer is then marked as a must-go customer or a may-go customer. Must-go customers need a delivery today to prevent stock out, while delivery to may-go customers is optional as explained in Section 2.1.

After the computation of the delivery window, the order volume is computed, based on parameters like the delivery window, the safety and maximum stock, the minimum delivery amount and the vehicle capacities. For this calculation there are multiple strategies available. Based on the maximum inventory level of the customer and the estimated inventory level at the time of arrival, a fixed volume can be determined. An example of the determined delivery window and fixed order volume has been shown in Figure 2.1. Instead of determining a fixed order volume, two options for flexibility can be used, namely the min-max volumes and time dependent volumes. In the min-max strategy a minimum and a maximum amount are calculated and in the routing phase all volumes between these bounds are allowed in the routing phase. This means that there is more flexibility in combining orders of different customers together in a vehicle. The maximum and minimum volume are calculated as a prespecified fraction of the fixed order volume. In the time dependent strategy there are different volumes calculated corresponding to different delivery moments. The final volume depends on the exact moment the order is planned. This means changing the estimated time of arrival in the routing also changes the order volume. The volume will be chosen as the maximum amount possible at the exact moment of delivery, as is visualized in Figure 2.5.

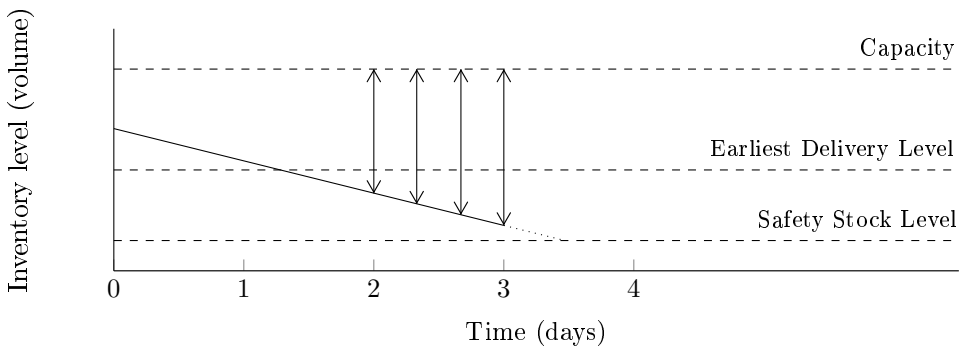


FIGURE 2.5: Four different order volumes depending on the exact moment of delivery during the day.

The output of the order generation phase is a set of must-go and may-go customers, where each customer is assigned a delivery window and one or multiple possible volumes.

Routing phase

After the order generation phase, the routing phase is executed. In this phase the delivery routes for the current day are constructed containing all must-go customers and possibly some may-go customers. The routing phase uses the cost per volume of this day as objective. This means that adding may-go customers is profitable if the detour needed on this day is relatively small while the volume that can be delivered is relatively high. The routing phase uses the flexibility in moments of delivery and volumes to deliver, to optimize the cost per volume. With this approach there is flexibility in scheduling the orders such that favorable orders can be combined in order to minimize long-term costs. This performance measure will be discussed into more detail in Section 2.2.3.

The first step of the routing phase is to assign all must-go customers to the set of necessary routes, since they require delivery on the current day. In this step the most 'difficult' customers are considered first. Difficulty is by default determined based on the largest distance to the depot, but aspects like small opening hours, order sizes or restrictions on which vehicles are allowed for delivery can also be taken into account. Orders are added to the vehicle in sequence of closest distance to the orders already in the vehicle. This means that must-go customers that are located far from the depot and far from other must-go customers, will be planned in separate vehicles. After all must-go customers are assigned to a route, the next step uses a greedy approach to add may-go customers to a route, until vehicles are full. However a may-go customer is only added if this decreases the cost per volume of this route, which indicates that it probably improves the long-term performance. The last step performs re-optimization of the solution to reduce the costs of the current planning. Several iterative optimization methods can be used for this improvement, that for example exchange orders within routes or between routes. Since one improvement method may lead to the possibility of improvement by another method, they are executed repeatedly. The cost criterion used in this optimization step is based on travel distance and travel time. There are no more customers added to or removed from the delivery schedule during this re-optimization.

The output of the routing phase is a set of vehicle routes that specify the customers in a vehicle, their volumes and the order of delivery.

Execution phase

After the orders are generated and the delivery routes have been determined, the actual deliveries can take place. During the deliveries, it may turn out that the forecasts are different from the actual situation. This might for example be true for the predicted usage rate or the driving times between customers. This can result in changes in the volumes and routes, so there is need for real-time updates and changes. It might even occur that all customers in a route require larger volumes, in which case a delivery to a may go customer will be skipped All the actual deliveries that are made are registered and together with new measurements of tank levels they will be used in the next iteration of the forecasting phase.

The output of the execution phase are the measurements at the customers and the actual deliveries that have been done.

2.2.3 Performance Measure

In standard vehicle routing, the goal is to minimize the total transportation costs within a scheduling period. In inventory routing the goal is to minimize long-term transportation costs, given that all usage at a customer eventually needs to be replenished. When the problem is solved day by day, only short-term costs are taken into account, which are costs of the daily routes. Minimizing these short-term costs will not necessarily result in minimum long-term costs. Minimizing the daily costs actually results in only delivering customers that reach safety stock on the upcoming day, since adding a customer almost always results in a detour. Delivering to a customer that does not need delivery can however be beneficial in the long run, especially when a very large volume can be delivered or when the detour is small compared to the detour on future days.

Taking the long-term aspect into account can be done by minimizing the cost per volume instead of the costs. A low cost per volume means that the delivery routes are cost-effective. When this performance measure is used, adding a customer to the routes that does not yet need delivery is seen as profitable when the detour is small, while the volume that can be delivered is high. Moreover, using this measure provides us with the ability to evaluate and compare distribution strategies at different moments in time. The costs can vary substantially based on the geographical locations and storage capacities of customers, however the cost per volume of delivering to a certain set of customers over a period of time does not fluctuate that much. The performance indicator within ORTEC Inventory Routing is therefore chosen as the cost per volume, which we refer to as the logistics ratio (LR). We define the long-term objective as the long-term logistics ratio (LTLR) over the planning horizon T:

$$\mathrm{LTLR}(T) = \frac{\sum_{t=1}^{T} \mathrm{cost}(t)}{\sum_{t=1}^{T} \mathrm{volume}(t)} \tag{2.1}$$

Seeing that OIR solves the problem day by day, a short-term objective is introduced for the optimization of a single day. When this short-term objective is appropriately defined, minimizing it should result in simultaneously minimizing the long-term objective of Equation 2.1. The short-term objective that is used in OIR for a day t of the planning horizon, considers the cost per volume of that day and is called the short-term logistics ratio (STLR):

$$\mathrm{STLR}(t) = \frac{\mathrm{cost}(t)}{\mathrm{volume}(t)} \tag{2.2}$$

Note that the sum or average of all short-term objective values is not equal to the value of the long-term objective. Minimizing the STLR will therefore not necessarily result in minimizing the LTLR. Even though minimizing the cost per volume for every day gives better results than minimizing the costs, it still does not guarantee optimal long-term performance. We illustrate this with a small example.

Example 2.2.2. Consider again the situation as displayed in Figure 2.2 and suppose we always deliver the maximum volume possible. On day one we have to deliver customers 2 and 3 to prevent stock out and we can decide whether to add customer 1 or customer 4 or both. The minimum is attained when both customers 1 and 4 are added, which results in a short-term objective value of 0.056. Minimizing the daily cost per volume in this situation, results in delivering all customers every day as was displayed in the natural solution in Figure 2.3a and gives us a total cost per volume of 0.068 over the horizon. However, the better solution as was displayed in Figure 2.3b results in a long-term objective value of 0.051. This

shows that finding the minimum cost per volume for every day can result in a higher long-term cost per volume and hence is suboptimal.

With this observation in mind, we will analyze the computational performance of alternative short-term objectives and how they actually perform in the long-term.

2.3 Research Motivation

The current product OIR as described in Section 2.2.2, is originally created by combining two products, one for forecasting demand and one for constructing routes. As a result, most improvements that have been executed over the years, were mainly focused on improving either the forecasting or the routing phase. The aspect that has been underexposed, is the decision which customers are delivered on a particular day. In the current situation this decision is integrated in the routing framework and the long-term performance measure of total cost per volume over the horizon is minimized by using the short-term performance measure of daily cost per volume. After all must-go customers are assigned to routes, may-go customers that improve the cost per volume of the day under consideration are selected for delivery.

In this framework the may-go customers are examined one by one, in a specific order and in the context of the initial solution for the must-go customers. These aspects may lead to a selection of orders that is not optimal with respect to the short-term objective. Furthermore, only the situation of the current day is considered when making the decision. This means that a solution that is optimal with respect to the short-term objective, might not be optimal with respect to the long-term objective. We discuss four examples where the current approach does not select the optimal order for a certain day. Examples 2.3.1 and 2.3.2 focus on optimality with respect to the short-term objective and Examples 2.3.3 and 2.3.4 focus on optimality with respect to the long-term objective.

Example 2.3.1. We consider a route that consists of two must-go customers. Suppose there are two may-go customers nearby that can both be added to the route. We may encounter the situation where adding both may-go customers decreases the cost per volume, however adding either does not decrease this performance measure. In the current approach this means that none of the customers is added, however adding both might give better results. A corresponding illustration can be found in Figure 2.6.

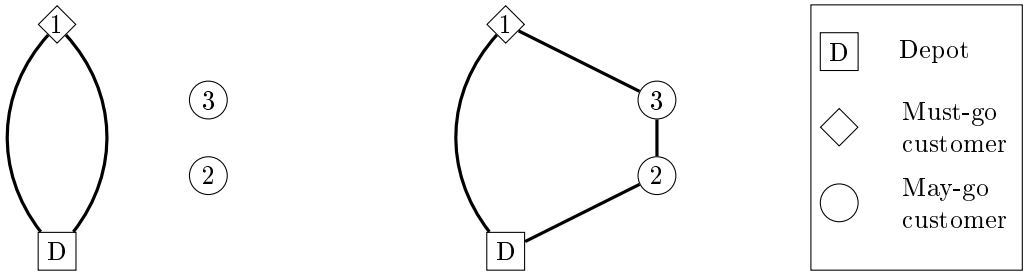


FIGURE 2.6: Illustration belonging to Example 2.3.1. Adding only one of those customers might increase the cost per volume, but adding both could decrease the cost per volume.

Example 2.3.2. Next, consider two must-go customers that are located far away from each other but fit together in one vehicle. The current approach will construct one route with both customers, while

it might be better to have two separate routes if there are profitable may-go customers located nearby both must-go customers. A corresponding illustration can be found in Figure 2.7.

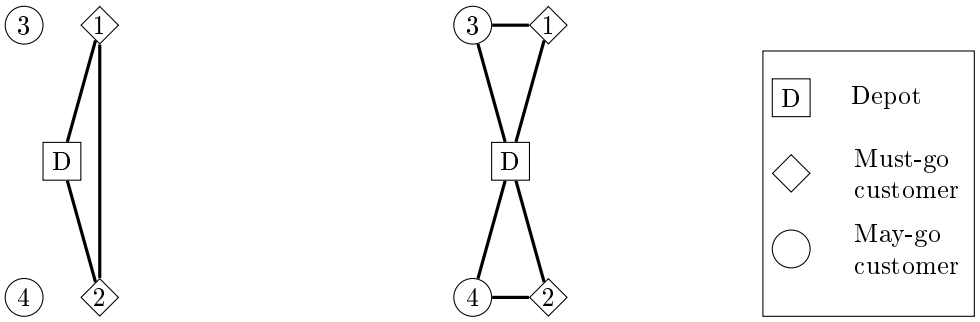


FIGURE 2.7: Illustration belonging to Example 2.3.2. Including all customers could result in the minimum cost per volume for this day. However the initial solution will combine customers 1 and 2, making it impossible to add another customer.

Example 2.3.3. This example describes a situation where an 'easy' customer is delivered too early. A situation where this happens is when adding a may-go customer on day t slightly improves the performance measure, while adding this customer on day $t+1$ improves the performance measure significantly more. In this case the customer will be planned on day t, while planning it on day $t+1$ would give a better long term result. This situation occurs when customers are relatively easy to deliver to, an example of such customers is given in Figure 2.8, where multiple customers are located close to the depot and close to each other.

Example 2.3.4. The last example illustrates a situation in which a 'difficult' customer is postponed too long. This happens for example when a certain customer is difficult to plan because the distance to all other customers is quite large. Adding this customer might always increase the cost per volume, which results in postponing the customer until it becomes a must-go customer. If its capacity is much smaller than the vehicle capacity and there are no close may-go customers on this day such that no efficient routes are possible, the cost per volume will be very high. A better solution would then have been to plan the customer earlier even though it resulted in a small increase in cost per volume at that time. Characteristics might show that a certain customer is always expensive to deliver, which might be an incentive to loosen the criteria for this customer. An example of a difficult customer can be found in Figure 2.8, where customer 1 can only be suitably combined with customer 2. In this example it might be smart to add customer 1 when a delivery is made to customer 2, even if this increases the daily cost per volume. Otherwise it might be postponed too long.

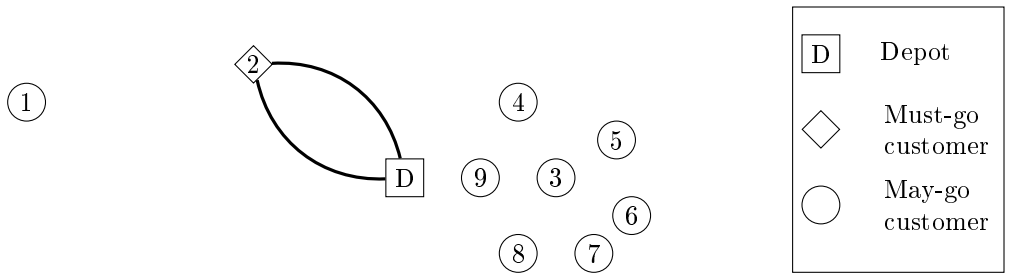


FIGURE 2.8: Illustration belonging to Examples 2.3.3 and 2.3.4. It is easy to deliver to customer 9. Even if adding it in this scenario decreases the daily cost per volume, it might not be the best choice. On the other hand it is difficult to deliver to customer 1. Adding it to the current route is probably smart, even if this increases the daily cost per volume.

These four examples show that there is still room for improvement when determining when to deliver to a customer in OIR. We summarize the motivation of our research as follows:

- Little research has been done into deciding the optimal moment of delivery for each customer. Instead the focus has been on forecasting the usage rate and constructing delivery routes.
- The current approach of selecting orders does not necessarily minimize the short-term performance measure as can be seen in Examples 2.3.1 and 2.3.2. Aspects involved are the construction of the initial solution and the fixed order of addition of may-go customers.
- The short-term objective that is currently used does not necessarily minimize the long-term objective, which is shown in Examples 2.3.3 and 2.3.4. When selecting the orders the focus is on the current day, without considering the past, the future or the characteristics of specific customers.

2.4 Research Scope

This research is performed at ORTEC, where a lot of knowledge is available on good solution methods for many optimization problems. We want to use the solution approaches that have been developed and improved over the years as much as possible. For this reason, this research will not focus on forecasting and routing, but instead take the software for these components as given. Hence we assume that we can construct routes when given a set of customers and that we are able to accurately predict usage rates based on historical data. This leads to the simplified setting of deterministic demands. Note that even though we assume our information to be deterministic, in reality the available customer information for the upcoming days might change. Moreover, the actual deliveries of the execution phase might differ from the planned deliveries as mentioned in Section 2.2.2. As a result, a daily approach should be maintained, where routes are constructed only for the current day and not for some bigger planning window.

We also have some assumptions that relate to the current approach of ORTEC and the requirements that are given by users of OIR. Inventory holding costs are not taken into account. That means that costs considered in the optimization consist of traveling distance, duration and fixed route costs. We assume that we have infinitely many homogeneous vehicles available to carry out delivery routes since extra vehicles can always be hired, which is taken into account in the cost function as fixed route costs. Last we assume that inventory levels can never be negative and we manage this by keeping the levels above safety stock level. This is justified because the product is mainly used in the oil and gas industry where backorders are not possible and enough product should always be available. We neglect additional constraints that might be used in OIR.

2.5 Research Goal

Section 2.2 identified that ORTEC Inventory Routing uses a daily solution approach and it discussed the short-term and long-term objectives that are used, which might be conflicting. Subsequently Section 2.3 highlighted that it is important to carefully consider which customers are delivered on which day, but that there has been little research into this aspect in ORTEC Inventory Routing. Afterwards Section 2.4 discussed that the forecasting phase and the routing phase lie beyond the scope of this research and

that we want to maintain a daily solution approach. These considerations bring us to the main goal of this research:

Identify opportunities to improve the long-term performance of ORTEC Inventory Routing by using different algorithmic approaches for the selection of orders.

We evaluate the long-term performance as the total cost per volume and we require that the order selection is performed on a day by day basis. Currently this order selection is integrated in the routing framework and the long-term objective of total cost per volume over the horizon is minimized by using the short-term objective of daily cost per volume. The examples in Sections 2.2.3 and 2.3, showed that this approach can result in a selection of orders that is not optimal with respect to both objectives. We consider two strategies to improve this decision mechanism, focusing either on the short-term objective or the long-term objective.

- Short-term solution approach. First, we attempt to improve the selection of orders by changing the current heuristic that is used, but preserving the short-term objective which minimizes the daily cost per volume. Chapter 4 examines the extend to which we can improve with respect to the short-term objective and evaluates the effect on the long-term performance.
- Long-term solution approach. Second, we opt to improve the order selection by focusing on the long-term effect of our decisions and questioning whether the current short-term objective is a good reflection of our long-term objective. Chapter 5 therefore examines the effect of different short-term objectives on the long-term performance.

2.6 Mathematical Problem Formulation

The mathematical problem of deciding which customers to deliver to can be stated as follows. Let $G = (V, E)$ be a complete graph with $V = \{0, 1, \dots n\}$ the vertex set and E the edge set. Vertex 0 is the depot and the other vertices $1, 2, \dots, n$ are customers that require deliveries during the planning horizon T. We denote these customers as $W = V \setminus \{0\}$. Each customer $i \in W$ has a fixed capacity C_i, a fixed usage rate u_i and an inventory level $I_i(t)$ that depends on the day t. To prevent stock out, the inventory level should always be bigger than the safety stock level, which is for every customer defined as a fraction SSF of its capacity. This gives us the following constraint

$$I_i(t) \geq SSF \cdot C_i \qquad \forall i \in W,\ \forall t \in T.$$

To satisfy this constraint we deliver a certain amount of product to the customers over the days, however we have to keep in mind that delivery to this customer is only allowed when the inventory level at a customer is smaller than the earliest delivery level of a customer, which is again specified as a fraction EDF of its capacity. We introduce a binary variable $x_i(t) \in \{0, 1\}$ that is one if delivery to customer i takes place on day t and zero otherwise. This means that we can calculate the inventory level of a customer at the beginning of day $t + 1$ before any deliveries have taken place as follows

$$I_i(t + 1) = I_i(t) - u_i + x_i(t) \cdot q_i(t).$$

In this formula $q_i(t)$ is the maximum volume that can be delivered at the beginning of the day. This means that it is based on its current inventory level of that day $I_i(t)$ and the maximum volume Q that can be transported in the vehicle:

$$q_i(t) = \min(C_i - I_i(t), Q).$$

The maximum volumes at the beginning of the day can be assigned as weights of the vertices W, which means that the graph is updated every day. The total volume that is delivered on a certain day can be found as

$$\text{volume}(t) = \sum_{i \in W} q_i(t) \cdot x_i(t). \tag{2.3}$$

To make sure that we satisfy the constraints imposed by the safety stock fraction and the earliest delivery fraction, a delivery window $[a_i(t), b_i(t)]$ is constructed for each customer at the beginning of day t, that specifies the earliest and latest moment of delivery for customer i. These boundaries are determined with the following formulas

$$a_i(t) = t + \left\lfloor \frac{I_i(t) - (1 - EDF) \cdot C_i}{u_i} \right\rfloor,$$
$$b_i(t) = t + \left\lfloor \frac{I_i(t) - SSF \cdot C_i}{u_i} \right\rfloor.$$

Given these delivery windows, we can define all customers for which $b_i(t) = t$ as must-go customers for the considered day and all customer for which $a_i(t) \leq t < b_i(t)$ as may-go customers. The chosen set of orders is then a subset of W, which should include all must-go customers and possibly a subset of the may-go customers. For every day in the planning period we have that:

$$\sum_{s=a_i(t)}^{b_i(t)} x_i(s) \geq 1 \qquad \forall i \in W,\ \forall t \in T.$$

Besides these inventory aspects, there are also costs involved in delivering to customers. A travel cost c_{ij} is defined between each pair of vertices $i, j \in V$, $i \neq j$, which is a weighted average of travel distance and travel time. We assume the travel costs to be symmetric such that $c_{ij} = c_{ji}$, $\forall i, j \in V$. Define $y_{ij}(t) \in \{0, 1\}$ as the binary variable that indicates whether an edge in the graph is used on day t or not. This means that every customer that is delivered on day t has one ingoing and one outgoing edge:

$$\sum_{j \in V} y_{ij}(t) = x_i(t) \qquad \forall i \in W, \forall t \in \{1, T\},$$
$$\sum_{j \in V} y_{ji}(t) = x_i(t) \qquad \forall i \in W, \forall t \in \{1, T\}.$$

Furthermore we have a set of routes $R(t)$, where we define a route $r \in R(t)$ as a set of consecutive edges starting and ending at the depot $\{(0, i), (i, j) \ldots, (k, l), (l, 0)\}$, such that $(i, j) \in r$ implies that $y_{ij}(t) = 1$. Note that there are infinitely many vehicles available, however a fixed cost f is incurred each time a vehicle is used. Moreover each vehicle has a maximum capacity of Q, resulting in a capacity constraint

on the customers delivered in a route:

$$\sum_{\substack{(i,j)\in r \\ i\neq 0}} q_i(t) \leq Q \qquad \forall r \in R(t),\ \forall t \in T.$$

The total cost of a day then depends on which routes $R(t)$ are constructed and the corresponding fixed costs and travel costs:

$$\text{cost}(t) = \sum_{r\in R(t)} \left(f + \sum_{(i,j)\in r} c_{ij} \right). \tag{2.4}$$

For every day t of the horizon T, we want to determine the best set of customers $O(t)$ that should be delivered on this day ($x_i(t) = 1 \iff i \in O(t)$), such that we do not violate any constraints and such that the cost per volume over the entire horizon is minimized. Filling in the long-term objective as specified in Section 2.2.3 with Equation 2.4 representing the daily cost and Equation 2.3 representing the daily volume, gives us the following objective to minimize:

$$\text{long-term objective}(T) = \frac{\sum_{t=1}^{T} \sum_{r\in R(t)} \left(f + \sum_{(i,j)\in r} c_{ij} \right)}{\sum_{t=1}^{T} \sum_{i\in O(t)} q_i(t)}. \tag{2.5}$$

3 Literature Review

This chapter provides an overview of the literature relevant to this research. We start by discussing the scope of this overview in Section 3.1 and end with a conclusion on the most useful aspects for this research in Section 3.5. In between we present background information on the Inventory Routing Problem in Section 3.2, we discuss various problems that include an aspect of order selection in Section 3.3 and we focus on the long-term performance of short-term solution methods for the IRP in Section 3.4.

3.1 Scope of the Literature Review

This literature review is meant to provide background information on the topics of inventory routing and order selection. The solution methods from other studies that are discussed here, are mainly used as an inspiration for the actual solution methods as described in Chapters 4 and 5. The aspects that are actually applied will be discussed again shortly in these chapters.

The structure of this review is based on our research goal. As stated in Section 2.5, the goal is to identify opportunities to improve the *long-term performance* of ORTEC *Inventory Routing* by using different algorithmic approaches for the *selection of orders*. Based on the emphasized aspects we formulate three information questions for this literature review.

1. *Inventory Routing:* What are the challenges when solving the Inventory Routing Problem and what are the most common used solution methods?
2. *Selection of orders:* Which methods for order selection have been applied in the IRP and in other similar problems?
3. *Long-term performance:* How can the long-term performance be optimized in the IRP when a daily solution approach is used?

We address the first question in Section 3.2. Note that there are many different structural variants of the IRP known in the literature. We start by shortly evaluating these different variants, but we merely focus on solution methods for the IRP as it is solved in OIR and defined in Section 2.2.1. Section 3.3 discusses the second question on order selection. Here we first focus on solution methods for the IRP that solve the problem in multiple phases, thereby separating the decision on which customers to deliver from constructing the routes. We first focus on order selection in similar problems that also deal with customers or nodes that are either optional or required to visit. Section 3.4 last examines the question on ensuring the quality of the long-term performance, while the IRP is solved on a day by day basis.

3.2 Inventory Routing Problem

As mentioned before in Section 2.2.1, the problem of combining inventory management and vehicle routing was first introduced by Bell et al. [2] more than thirty years back. Since this introduction there has been a considerable amount of research on the Inventory Routing Problem (IRP). The problem emerges in many different industries, with a large variety of different problem variants. Almost every new study on the subject reviews a different version of the problem as is shown in the two most recent literature surveys on IRPs of Andersson et al. in 2010 [1] and of Coelho et al. in 2014 [3]. Most papers make a distinction in structural variants of the problem, based on criteria like structure, inventory policy and fleet characteristics. The aspects that influence the chosen solution method most, are the assumptions about the demand and the objective. In this research we assume a deterministic demand and we consider only transportation costs for our objective, without taking inventory holding costs into account. We seek to find literature that is largely in line with these assumptions.

The IRP is a generalization of the Vehicle Routing Problem (VRP) that constructs the optimal set of routes for a fleet of vehicles in order to deliver a specified set of customers. Since the VRP is known to be NP-hard [4], the IRP is NP-hard as well. Examples of recent exact approaches are those of Coelho and Laporte [5] and Adulyasak et al [6], who propose an ILP formulation that uses branch-and-cut strategies to find an exact solution. Instances with up to 45 customers, three periods and three vehicles have been solved to optimality with CPLEX using this formulation. However most research is focused on robust heuristics that provide good solutions quickly, since exact solutions are only capable of solving small instances. In this research we are mainly interested in practical applications, so we focus on heuristic approaches. Recently, Archetti et al. [7] analyzed the benefits of integrating inventory management and fleet management by comparing the situation where customers decide their own delivery moments to the situation where the supplier determines the replenishment schedules. The results show that the savings are on average 9.5%, even when a heuristic algorithm is used to construct the integrated policies. This shows that the benefits of considering the integrated problem overcome the limitations due to the problem complexity, provided the quality of the heuristic for the integrated problem is reasonable.

When we consider heuristics for the IRP, the distinction between an integrated or a decomposed approach can be made. Most integrated approaches are based on the concept of metaheuristic which apply local search procedures and a strategy to avoid local optima, and perform a thorough evaluation of the search space. Examples of this approach in IRPs are iterated local search, variable neighborhood search, greedy randomized adaptive search, memetic algorithms, tabu search and adaptive large neighborhood search [3]. In a decomposed approach the IRP is decomposed into hierarchical subproblems, where the solution to one subproblem is used in the next step. The most common decomposition consists of two phases, where the first phase plans which customers to deliver and how much to deliver them, and the second phase constructs the delivery routes. Examples of such a decomposition approach can be found in the research of Campbell et al. [8] and Kooijman [9]. Since our research focuses on the decision when to deliver a customer and not on constructing the delivery routes, we are mainly interested in such decomposition approaches. Research that focuses on this selection of orders for delivery separately will therefore be discussed in Section 3.3.1.

The IRP has a long-term nature, but in practical usage a reduction to a short-term horizon is often crucial. One method in the literature is to construct fixed, periodic solutions that can be repeated over the horizon. These approaches are often referred to as IRPs with an infinite horizon and an overview

can be found in [1] or [3]. Other approaches assume a finite horizon and construct solutions for many days ahead simultaneously. However in ORTEC Inventory Routing we assume that the actual schedule of orders and routes is made only one day ahead, because of the availability of information. The most common way to handle long-term effects in this case, is to use a rolling horizon and solve the problem for a longer period than is actually needed for the immediate decision. This and other methods to take the long-term effects of our short-term decisions are discussed in Section 3.4.

3.3 Order Selection in Various Problems

Our goal of this research is concerned with improving the decision on which customers to visit on the upcoming day and which not, within the inventory routing problem. There are many problems that also incorporate such a selection aspect and we are interested in whether we can use similar approaches. An example arises when a transportation company has to construct delivery routes, but in addition has the possibility to outsource orders to another transportation company for a specific cost. This problem is referred to as the Vehicle Routing Problem with Outsourcing (VRPO) or the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC) and is discussed in Section 3.3.2. Besides the VRPO, another class of similar problems are the Prize-Collecting Problems or the Profitable Tour Problems [10]. The key characteristic of this class of problems is that the set of customers to serve is not given, but still needs to be decided. Moreover each customer has a profit when delivered, but the arcs that are used to reach this customer have a cost. The objective is to maximize the profits minus the costs or minimize the costs minus the profits. We will look at the Prize-Collecting Steiner Tree in Section 3.3.3. We start however with examining approaches to the Inventory Routing Problem where the selection of orders is done separately in Section 3.3.1. This arises when the problem is decomposed into a planning phase that decides which customers to deliver and a routing phase that constructs the routes.

3.3.1 Order Selection in Inventory Routing

One of the first approaches that explicitly considers the decision about whether to visit a customer or not before the actual routing, is that of Golden et al [11]. In their approach a threshold is used to decide which customers to consider and next the planning is done based on the degree of urgency of a customer. The threshold α is the maximum relative inventory level that a customer is allowed to have before delivery takes place. If the inventory level as a percentage of the capacity is higher than α the customer is excluded from delivery. The urgency depends on the ratio between the capacity and the remaining inventory level at a customer and is used to determine the order of addition when constructing the delivery routes.

Both Dror et al. [12] and Bard et al. [13] use a planning window that consists of multiple days, which results in the problem of deciding the optimal day to deliver each customer. A single customer cost-function is used that reflects the difference in future cost between visiting the customer earlier then the latest possible day and visiting the customer on the latest possible day. This difference is based on the increase in the number of necessary deliveries when forwarding the delivery. Based on these costs, the customers are assigned to different days using an ILP, considering the available vehicles for every day.

Campbell and Savelsberg [8] also use a planning window of multiple days and solve an ILP to find which customers to serve on each day and how much to deliver them. The most important decision variable indicates whether a certain route is used on a specific day and the objective minimizes the travel costs over the planning window. However the huge number of possible delivery routes and the length of the planning horizon make this integer program not very practical. To solve this, only a small set of the delivery routes is considered and the periods toward the end of the planning horizon are aggregated. Reducing the number of routes is done by constructing clusters of customers and only allowing customers to be in the same route if they are in the same cluster. A good set of disjoint clusters covering all customers is obtained in three steps. First generate a large set of possible clusters based on geographical and capacity considerations, next estimate the cost of serving each cluster for a certain period of time sufficient for all customers to receive a delivery and last solving a set-partitioning problem to select the best clusters. To estimate the costs of a cluster an ILP is solved that minimizes $\sum_{r \in R} c_r z_r$, with parameters c_r that denote the cost of an optimal route r from the set of all routes R through a subset of customers in a cluster and decision variable z_r that indicates how often a certain route is used in the planning window.

The planning phase of Kooijman [9] mainly focuses on balancing the workload of the deliveries equally over the days in a rolling horizon framework. The first step is to assign delivery windows to customers with a high usage rate, based on the time until stock out. Next, a greedy approach is used to plan the customers with a lower usage rate, starting with the day that has the least planned delivery volume until now. Allowed customers for a certain day are the ones that still have an inventory higher than their safety stock and that can be delivered a volume higher than a certain percentage of their capacity. The allowed customer that has the smallest distance to one of the customers already scheduled on that day will be planned on that day. This way the approach tries to balance the workload but also considers the geographical distance by maximizing the volume delivered per kilometer.

Cordeau et al. [14] considers a multi-product IRP with inventory costs and solves this with a decomposition heuristic that contains an extra reoptimization phase after the planning and routing phases. The planning phase solves a mixed-inter program that minimizes the sum of the customers' inventory and transportation costs. The transportation cost of each customer is estimated as half of the cost of a direct delivery. After the routing phase the reoptimization phase expands the set of routes by combining certain routes and solves a similar mixed integer program as in the planning phase, where the only difference is that the actual distances of the found routes are used.

Summarizing we see that many approaches are influenced by the number of trucks available therefore by the need to balance the customers over the days to divide the workload equally ([12], [13], [9]). Since this constraint is not applicable to our research where we assume infinite fleet size, we do not want to adopt these approaches directly. The same holds for the approach used in [8], where global clusters of customers that have to be delivered together are generated for a longer period of time. This mechanism is not very flexible and is only suitable when the used information is mainly static, which is not the case in ORTEC Inventory Routing. Generating clusters of customers can be useful in our research when the clusters are flexible and can be recalculated fast.

3.3.2 Order Selection in Vehicle Routing with Outsourcing

In the Vehicle Routing Problem with Outsourcing (VRPO) there is the possibility to outsource order to another transportation company for a predefined cost, before the delivery routes are constructed. We distinguish two global approaches to this problem. In the first approach heuristics are used to construct the routes and the decision whether to outsource certain customers or not is taken simultaneously, for example with tabu search or large neighborhood search [15]. The second approach focuses on fast and simple rules to guide the decision without constructing the actual routes [16].

We are especially interested in the second approach, since this approach is similar to the order selection in the IRP. The main difference is the predefined cost for outsourcing an order, which is not known in the order selection in the IRP. Whether an order should be outsourced depends on the expected routing costs of the order and its outsourcing costs. However the expected routing costs of the order depends on the other orders which are not outsourced. This is the case for all orders, which creates an interdependence between them. Furthermore it could be that an order is outsourced even when the expected routing cost of an order is smaller than the outsource cost, because of capacity constraints and other orders that are even more profitable. We distinguish two aspects in this solution approach for the VRPO, estimating the cost corresponding to delivering a customer and selecting the most profitable ones for delivery.

Estimating the Cost of a Customer

The research of Huijink [16] compares five different studies to estimate the route costs for the classical Vehicle Routing Problem (VRP). Each method splits the estimation into three separate parts. The fixed cost, which is based on the startup cost of a route, the stem cost, which estimates the cost of driving to the area in which the order lies and and the inter-order cost, which estimates the cost for driving between customers. We discuss the two studies that are most in line with this research into more detail. The solution methods of Huijink [16] and Goudvis [17] only differ in the way they estimate the inter-order cost. The fixed cost FC_i and the stem cost SC_i for customer i are in both studies calculated based on which part of the vehicle capacity is used by the customer. This results in the following formulas:

$$FC_i = f \cdot \frac{q_i}{Q}$$
$$SC_i = 2c_{0i} \cdot \frac{q_i}{Q}.$$

Here q_i is the delivered volume to customer i, Q the vehicle capacity, f the fixed cost of a route and c_{0i} the cost of traveling from the depot to customer i. The inter-order part depends on which other orders will be in the routes, which is not yet decided at this point. Every study has its own method of dealing with this uncertainty and determining the characteristics of the orders. Huijink uses the insertion distance between orders to estimate the inter-order costs. He compares two approaches where either the five or the ten closest orders are considered. Subsequently all possible insertion distances for order i between each pair of neighbors are calculated and the inter-order cost (IOC) is determined by the weighted average. When we define the travel cost between customer i and j by c_{ij}, the corresponding formula looks as follows:

$$IOC_i = \frac{1}{\sum\limits_{k,l \text{ pair}} \frac{1}{1+c_{ki}+c_{il}-c_{kl}}} \sum_{k,l \text{ pair}} \frac{c_{ki}+c_{il}-c_{kl}}{1+c_{ki}+c_{il}-c_{kl}},$$

The method of Goudvis also uses a weighted average to determine the inter-order cost for customer i, but instead of considering the five or ten closest orders, it only considers those customers that have a high probability of being in the same route as customer i. This probability is unknown, so for each customer j a weight $w_{ij} \in \{0, 1\}$ is determined that depends on aspects like travel cost, volume and frequency. The weight is calculated by multiplying factors that represent the mutual travel costs, the fit of the volumes in a vehicle, the similarities between the delivery frequencies and other possible constraints. The weight factor for travel costs is for example given by $\frac{1}{1+c_{ij}}$. The final inter-order cost is then determined as a weighted average of the distances between the customers and its N highest weighted neighbors. The number of considered customers is determined by a stopping criteria that depends on the characteristics of the dataset.

$$IOC_i = \frac{\sum_{j=1}^{N} w_{ij} c_{ij}}{\sum_{j=1}^{N} w_{ij}}$$

where the customers j are sorted according to maximum weight.

Selecting the Most Profitable Customers

Once the costs are estimated for each customer i, a score s_i can be calculated that represents the profitability of an order. This score depends mainly on the difference between the estimated costs and the outsource price, p_i.

$$s_i = \frac{p_i - (FC_i + SC_i + IOC_i)}{q_i}.$$

To select which orders to outsource and which to include in the routes, the orders are first ranked in a descending manner according to s_i. Next the orders are sequentially divided into disjoint batches that are as big as possible under the constraint that for each new batch it holds that the sum of the capacities of the orders in this batch and all its previous batches is smaller than the number of batches times the vehicle capacity. The last step is to select all batches for which the sum of the scores of the orders is positive. Hence, full truckloads of orders are added until the cost of adding such a set of orders is higher than the outsourcing costs for these customers.

Relevance VRPO with respect to the IRP

The approach where a fixed, stem and inter-order cost are calculated in order to estimate the cost of delivering to a customer can also be applied in solution methods for the IRP. The selection procedure is however less suitable, because there are no well defined outsourcing costs available in this case. Huijink [16] compares the different estimation methods based on quality of solution, correctness of outsourced orders and quality of estimation. The method of Huijink itself on average has the best quality of solution and the highest percentage of correct outsourced orders, while the method of Goudvis has on average the best quality of estimation. The quality of the estimation method is most relevant to our research, since we cannot directly use the selection procedure. Besides this, we note that the study of Goudvis introduces weight factors which indicate whether customers are likely to be in the same route. This might be a useful aspect when deciding which may-go customers are best to combine with the set of must-go customers.

These considerations result in the conclusion that the estimation method of Goudvis would be most suitable to use in the order selection for the IRP. When this method is used, a selection method needs to be constructed that takes the interdependence of the costs of the orders into account. Besides this, it should be taken into account that the intention of the weight factors is to express a long-term compatibility, while we are more interested in a compatibility for the upcoming day. This means that the precise definitions of the weights need to be redefined.

Last, the recommendations of Huijink note that the current way of selecting the customers ignores the interdependence between the orders and one way to solve this would be to use each method a few times, while only the orders that are delivered are allowed to be used in the new cost estimation. Other follow up research that is suggested is to use a minimum spanning tree to estimate the cost of a set of orders.

3.3.3 Order Selection in a Prize-Collecting Steiner Tree

The Prize-Collecting Steiner Tree (PCST) consists of a set of terminal nodes and a set of Steiner nodes. The edges between the nodes have costs and the nodes itself each have a prize that represents the potential revenue of adding this node to the solution. The goal is to find a connected subgraph that includes all terminal nodes and possibly some Steiner nodes, that maximizes the profit, which is defined as the prizes minus the costs. The PCST is a generalization of the Steiner Tree and is therefore NP-hard. Literature shows that exact approaches to the PCST are only suitable for small instances. Most developed heuristics use a similar strategy with a construction phase and a pruning phase. In the construction phase a Minimum Spanning Tree (MST) is constructed on all nodes, which is a subgraph that connects all vertices together using edges of minimal total cost. Then in the pruning phase leaf nodes for which the prize is smaller than the costs are pruned.

Construction and Pruning

In the research of Hosokawa and Chiba [18] edges are iteratively added in order of maximum profit, which is the prize of adding the node minus the costs of adding the node. When all nodes have been added, a reverse delete strategy is applied, where for each node it is computed whether it is more profitable to keep this node or to delete it. If this node has leaf nodes, the deletion of these nodes is also taken into account when calculating which action is more profitable. In this research no distinction is made between terminal nodes and Steiner nodes, however this distinction can be taken into account by assigning very high prizes to the terminal nodes, such that it is always profitable to include those nodes. This will result in adding the terminal nodes first, before any of the Steiner nodes.

The approach of Akhmedov et al. [19] consists of two phases. The first phase is an iterative process, where in each iteration more Steiner nodes are added by constructing an MST, until no improvement is possible anymore. The second phase tests all leaf nodes of the resulting graph for pruning. If the prize of the leaf node is smaller than the connection costs, the node is deleted from the graph. In this research the definition of Steiner nodes and terminal nodes differs from other formulations of the PCST, namely terminal nodes are simply the nodes with a positive prize and Steiner nodes the rest of the nodes. This means that terminal nodes can be deleted in the pruning phase, which will also delete all Steiner nodes connecting this terminal node to the tree.

Relevance PCST with respect to the IRP

When we define the delivery volumes of the customers as prizes of the corresponding nodes, we can use an approach of construction and pruning to selecting which customers to deliver in the IRP. However, in the IRP we want to optimize the ratio between costs and prizes instead of the difference between them. This variants arises in the Fractional Prize-Collecting Steiner Tree (FPCST), a problem that is studied less than the normal PCST. The only research that is known on this problem is the one by Klau et al. [20], which proposes a parametric formulation for FPCSTs on trees. The study uses Newton's iterative method, where several linear PCST instances have to be solved to optimality for obtaining one solution for the FPCST. Another note to this approach is that the PCST does not consider aspects like vehicle capacity. Taking this extra constraint into account makes the problem more difficult.

3.4 Long-Term Performance in Inventory Routing

According to Song et al. [21], the most popular performance measure that is used in practice to evaluate distribution strategies, is the cost per volume, sometimes also referred to as the logistics ratio (LR). Optimizing the cost per volume instead of the costs allows for better anticipations, but does not necessarily lead to optimal long-term solutions as we have seen in Example 2.2.2. The literature on IRPs describes some strategies to take long-term performance into account, even when a day by day solution method is used. A good indicator of the impact of a short-term decision on the long-term cost according to Dror and Ball [12], is the relative delivery size, which is the fraction of the volume that can be delivered at that moment compared to the capacity. Campbell et al. [8] state that the quantity delivered to a customer should always be maximized, because this minimizes the number of visits to a customer in the long-term. Maximizing the vehicle utilization is also an important aspect, since transportation costs are better used when the vehicle is as full as possible. Two methods discussed in the literature that incorporate the long-term effect even better are the use of a rolling horizon and the use of an adapted short-term objective.

Rolling Horizon

A popular way to incorporate the long-term effect, is to use a rolling horizon and solve the problem for a longer period than is actually needed for the immediate decisions. An example might be to decide on the deliveries and volumes of customer for an entire week, after which the delivery routes are only constructed for the first day. In the next iteration the planning window will then shift by one day and new information is used to plan the deliveries for days two to eight. Bard et al. [13] use a rolling horizon framework where the planning is executed for two weeks and the routing for one week, after which the planning window is shifted with one week. In this case only customers with the optimal replenishment time within the planning period are assigned a delivery day. Other examples of the use of a rolling horizon can be found in [12], [22] and [9].

Adapted Short-Term Objective

Benoist et al. [23] define a surrogate short-term objective that can be summarized into the following rule: "Never put off until tomorrow what you can do optimally today." This objective focuses on minimizing the global extra costs per volume, compared to the optimal cost per volume. To accomplish this, a lower bound for the cost per volume of a customer needs to be defined first. A simple lower bound on the minimum cost per volume of customer i would be the cost of the cheapest trip visiting customer i by the maximum amount that can be delivered. When the capacity at a customer is higher than the vehicle capacity, the optimal distribution strategy is to always deliver a full truckload to the customer. This results in the following lower bound on cost per volume for customer i

$$LB(i) = \frac{2c_{0i}}{Q}.$$

Here c_{0i} is the cost of traveling from the depot to customer i and Q is the vehicle capacity. The optimal route costs (RC^*) for a route r can now be calculated as $\sum_{i \in r} LB(i) \cdot q_i$, with q_i being the delivered volume to this customer in this route. The surrogate logistic ratio of Benoist et al. (LR_{BEN}) is defined as

$$LR_{\text{BEN}} = \frac{\sum_{r \in R}(RC(r) - RC^*(r))}{\sum_{r \in R} \sum_{i \in r} q_i},$$

and represents the global extra cost per volume over all routes R compared to the lower bound.

Another long-term objective function is developed by Singh et al. [24]. First of all they note that the lower bound used by Benoist et al. may not be realizable in practice for some customers, so they introduce a practical lower bound (PLB) that is always achievable and represents a realistic cost of delivering volume to a customer. Instead of taking the vehicle capacity as the maximum amount that can be delivered, they calculate the maximum delivered quantity based on the capacity of the customer C_i and the safety stock fraction SSF, since it is assumed that delivery always happens before safety stock is reached.

$$PLB(i) = \frac{2c_{0i}}{\min\{(1 - SSF) \cdot C_i, Q\}}.$$

Their second observation is that the SLR is quite sensitive to the LB and encourages sub-optimal deliveries to may-go customers, because it does not consider inter-delivery distances among customers in a route for calculating the optimal route cost. Therefore they define a long-term logistic ratio that focuses on the optimality of each customer in the route instead of only the route optimality. To achieve this the total costs of a route are allocated to the customers in it, based on urgency to avoid stock outs. This means that must-go customers are allocated most of the costs and may-go customers are only allocated the marginal increase. This marginal cost increase is calculated for each may-go customer as the difference between route costs when the customer is or is not included. The cost of the must-go customers is then calculated by dividing the remaining route costs over all must-go customers in the route according to distance and quantity. The logistic ratio for each customer ($LR(i)$) can then be calculated by dividing these allocated costs by the volume delivered to this customer. The final objective of Singh et al. (LR_{SIN}) can finally be defined as

$$LR_{\text{SIN}} = \sum_{r \in R} \sum_{i \in r} (LR(i) - PLB(i)),$$

and represents the extra cost per volume for each customer compared to its lower bound.

3.5 Conclusions of the Literature Review

The literature provides us with some insights on used solution methods regarding the decision whether or not to deliver to a customer. An important aspect in deciding if a customer is profitable or not is the estimation of the delivery costs. Promising methods in the literature suggest that this can be done by estimating the fixed, stem and inter-order costs for each order separately ([16], [17]). Other approaches include a comparison in costs between visiting a customer or not visiting a customer ([12]), or suggest a cost estimation with a minimum spanning tree ([16]).

To deal with the interdependence between orders when estimating the costs and selecting the orders, calculations can be repeated iteratively while including new orders in each iteration ([16]). Besides this, sorting customers on either urgency ([11]) or expected profit ([16]), is useful to sequentially decide on which customers to add. Heuristics for the Prize-Collecting Steiner Tree are divided into a construction phase that adds promising customers and a pruning phase that removes customers who turn out to be nonprofitable. The sorting of customers, when deciding which not to include, is done in backward order of profitability ([18], [19]). One remark that should be made is that the described methods cannot be applied directly, since they assume a linear objective function, while we are interested in a fractional one, namely the cost per volume.

Regarding the long-term performance of a day by day approach we found that the most common used approach is to incorporate a rolling horizon framework with a planning window that is bigger than just one day ([12], [13], [9]). Besides this we found that the cost per volume is a good performance measure for the long-term, however optimizing the cost per volume for a short-term planning window may not lead to long-term optimal schedules. Objectives that reflect the difference between the optimal cost per volume and the attained cost per volume provide a better measure to take the effect of short-term decisions on the long-term performance into account ([23], [24]).

4 Short-Term Solution Approach

As explained in Section 2.5, our goal is to design a decision mechanism that selects the best customers for delivery on the upcoming day, while minimizing the long-term cost per volume. This chapter provides the algorithmic solution approach when we aim to minimize the short-term objective of OIR, the cost per volume of the upcoming day. First we provide a general overview of this approach in Section 4.1. Then we suggest two methods for the order selection in Sections 4.2 and 4.3

4.1 Solution Overview

Given a certain day and a set of must-go and may-go customers, the goal in this short-term solution approach is to find the set of orders that minimizes the daily cost per volume as stated in Equation 2.2. We aim to improve the current selection of customers as done by OIR for each single day, however in designing this new approach we do not yet take the long-term performance into account. Improving the long-term objective directly will be the focus in Chapter 5. For the short-term approach we are interested in how much we can improve the daily cost per volume using different selection methods. Besides this we want to know the effect of better short-term results on the total cost per volume over the horizon.

In the current situation of OIR, the order generation provides us with a set of must-go customers and a set of may-go customers. In the new short-term approach we use this information as input and decide which may-go customers are most profitable to add for delivery on the upcoming day. In Examples 2.3.1 and 2.3.2 we identified two main aspects that allow for improvement. First, the construction of the initial solution, which is currently only based on the must-go customers. Second, the order of addition of the may-go customers, which is currently fixed with respect to the closest distance to one specific order of the route. For both aspects it holds that better solutions might arise when the most profitable may-go customers are selected beforehand, without the construction of the actual delivery routes. The

Whether a delivery to a may-go customer is profitable depends on the travel costs involved in the delivery and the maximum delivery volume of this customer on this day. This means that we need to estimate this cost in order to identify the most profitable customers. One problem that we encounter with this, is that the delivery costs of customers are interdependent. They heavily depend on which other customers are delivered and whether the customer will be delivered alone or can be combined in a route with neighbors. For this reason it is not straightforward to identify which set of customers attains the minimum cost per volume and we have to compare different sets of orders to determine this. So to summarize we see that the delivery costs of the customers depend on the selection of orders and the selection of orders depends on the delivery costs of the customers. In our solution approach we therefore want interdependent

mechanisms to estimate the costs of delivering to a customer and to select the orders. In Sections 4.1.1 and 4.1.2 we discuss such mechanisms into more detail.

4.1.1 Delivery Cost Estimation

The costs involved in delivering a certain set of orders can be found by solving a Vehicle Routing Problem (VRP). Since the VRP is an NP-hard problem and we need to solve it multiple times to compare different sets of orders, this would be computationally heavy. However to make a global comparison we are not interested in the exact sequencing of orders in routes, so we do not want to take such aspects into consideration explicitly. We want an approach that roughly estimates the delivery costs of orders. We propose two different estimation methods that each have their advantages and disadvantages. One where we evaluate the cost of each customer separately and one were we form clusters of customers. Since it is our goal to use these methods as a subroutine when selecting the best set of orders, the methods should be able to quickly evaluate the cost of a given set of customers and give a reasonable representation of the costs that provides us with the possibility to compare different sets of orders.

The first method that we consider obtains an estimation of the delivery cost for each customer separately. It is inspired on estimation methods that have also been used in the research on the Vehicle Routing Problem with Outsourcing (VRPO) as described in Section 3.3.2. We estimate a fixed part, a stem part and an inter-order part to represent the total expected cost of delivering a certain customer on a certain day. The final value of this cost mainly depends on the travel cost to the depot, on the delivery volume compared to the vehicle capacity and on close neighbors. This approach is described into more detail in Section 4.2.

Another possible approach is to take the clusters of customers, which will be formed because of the vehicle capacity, into account when estimating the costs. The orders with which a certain customer is combined, highly influences the final cost of delivery. The sizes of the volumes compared to the vehicle capacity and to each other play a big role in how many and which routes will eventually be constructed and hence in the resulting costs. Therefore we consider a second approach where we take volumes and vehicle capacity into account to form clusters of customers, but where we do not consider sequencing in routes. This means that we want to construct clusters of customers that are a good match and are highly likely to end up in the same route, based on both geographical and capacity considerations. This approach is described into more detail in Section 4.3.

4.1.2 Order Selection

We want to design a mechanism that does not enumerate all sets of orders, but only explores promising combinations. We want to use the two estimation methods that have been defined in Section 4.1.1 as a subroutine in determining which set of orders has the lowest cost per volume, and design for each method a suitable order selection. By not constructing the actual delivery routes in this process, we avoid the problems as mentioned in Examples 2.3.1 and 2.3.2. Section 4.2.2 presents the order selection when we evaluate the cost of each customer separately and Section 4.3.2 discusses the order selection when clusters are formed int he estimation process.

4.2 Order Selection with Separate Customer Evaluation

Our goal is to find the set of customers that results in the minimum cost per volume when delivered. To express which customers are delivered and which not, we introduce the probability of delivering a customer, p_i, which depends on its estimated cost per volume. We estimate the cost of delivering to a customer based on its volume, the distance to the depot and its close neighbors, however its neighbors again depend on the probabilities of delivery to the other customers. We therefore suggest a recursive definition for the probabilities of delivery to the customers. In this recursive definition all probabilities eventually converge to either zero or one. The selected orders will be the customers with probability one. Section 4.2.1 proposes the formulas to estimate the costs for delivery to the customers and Section 4.2.2 explains how to determine the probabilities.

4.2.1 Fixed, Stem and Inter-Order Costs

As described in the literature in Section 3.3.2 we can estimate the cost of an order with a fixed part, a stem part and an inter-order part. The fixed cost (FC) represents the start up cost of the route, the stem cost (SC) represents the cost of traveling to the area of the route where the customer is in, and last the inter-order cost (IOC) represents the cost to travel the extra distance between the customers. We define the formulas for these costs similar to the formulas used in Goudvis [17], however we adjust the formula for the inter-order cost by introducing probabilities for delivery of each customer. The most important parameters that are used in the calculation of these cost factors for customer i are its delivered quantity q_i, the available vehicle capacity Q and the travel cost c_{ij} for $j \in \{0, 1, \ldots, n\}$ with $j = 0$ representing the depot.

The total fixed cost of a route f should be divided over all customer delivered by the vehicle. However at this point we do not yet know how many customers will be in the same vehicle as customer i. We do know the maximum capacity that fits in the vehicle, so we divide the fixed cost in proportion to the volume that is delivered to this customer. This gives us

$$FC_i = f \cdot \frac{q_i}{Q}. \tag{4.1}$$

Note that this value is an underestimation of the actual total fixed cost since it assumes that the total vehicle capacity will be used, which is not always the case. However we are especially interested in using the estimate to compare the costs of different customers. Moreover the underestimation is compensated by the inter-order cost defined later, which is an overestimation. We use the same principle for the stem cost. In this case the stem value for the entire route is also uncertain, so for this we consider the cost of traveling from the depot to the customer and back. This results in

$$SC_i = 2c_{0i} \cdot \frac{q_i}{Q}. \tag{4.2}$$

The definition of the inter-order cost (IOC) is less straightforward, because it depends on the distance between orders delivered by the same vehicle, while it is not yet known which customers this will be. For this reason we are interested in finding the orders that are likely to be in the same route as customer i. To achieve this we determine a probability $p_{ij} \in \{0, 1\}$, for every other customer j, that expresses the

likelihood of customer i and j ending up in the same route. This probability is based on different aspects, like geographical location, delivery quantities and the whether customer j is likely to be delivered on that day. The calculation of this probability will be explained in Section 4.2.2.

Given these probabilities p_{ij} for all combinations of customers, the inter-order cost of customer i can be calculated as a weighted average of the direct distances to the neighbors that have the highest probability of ending up in the same route as this customer. This means that we can sort all customers according to decreasing probability, select the m best ones and calculate the weighted average distance to those customers. Based on the literature and some computational experiments we fix the number of neighbors to consider at five.

$$IOC_i = \frac{\sum_{j=1}^{m} p_{ij} \cdot c_{ij}}{\sum_{j=1}^{m} p_{ij}}. \tag{4.3}$$

Note that it can happen that p_{ij} is zero for all neighbors j of customer i, for example because the delivery quantity of customer i is equal or almost equal to the vehicle capacity and it cannot be combined in a vehicle with any other customer. In this case we do not want to base the estimate on fixed, stem and inter-order costs. Since we know that the customer will end up in a vehicle by itself, we can determine the cost even more precise. In this case the delivery cost is the cost for direct delivery, consisting of the fixed cost and twice the distance to the depot: $f + 2 \cdot c_{0i}$. Combining this principle with Equations 4.1, 4.2 and 4.3 gives us the the following expressions for the delivery cost (DC) of a customer i.

$$DC_i = \begin{cases} \left(f \cdot \frac{q_i}{Q}\right) + \left(2c_{0i} \cdot \frac{q_i}{Q}\right) + \frac{\sum_{j=1}^{m} p_{ij} \cdot c_{ij}}{\sum_{j=1}^{m} p_{ij}} & \text{if } \sum_{j=1}^{x} p_{ij} \neq 0, \\ f + 2c_{0i} & \text{else.} \end{cases} \tag{4.4}$$

4.2.2 Initializing and Updating of the Probabilities

The inter-order cost for customer i is based on the probability p_{ij} that customer j is in the same route as this customer i. We can express this probability as the product of two factors. The first is the probability that j is in the same route as customer i, given that j is delivered today, we will refer to this as the weight factor w_{ij}. The second is the probability that j is delivered today, we will refer to this as the delivery probability p_j. This gives us the following relation for the probability that i and j are in the same route:

$$p_{ij} = w_{ij} \cdot p_j.$$

The weight factor w_{ij} should express how well customers i and j fit together in a route. This depends on aspects like the delivery volumes and the geographical locations. The delivery probability p_j should represent whether delivering customer j today is profitable. This mainly depends on the cost per volume of this customer. First we initialize p_{ij} by calculating w_{ij} and assigning an initial value to p_j for all customers. Then we update the values for p_j iteratively until each value is either zero or one. We finally select the customers that have probability one as the orders. The code for this can be found in Algorithm 1. We discuss how to determine the weight factors and the delivery probabilities in the next paragraphs.

Algorithm 1 Order Selection using Fixed, Stem and Inter-Order Costs.

```
OrderSelection(customers)
    { w_ij | i, j ∈ customers } ← CalculateWeightFactors(customers)          ▷ Eq. 4.5
    { p_j(0) | j ∈ customers } ← InitializeDeliveryProbabilities(customers)  ▷ Eq. 4.6
    n = 0
    while ∃j ∈ customers : 0 < p_j(n) < 1 do
        { p_j(n + 1) | j ∈ customers } ← UpdateDeliveryProbabilities(n, customers)  ▷ Alg. 2
        n = n + 1
    end while
    orders = { j | j ∈ customers, p_j(n) = 1 }
    return orders
end
```

Determining the Weight Factors

We calculate w_{ij} as a product of multiple factors that influence the likelihood of customers i and j being in the same route, for example whether the customers fit together in a vehicle, whether they are located close to each other and whether there are objections on combining these customers in a vehicle.

First we calculate a quantity factor $\alpha_{ij} \in \{0, 1\}$ that represents how well the delivery quantities match. If the sum of the quantities is bigger than the vehicle capacities the customer cannot fit in the same vehicle, so this quantity factor should be zero. Otherwise the customers fit together in a vehicle, so we set this factor to one.

$$\alpha_{ij} = \begin{cases} 0 & \text{if } q_i + q_j > Q \\ 1 & \text{else} \end{cases}$$

Secondly we look at the travel cost between two customers and calculate a cost factor β_{ij} that increases as customers are located more closely. This formula is derived from the approach of Goudvis and it takes into account that we are interested in the relative distance difference between different customers instead of the absolute difference.

$$\beta_{ij} = \frac{1}{1 + c_{ij}}.$$

Another factor that we introduce, γ_{ij}, is used to express the correlation between two customers based on historical information and is based on the number of times customer i and j where in the same route, m_{ij}. Besides this it also depends on the number of times customer i and j were delivered on the same day. Remember from Section 2.6 that we can express this as $\sum_{s=0}^{t} x_i(s) \cdot x_j(s)$. The ratio of m_{ij} and $\sum_{s=0}^{t} x_i(s) \cdot x_j(s)$ represents whether customer i and j are likely to be in the same route if they are delivered on the same day. However when the number of times that i and j where delivered on the same day is small, this factor is not very reliable and might depend too much on the specific characteristics of this day, like the delivery volumes or the other customers. Therefore we propose a threshold that prevents this factor from having a big effect if there is too little information available on the customers. This threshold depends on the number of times that i and j have been delivered on the same day until

now. Combining all these aspects gives us the following formula:

$$\gamma_{ij} = \begin{cases} \dfrac{m_{ij}}{\sum_{s=0}^{t} x_i(s) \cdot x_j(s)} & \text{if } \displaystyle\sum_{s=0}^{t} x_i(s) \cdot x_j(s) \geq 5 \\ 0.5 & \text{else} \end{cases}$$

In the same way we can add even more factors. We introduce $\delta_{ij} \in \{0, 1\}$ which is zero if there are constraints that prevent these customers from being in the same route. An example of such a constraint could be that they are only allowed to be transported in different vehicles. However in this research we disregard such constraints and fix δ_{ij} at one for every combination of customers. Once all factors have been calculated we multiply them to represent the likelihood of customers i and j being in the same route, if both customers are delivered today.

$$w_{ij} = \alpha_{ij} \cdot \beta_{ij} \cdot \gamma_{ij} \cdot \delta_{ij}. \tag{4.5}$$

Note that this is merely a relative weight that expresses the preference of j being in the same route as i, compared to other customers being in the same route as i, since we normalize the values in Equation 4.3. This weight will be zero when the customers do not fit together in a vehicle, it will be small when the customers are located far away from each other or have never been in the same route before and it will be big if they are located close or have been in the same route often.

Determining the Delivery Probabilities

By now we determined a weight factor w_{ij} that expresses whether two customers are favorable to combine, however we have to keep in mind that only profitable orders will be delivered. If w_{ij} is high, but customer j itself is not profitable enough to deliver, it should not be taken in to account when calculating the cost of delivering customer i. Therefore we also take the probability p_j that customer j is actually delivered into account.

We use a recursive definition for the probabilities to select which orders to deliver. On initialization we know that all must-go customers will be delivered. Since we do not know the delivery costs of the may-go customers, we assume that they are all equally likely to be delivered or not. Therefore we initialize the probabilities of all must-go customers as one and the probabilities of all may-go customers as 0.5.

$$p_i(0) = \begin{cases} 1 & \text{if } i \text{ must-go customer} \\ 0.5 & \text{if } i \text{ may-go customer} \\ 0 & \text{else} \end{cases} \tag{4.6}$$

Once these delivery probabilities are initialized, Equation 4.4 provides us the corresponding costs of all customers. For each iteration n we use the delivery probabilities, $p_j(n)$, of this iteration . We redefine

this formula to incorporate the recursive aspect by substituting $w_{ij} \cdot p_j(n)$ for p_{ij}, which gives us

$$DC_i(n) = \begin{cases} \left(f \cdot \frac{q_i}{Q}\right) + \left(2c_{0i} \cdot \frac{q_i}{Q}\right) + \frac{\sum_{j=1}^{m} w_{ij} \cdot p_j(n) \cdot c_{ij}}{\sum_{j=1}^{m} w_{ij} \cdot p_j(n)} & \text{if } \sum_{j=1}^{x} w_{ij} \cdot p_j(n) \neq 0, \\ f + 2c_{0i} & \text{else.} \end{cases} \tag{4.7}$$

In the same way we can also obtain an estimate for the total cost per volume of today. We assume the following formula for the objective corresponding to iteration n:

$$obj(n) = \frac{\sum_i p_i(n) \cdot DC_i(n)}{\sum_i p_i(n) \cdot q_i}. \tag{4.8}$$

Given the estimated objective value for iteration n, we can improve the probabilities of delivery for the may-go customers for the next iteration $n+1$. If the cost per volume of a certain customer is very high compared to the objective we do not want to deliver this customer and the probability should be zero. On the other hand if the cost per volume of the customer is very low compared to the objective, delivery is profitable and the probability should be one. If the cost per volume is close to the objective value it is not yet certain whether this customer will be profitable or not since the objective is only an estimate and will change when the probabilities change.

In each iteration we update the delivery probability of exactly one may-go customers from 0.5 to either zero or one. Therefore, the number of iterations is equal to the number of may-go customers. We choose to update the customer of which the difference between the cost per volume of the customer and the total cost per volume is the largest. If the cost per volume of this customer is smaller than the current estimated objective, delivery to this customer is cost efficient so we set the probability to one. In the other case delivering to this customer will probably increase the cost per volume, so we fix the probability value to zero. In determining which cost per volume deviates most of the objective we only consider customers that have not yet been updated and still have a probability of 0.5. The pseudo code of the update of the probabilities that is executed each iteration n, can be found in Algorithm 2. Once all customers have a probability of either zero or one, we select the customers with probability one and take these as the orders.

Algorithm 2 Updating the Delivery Probabilities.

```
UpdateDeliveryProbabilities(n, customers)
    for i ∈ customers do
        DC_i(n) ← CalculateDeliveryCosts(i, n)                ▷ Eq. 4.7
    end for
    obj(n) ← EstimateObjective(n)                             ▷ Eq. 4.8
    K = { i | i ∈ customers, 0 < p_i(n) < 1 }
    x ← arg max_{k∈K} |DC_k(n) − obj(n)|
    if DC_x(n) − obj(n) ≤ 0 then
        p_x(n + 1) = 1
    else
        p_x(n + 1) = 0
    end if
    return { p_j(n + 1) | j ∈ customers }
end
```

4.3 Order Selection with Route Clusters

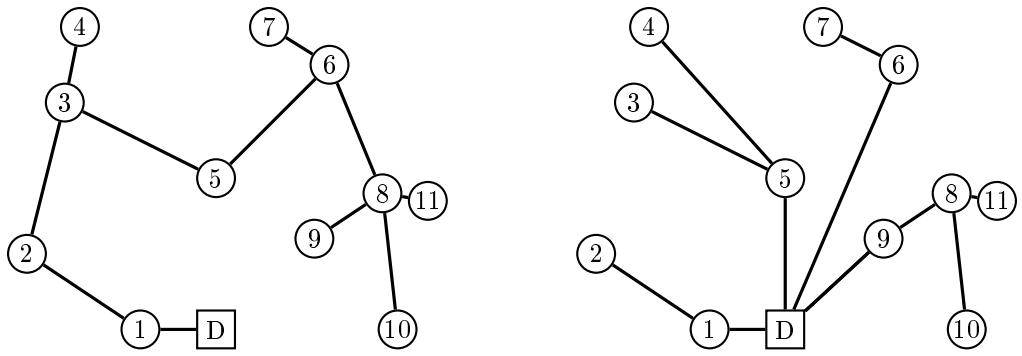
The goal of the second method is to generate clusters of customers that represent the final routes. This means that we do not want to estimate the cost for each customer separately, but estimate if for an entire set of orders at once. This way we can also take into account that some customers might match really good, but not all of them fit together at once in a vehicle.

4.3.1 Capacitated Minimum Spanning Tree

Our observation is that one of the best heuristics for the VRP from the point of worst case behavior, is based on the use of spanning trees [25]. A Minimum Spanning Tree (MST) is a tree that connects all vertices together using edges of a minimal total cost. An MST provides a lower bound for the VRP and it can be computed in polynomial time. We mark the depot as the root of the tree and call each subtree emanating from this root node a branch B_k. We want every branch in the spanning tree to correspond to a route in the solution of the routing phase. To accomplish this we also have to consider the capacity constraint. One way to do this is to consider a Capacitated Minimum Spanning Tree (CMST). In this tree the sum of the node weights of each branch should be smaller than some predefined capacity. In this case this means that we incorporate the delivery quantities by assigning each customer node its corresponding volume as weight. The maximum capacity for a branch is than the vehicle capacity. The depot has weight zero and is included in every branch. We say that the tree satisfies the capacity constraint if every branch B_k satisfies the capacity constraint.

$$\sum_{i \in B_k} q_i \leq Q. \tag{4.9}$$

An example of a CMST compared to an MST can be seen in Figure 4.1, which shows that the CMST will probably give a better representation of the actual VRP routes than the MST. Adding the capacity constraint makes the problem NP-hard [4], but heuristics like Esau-Williams [26] produce solutions close to optimal in polynomial time.



(a) Minimum Spanning Tree (b) Capacitated Minimum Spanning Tree.

FIGURE 4.1: Comparison between an MST in Figure 4.1a and a CMST in Figure 4.1a. Node D is the depot, the CMST has four branches.

Using a CMST to estimate the delivery costs results in a lower bound. If we look at a separate branch with n customers, we see that such a branch consists of n edges, while a route would exist of $n+1$ edges,

since it starts and ends at the depot. A fast and easy way to improve this lower bound is to change the cost matrix. We suggest to double the costs of all edges between the depot and a customer. In the same way we can also incorporate the fixed cost of a route in the cost matrix in the parameters between the depot and all customers. Combining us gives us the new cost values:

$$\tilde{c}_{0i} = f + 2c_{0i}.$$

To summarize we can estimate the delivery costs D of a set of orders X by constructing a Capacitated Minimum Spanning Tree T and summing the costs of all included arcs:

$$D(X) = \sum_{(i,j)\in T(X)} c_{ij} + \sum_{(0,k)\in T(X)} (f + c_{0k}).$$

4.3.2 Construction and Pruning of the Tree

In Section 3.3.3 of our literature review we discussed the Prize-Collecting Steiner Tree (PCST), where the goal is to find a connected subgraph that maximizes the profit, while including all terminal nodes and possible some Steiner nodes. In our case the terminal nodes that need to be included are the must-go customers and the may-go customers that can be included represent the Steiner nodes. The costs are the travel costs between the customers and the prizes are the delivered volumes. We want to minimize the cost per volume, while satisfying the vehicle capacity constraint, as compared to just maximizing the volume minus the costs in the PCST. Given the differences with the problem as described in the literature, we merely use the core principles discussed here, namely to distinguish two phases, a construction phase and a pruning phase. In the construction phase we add a part of the may-go customers and in the pruning phase we evaluate these added customers again to see if they are indeed a good fit. We repeat these phases until no change occurs anymore. The output is the set of nodes which are the selected orders. The pseudo code is shown in Algorithm 3.

Algorithm 3 Order Selection using a Capacitated Minimum Spanning Tree.

```
OrderSelection(customers)
    G = (V, E), V = customers, E = ∅
    while change do
        E = ∅
        G = Construction(G)                ▷ Alg. 4
        G = Pruning(G)                     ▷ Alg. 5
    end while
    return V
end
```

Construction Phase

In the construction phase a Capacitated Minimum Spanning Tree is constructed on all must-go customers and a part of the may-go customers. One of the first heuristic algorithms that was designed for the CMST problem is the Esau-Williams (EW) heuristic [26]. Since then multiple extensions and different approaches have been proposed, however the EW heuristic is still considered as one of the most efficient,

regarding computation time and optimality of results [27]. We propose an adaption of the EW heuristic for our construction phase, that incorporates the prize-collecting aspect.

Just as in the EW heuristic we use a trade-off function that expresses which new connection in the graph is most profitable. To compute the trade-off for a node, we compare the cost of linking a node to its closest neighbor, to the cost of linking it to the root. Linking a node to its neighbor represents adding this customer to the orders of today, while linking it to the root represents postponing it with the risk of delivering it by itself on a later day. Customers for which this difference in cost is high, are more profitable to connect to its neighbor. By computing these trade-offs, we build multiple subtrees simultaneously that will later become the branches of the CMST. Each subtree satisfies the capacity constraint throughout the process, which means that the final CMST also satisfies the capacity constraint. We start of each subtree as one node, which gives us $B_1, \dots B_x$, where x is equal to the number of must-go customers plus the number of may-go customers. We will refer to the subtree containing node i as $B(i)$, which means that we have $B(i) = B(j)$ if i and j are in the same subtree. We express the volume of this subtree as $q(B(i)) = \sum_{k \in B(i)} q_k$, which is the sum of the delivery volumes of all customers in this subtree. For each node we compute the trade-off $tr(i)$ that represents the savings of connecting this node to its closest neighbor that is not already in its subtree instead of connecting its subtree to the root.

$$tr(i) = \min_{\substack{j \in V \\ j \notin B(i)}} c_{ij} - \min_{k \in B(i)} c_{0k}. \tag{4.10}$$

The most negative trade-off value represents the most profitable merge of subtrees $B(i)$ and $B(j)$. However, such a merge is only allowed if it satisfies the capacity constraint

$$q(B(i)) + q(B(j)) \leq Q. \tag{4.11}$$

In each step the two subtrees with the most negative trade-off value are connected if this merge is allowed. This continues until all trade-off values that satisfy Equation 4.11 are positive. Then the subtrees with the smallest trade-off are added to the root. We continue this until all branches that contain a must-go customer are added to the root. The rest of the subtrees that have been built are then deleted from the graph. The corresponding algorithm can be found in Algorithm 4.

Algorithm 4 Construction Phase

```
CONSTRUCTION(G)
    B = { B_i = {i} | i ∈ V }
    while ∃ must-go customer ∈ V unconnected to root do
        for i ∈ V do
            j ← node ∉ B(i) closest to i that satisfies capacity constraint of Equation (4.11)
            k ← node ∈ B(i) closest to root
            tr(i) = c_ij − c_0k
            conn(i) = j
        end for
        i ← arg min_{i∈V} tr(i)
        j ← conn(i)
        Merge B(i) and B(j)
        E = E + (i, j)
    end while
    Remove all nodes and edges of branches unconnected to the root
    return G
end
```

Note that the actual implementation can be done more efficient than shown in Algorithm 4, when we use a global heap structure, a heap structure for every node and do not calculate the trade-offs in every iteration. We only recalculate trade-offs when we encounter them in line 10 and they prove to be incorrect with respect to the current situation. This can for example happen when the capacity constraint is no longer satisfied or a cheaper connection to the root is available because of a new node in the branch. This means that we only have to execute the loop of lines 4 till 9 once.

Pruning Phase

After this construction phase we evaluate each node that represents a may-go customer and determine whether it is better to keep this customer or delete it. We define the successors of a node i as all nodes s of which the shortest path from the depot to s runs through node i. When a customer has no successors or only successors that are may-go customer we can do this by simply deleting this customer and all its successors from the graph and calculating the new objective value. If the new objective value is better than the original value we keep this change, otherwise we reverse the deletion. If deletion of all its successors is not profitable or the customer has must-go customers as successors such that we cannot delete this entire part of the tree, we use an approach where we only delete this customers from the tree. In this case we evaluate the new objective when only this customer is deleted and the successors are reconnected to its predecessor by a new MST. We perform this pruning phase by starting at the leaf nodes and working to the inside. We define a set of active nodes that is initialized as all leaf nodes and to which nodes are added once all their successors have been examined for deletion. We stop the procedure once all nodes have been examined for deletion.

Algorithm 5 Pruning Phase

```
PRUNING(G)
    active ← { i | i ∈ V, degree(i) = 1 }
    pruned ← ∅
    for i ∈ active do
        p(i) ← predecessor of i
        S(i) ← i ∪ all successors of i
        if no must-go customers in S(i) then
            Calculate objective value when S(i) is deleted from V
            if objective improves then
                Delete S(i) from V and all connected edges
            end if
        end if
        if i ∈ V and i not a must-go customer then
            Calculate objective value when i is deleted and a new MST is constructed on S(i) ∪ {p(i)}
            if objective improves then
                Delete i from V and update edges of branch according to new MST
            end if
        end if
        Remove i from active
        Add i to pruned
        if S(i) ⊆ pruned then
            Add p(i) to active
        end if
    end for
    return G
end
```

5 Long-Term Solution Approach

So far we focused on minimizing the current short-term objective, which is the daily cost per volume. We referred to this approach as the short-term solution approach, since we did not explicitly considering the effect on the long-term objective. This chapter is reconsidering whether the daily cost per volume is the best choice for the short-term objective, if we in fact want to minimize the total cost per volume over the horizon. We design new short-term objectives that aim to better consider the long-term objective, for this reason we refer to this approach as the long-term solution approach. First we give an overview of the solution strategy in Section 5.1. In this section we identify two possible improvements for the short-term objectives which will be discussed in Sections 5.2 and 5.3 respectively.

5.1 Solution Overview

The long-term objective is the total cost over the horizon divided by the total volume. The current short-term objective that is used, is the daily cost per volume, however Example 2.2.2 showed that minimizing this short-term objective does not always minimize the long-term objective. Therefore we are interested in the effect of using different short-term objectives on the long-term objective. In order to design better short-term objectives, we first identify possible problems with the current one.

The first disadvantage of using the daily logistics ratio, is that the logistics ratio of the must-go customers of that day, influences the decisions with respect to the may-go customers. Suppose that delivering the must-go customers of a certain day results in a cost per volume that is much higher than the average daily cost per volume. In this case many may-go customers will be added that decrease the cost per volume of the current day, but that might increase the total cost per volume. On the other hand we can encounter days with must-go customers to which very cost efficient deliveries can be made. On this day too little may-go customers might be added. We will refer to this effect as the influence of logistics rate of the must-go customers on the decisions. A possible solution for this effect, would be to use a short-term objective that fluctuates less. We discuss some objectives regarding this aspect in Section 5.2.

Another disadvantage has already been discussed in the motivation of this research in Examples 2.3.3 and 2.3.4. Using the daily cost per volume as performance measure might result in 'easy' customers being delivered too early and 'difficult' customers being delivered too late. An 'easy' customer can be a customer that is geographically convenient to delivery to, for example because it is close to the depot or has many neighbors with which it can be combined. In this case delivery to this customer might decrease the cost per volume on many days, however postponing it might be better for the total cost per volume. A customer can be considered 'difficult' if it is located very far away from the depot and other customers

or a customer that has a very small maximum volume. In these cases delivering to the customer will almost always increase the daily cost per volume, which means the customer will probably be postponed until it is a must-go customer. This is however not always the best decision. We examine an approach that aims to optimize each delivery to a customer instead of optimizing each day in Section 5.3. We refer to this as customer optimality.

We propose a solution approach that is not necessarily optimal, but provides us with the opportunity to compare the different objectives. The approach is similar to the approach that is used in OIR currently. First, we construct initial routes of all must-go customers. Next we calculate the best insertion distance for each customer. In this calculation we only consider routes where insertion of the customer is allowed considering the delivery volumes of the customers. We sort all may-go customers according to their profitability, which we define as the improvement in the short-term objective when this customer is added. When the short-term objective is the daily cost per volume, the profitability would be the insertion cost divided by the delivery volume of the customer, but depending on the objective the sorting can be different. In this order, we add all may-go customers that improve the chosen short-term objective. The resulting customers in the routes form the selected orders for this day. In the routing phase that is executed afterwards, new routes are constructed of the chosen may-go customers together with the must-go customers. This can be seen as a reoptimization phase that is also executed in OIR. Note that this solution strategy again encounters the problems as described in Examples 2.3.1 and 2.3.2, namely that the initial solution and the order in which the may-go customers are considered can have a negative influence on the decisions. However, our goal in this approach is not to find the optimal solution strategy, but rather to compare the differences in performance of various objectives.

5.2 Influence of Logistics Ratio Must-Go Customers

We want to decrease the effect that the logistics ratio of the must-go customers has on our decisions. A first and simple adaptation is to consider all days of the past when when we decide which deliveries are profitable to add. Remember from Section 2.2.3 that our current short-term objective is defined as the daily logistics ratio:

$$STLR_d(t) = \frac{\text{cost}(t)}{\text{volume}(t)}. \tag{5.1}$$

We adapt this objective by taking the total cost per volume up to this day. We call this the cumulative short-term logistics ratio ($STLR_c$):

$$STLR_c(t) = \frac{\sum_{s=0}^{t} \text{cost}(s)}{\sum_{s=0}^{t} \text{volume}(s)}. \tag{5.2}$$

This cumulative logistics ratio has a more stable behavior than the daily logistics ratio. This means that there are less fluctuations in the short-term objective when this function is used. The decision on which customer to deliver to therefore depends less on the logistics ratio of the must-go customers. We also examine a short-term objective that neglects the logistics ratio of the must-go customers entirely. For this we introduce a fixed value for the logistics ratio and we select may-go customers for delivery if their logistics ratio on this day is smaller than the fixed value. We calculate the fixed value by taking the average of the lower bound LR_{LB} and the upper bound LR_{UB}. These bounds will be formally defined

in Section 6.1.2 in Equations 6.2 and 6.3, respectively. We calculate the delivery cost of a customer on this day ($dc_i(t)$) as the insertion costs as will be explained in the cost sharing protocol in Section 5.3.1. This gives us the following objective for the fixed short-term logistics ratio ($STLR_f$):

$$STLR_f(t) = \sum_{i \in V} \max \left\{ 0, \frac{dc_i(t)}{q_i(t)} - \frac{LR_{LB} - LR_{UB}}{2} \right\}. \tag{5.3}$$

Another short-term objective that we want to examine is inspired by the following practical consideration. Currently the actual short-term objective that is used in ORTEC Inventory Routing is different from the objective in Equation 5.1. Computational reasons provide difficulties in evaluating the change in the daily cost per volume after a may-go customer is added to a route. For this reason OIR only calculates the change in the cost per volume of the route the may-go customer is added to. We define this as the short-term logistics ratio with respect to the routes ($STLR_r$), which looks as follows

$$STLR_r(r) = \frac{\text{cost}(r)}{\text{volume}(r)}. \tag{5.4}$$

Note that the addition of a may-go customer can decrease the cost per volume of a route, but at the same time increase the cost per volume of the day. This happens when the cost efficiency of this route is bad compared to the other routes of this day. We can also encounter the opposite situation, when a route is very cost efficient compared to the other routes of this day. In this scenario, where we minimize the cost per volume of each route instead of each day, we have high fluctuations in the short-term objective. This means that the logistics ratios of the must-go customers influence the decisions even more than when we use the daily cost per volume.

5.3 Customer Optimality

We mentioned the concept of 'easy' and 'difficult' customers, however all customers are treated equally in all solution methods so far. To improve the long-term cost of constructing the routes, we do not want to pursue optimality of a certain day or route, but are more interested in optimality of a certain customer. To achieve this, we optimize the cost per volume of each delivery to a customer separately. When doing this we take the specific characteristics of this customer into account, for example the geographic location with respect to the depot. We define a Desired Logistics Ratio (DLR) for each customer i, which expresses the cost per volume that is assumed acceptable for delivery of this customer. This means that if we can attain a value for the cost (dc_i) per volume (q_i) of this customer that is lower than its DLR, we want to deliver to this customer. If the value on the upcoming day is higher we postpone delivering to the customer since we assume that the delivery will be more profitable in the future. We refer to the corresponding objective as the Customer Logistics Ratio (CLR):

$$CLR(t) = \sum_{i \in V} \left(\frac{dc_i(t)}{q_i(t)} - DLR(i) \right). \tag{5.5}$$

Because we sum the logistics ratios of all customers, each may-go customer will only be added when it directly decreases the total logistics ratio. To use this approach we need to calculate two values: the cost of delivering to the customer on the upcoming day, $dc_i(t)$ and the desired logistics ratio of this customer, $DLR(i)$. The cost of delivering to a customer can be calculated by dividing the costs of the day over all

customers of this day. For this we need a cost sharing protocol, which we discuss in Section 5.3.1. The desired logistics ratio for delivery to a customer should be a realistic value of the cost per volume that is close to optimal. We discuss how to obtain this value in Section 5.3.2.

By using the CLR, we solve the problems as mentioned in Examples 2.3.3 and 2.3.4. A difficult customer will have a high DLR and will therefore be selected earlier than it would have been in the current approach. An easy customer will have a low DLR and the bound for addition is therefore stricter, resulting in postponing this customer longer. Note that we do not solve the problems as mentioned in Examples 2.3.1 and 2.3.2, since we still construct an initial solution based on the must-go customers.

5.3.1 Cost Sharing Protocol

We determine the cost of delivering to a certain customer by sharing the total route cost between all customers in the routes. There are many cost sharing protocols known, that differ in which fairness criteria are ranked most important. Sun et al. [28] present five fairness criteria that should be satisfied when allocating the transportation cost of a given route.

1. The total route cost should be completely allocated.
2. Every customer's cost should be non-negative and no more than its single stop delivery cost.
3. The allocation of the cost should not be influenced by the direction of the route.
4. The cost allocation should be monotonic in the customers' contribution to the route cost.
5. The incentive to leave the route coalition should be as small as possible for every customer.

They analyze several methods based on these criteria. One example is to allocate the costs proportionally to the travel cost to the depot, which is easy to calculate but ignores the interaction information among customers on a route. Another example is to calculate the Shapley Value, a solution concept in cooperative games, which is computationally complex but takes the marginal contribution of each player into account.

Sun et al. state that criteria 1, 2 and 3 should always be satisfied. However, the extent to which criteria 4 and 5 are satisfied can depend on the problem and the solution method. We note that in our case, where we want to divide the route costs over the must-go and may-go customers, the interpretation of criterion 4 is different than in most cost sharing problems. The routes are constructed in order to satisfy the deliveries to the must-go customers, whereas may-go customers are only included to make the routes more cost efficient. This means that must-go customers contribute more to the route cost compared to may-go customers.

For our cost sharing protocol we allocate the route costs for must-go and may-go customers differently, where most of the cost gets assigned to the must-go customers. This is different from most approaches in other cost sharing research, since these do not make a distinction between must-go and may-go customers. Therefore we do not want to use any of the methods as described by Sun et al. in their research. To the best of our knowledge, Singh et al. [24] is the only research that considers a similar problem of cost sharing in inventory routing problems. They make the distinction in cost functions for must-go and may-go customers and we use a similar approach. Suppose we have a route r that contains must-go and

may-go customers. We assign the may-go customers the marginal decrease in route cost, corresponding to removing this customer from the route. This is the same as the insertion cost of the customer if it was not yet included. Since we require the route cost to be allocated completely, we calculate the remaining route cost if none of the may-go customers are included. We divide these costs over the must-go customers proportionally to the travel cost from the depot. This provides us with formulas for the delivery costs dc of the must-go customers and the may-go customers in route r.

$$dc_i = \begin{cases} c_{ji} + c_{ik} - c_{jk} & i \in \text{may}(r) \text{ and } (j,i),(i,k) \in r \\ \left(f + \sum_{(j,k)\in r} c_{jk} - \sum_{j \in \text{may}(r)} dc_j \right) \frac{c_{0i}}{\sum_{j\in\text{must}(r)} c_{0j}} & i \in \text{must}(r). \end{cases} \tag{5.6}$$

This formulation satisfies the necessary fairness criteria 1, 2 and 3. It also satisfies our wish of allocating the biggest portion of the route costs to the must-go customers, since the may-go customers are only assigned their marginal increase.

5.3.2 Desired Logistics Ratio

The logistics ratio of the delivery to a customer expresses the cost efficiency of the delivery. For customers that are located far from the depot and do not have neighbors nearby, this cost per volume of a delivery is almost always high. Customers to which a full truckload can be delivered or that have many close neighbors are likely to have a low cost per volume for a delivery. When determining the desired logistics ration ($DLR(i)$) for customer i we take certain characteristics of the customer into account. Possible aspects that influence the cost efficiency of a delivery are:

- Geographical location with respect to the depot.
- Geographical location with respect to its neighbors.
- Maximum volume that can be delivered.
- Visiting frequency based on ratio between maximum volume and usage rate of the customer.

We can also consider a desired logistics ratio that not only depends on customer characteristics, but also on time: $DLR(i,t)$. The threshold for delivery might be more strict when we know that we still have two weeks left to find a better value for delivery than when we know we only have two days left for this. Examples of extra aspects that influence this cost efficiency are:

- Number of days left for delivery to the customer.
- Historical information on the delivery costs.

It is difficult to combine all these aspects into one DLR, since we do not know the extent to which a certain factor influences the value. Therefore we propose multiple possible formulas and compare the performance in the experiments in Chapter 6.

The first DLR that we describe only considers the differences in the geographical locations of the customers. If the maximum volumes of all customers are higher than the vehicle capacity, the optimal

distribution strategy is to always deliver a full truckload to each customer. This gives us a lower bound on the logistics ratio of delivering to a customer when it is a must-go. We want to expedite a delivery to a customer if we can achieve a logistics ratio that is better than this lower bound. This gives us the desired logistics ratio for each customer:

$$DLR_Q(i) = \frac{f + 2c_{0i}}{Q}. \tag{5.7}$$

A disadvantage of this DLR is that it only differs between customers in the travel cost to the depot and not any of the other listed aspects.

The second DLR we discuss also takes the maximum volume of a delivery into account. This DLR considers the cost per volume corresponding to a direct delivery of maximum delivery quantity. This logistics ratio is achieved when we postpone all deliveries until the latest moment and then use only direct deliveries to satisfy these.

$$DLR_q(i) = \frac{f + 2c_{0i}}{\min(q_i^{\max}, Q)}. \tag{5.8}$$

However this bound is quite bad when the maximum volume is very small compared to the vehicle capacity. In this case a delivery to this customer can almost always be combined with many different customers, improving the cost efficiency of the delivery. The insertion cost will mostly be small when compared to the cost of a direct delivery, which means that customers might be delivered too early.

A third option that we consider, is to base our desired logistics ratio on the historical information of the deliveries. In this scenario we apply the cost sharing protocol as explained in Section 5.3.1 after each day of the horizon and we calculate the average logistics ratio of best deliveries to the customer. Note that we should be careful with this strategy, since there is the risk of reinforcing bad decisions. For this reason we require a certain number of deliveries to a customer before we can use this historical bound. Assume that we have set H that contains the logistics ratios of all deliveries to customer i up to day t:

$$H_i(t) = \left\{ \frac{dc_i(s)}{q_i(s)} \;\middle|\; s \in \{0, t\},\, x_i(s) = 1 \right\}.$$

Now assume that $\tilde{H}_i(t)$ contains the three smallest values from our historical data $H_i(t)$, and that we use a threshold of five deliveries to a customer before we use the historical data. This gives us the following formula for a time-dependent DLR:

$$DLR_3(i,t) = \begin{cases} \dfrac{\sum \tilde{H}_i(t)}{|\tilde{H}_i(t)|} & \text{if } |H_i(t)| \geq 5 \\ DLR_Q(i) & \text{else.} \end{cases} \tag{5.9}$$

The different approaches suggested here will be computationally evaluated with the Experiments as suggested in Chapter 6, after which the results are analyzed in Chapter 7.

6 Experimental Design

This chapter describes how we tested the algorithmic approaches to select the orders. Section 6.1 introduces the experimental environment that is used to test our solution methods and Section 6.3 presents the experiments that have been carried out and the data used in these experiments.

6.1 Experimental Environment

To execute our experiments we have designed an experimental environment in Python that imitates the behavior of ORTEC Inventory Routing (OIR) and incorporates our new suggested solution methods to select which customers to deliver to. We first describe how we implemented the different phases of OIR and how we incorporated the new order selection into this in Section 6.1.1. Next we explain how we implemented a benchmark solution that simulates the behavior of OIR in Section 6.1.3 and we finish by explaining how we validated the environment in Section 6.1.2.

6.1.1 Phases of OIR

As described in Section 2.2.2, the product OIR consists of four phases: forecasting, order generation, routing and execution, that are executed every day. This section explains how we implemented these phases in our own experimental environment. Besides these four phases we added a fifth phase, the order selection, which is executed after the order generation and before the routing. We discuss how these phases are implemented in the experimental environment.

Forecasting Phase

In this research we omitted the stochastic component of the problem and assumed all usage rates of the customers to be deterministic, which means that we do not have to do any forecasting in our experimental environment. This phase simply extracts the daily usage rates from the input.

Order Generation Phase

This phase first determines the day that each customer reaches its safety stock, based on the current inventory level and the daily usage rate. Customers that reach their safety stock level today are marked as must-go customers. Customers that do not reach this level, but are below their earliest delivery level

at the end of the day are marked as may-go customers. Next we calculate the maximum delivery volumes for the customers based on the vehicle capacity and their inventory level at the beginning of the day compared to their capacity.

Order Selection Phase

The order selection phase decides which customers are delivered to today. We execute one of the short-term or long-term solution approaches as proposed in Chapters 4 and 5. The input are all must-go and may-go customers and their delivery volumes and the output is a selection of the most profitable orders, containing all must-go customers and possibly some may-go customers.

Routing Phase

To simulate the routing phase we want to use a similar method as used in OIR. Therefore we decided to use ORTEC Cloud Services for Vehicle Routing. This product can be used as a standalone solution to construct routes using similar algorithms as used by OIR. The difference is that the optimization criteria is costs instead of cost per volume and no direct distinction can be made between must-go customers and may-go customers, which means that all customers that are given as input are scheduled. We can take this distinction into account partially, by giving all must-go customers high priority and all may-go customers low priority. ORTEC Cloud Services uses these priorities by first planning all must-go customers in routes. Afterwards each may-go customer is added to the schedule, by either inserting it into a route with must-go customers or by constructing a new route for this customer. After the routes are constructed we delete all routes that do not contain any must-go customers, since these routes would not be constructed in OIR. Note that including may-go routes might be optimal in some cases, however we neglect this aspect in this research.

Execution Phase

In the execution phase we update all inventories based on usage and deliveries. Since we do not take stochasticity in usage rates and travel times into account, the execution phase updates the inventories exactly according to the routes constructed in the routing and the delivery volumes determined in the order generation.

6.1.2 Validation

To assess whether the results of our approaches are realistic and more or less in scale, we compare the cost per volume that is attained to a simple lower and upper bound on the cost per volume. To calculate the lower bound we assume that all deliveries have been executed as favorable as possible. Assume that for each customer, the maximum volume that can be delivered is larger than the vehicle capacity. In this case the best solution would be to postpone all deliveries until we can deliver a full truckload. If the maximum volumes that can be delivered are smaller we can not achieve cost per volume of this strategy, but it provides us with a lower bound. The volume that is delivered in this strategy is equal to the usage

over the planning horizon of T days:

$$\sum_{i \in V} u_i T. \tag{6.1}$$

The corresponding costs to this strategy can be determined by calculating the number of full truckloads that is necessary to satisfy each customers demand and multiplying it with the direct delivery cost to the customer.

$$\sum_{i \in V} (f + 2c_{0i}) \frac{u_i T}{Q}.$$

These two expressions provide us with an expression of a lower bound on the logistics ratio. How close the actual cost per volume is to the lower bound, depends on the difference between the maximum volume of each customers compared to the size of the vehicles [21].

In a similar way we can determine an upper bound on any reasonable delivery schedule. To determine this bound, we again assume that we postpone every customer until delivery is necessary. The strategy we consider in this case is to carry out a direct delivery to each customer, however we deliver the maximum volume instead of a full truckload. The logistics ratio corresponding to this strategy can always be achieved and, given the fact that we always deliver the maximum volume to a customer in our solution approaches, should therefore be an upper bound to our logistics ratio. The delivered volume corresponding to this strategy is again as in Equation 6.1, but the travel costs in this case are:

$$\sum_{i \in V} (f + 2c_{0i}) \frac{u_i T}{q_{i,\max}}.$$

The maximum volume $q_{i,\max}$ is the minimum of the vehicle capacity and the capacity of the customer minus the safety stock level that should be in inventory at the moment of delivery.

Combining the previous equations gives is the following lower bound and upper bound.

$$LR_{LB} = \frac{\sum_{i \in V} (f + 2c_{0i}) \frac{u_i T}{Q}}{\sum_{i \in V} u_i T}, \tag{6.2}$$

$$LR_{UB} = \frac{\sum_{i \in V} (f + 2c_{0i}) \frac{u_i T}{q_{i,\max}}}{\sum_{i \in V} u_i T}. \tag{6.3}$$

If the maximum volumes of all customers are equal to the vehicle sizes, this upper bound will be equal to the lower bound and the optimal strategy is straightforward. If the gap between the maximum volumes and the vehicles sizes is big, the difference between LR_{LB} and LR_{UB} will be large and the logistics ratio of our solution methods should be somewhere in between. These bounds are used only to validate feasibility and put the experiments to scale: all proposed algorithms must of course perform within these boundaries. The results of the validation can be found in Appendix B

6.1.3 Benchmark Solution

Cost per volume by itself, in absolute terms, does not reveal much about the quality of a solution method, because it is impacted by many factors, such as the geography of the customer locations and the customer

usage patterns. In order to compare the performance of the new algorithms as suggested in this thesis we suggest a simple benchmark solution approach.

In our benchmark solution we construct an initial solution of all must-go customers first, as it is also done in OIR. Afterwards we add all may-go customers and let ORTEC Cloud Services decide which may-go customers fit best in the initial routes. This assessment is done based on minimizing the costs, with the constraint that all may-go customers should be included into the final schedule. This is a slightly different approach from OIR, where may-go customers are only added to the initial solution when they decrease the cost per volume. Afterwards we delete the routes that only contain may-go customers, since these would not be constructed in OIR.

The result is a selection of may-go customers that mimics the actual behavior of OIR. This solution is used as benchmark for the new algorithms that we have proposed. Note that it is not possible to design a benchmark that simulates the behavior of OIR exactly. As a matter of fact, the performance of the benchmark is probably a bit worse than that of OIR, since it adds more may-go customers than OIR would add. However, the order in which the customers are evaluated for addition is determined by ORTEC Cloud Services, which uses the same algorithms as OIR.

6.2 Data

This section discusses the data that is used in the experiments. We base our data for the experiments on characteristics of Airgas, the largest American distributor of industrial, medical and specialty gases and one of the users of ORTEC Inventory Routing.

Customers

Airgas is located in the United States, however ORTEC Cloud Services does not contain a map of the United States. Therefore, we use addresses of hospitals in the Netherlands in our experiments as locations for our customers. The set contains 94 location points and the corresponding geographical coordinates are shown in Figure 6.1.

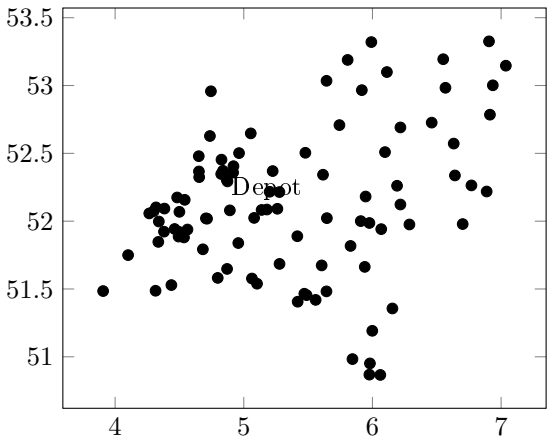


FIGURE 6.1: Coordinates of hospitals in the Netherlands

We consider two possibilities for the number of customers, where we include either all locations or a subset. The first option is to use all location points and therefore to consider 94 customers. All these instances will have the same locations, but the customer characteristics assigned to each location will differ. The second option is to consider 50 customers and in this case we will select which locations to use randomly. Every instance will have some different and some overlapping locations.

Customer Capacities

We consider five possible capacities for the customers: 500, 900, 1500, 3000 and 9000. We assign capacities to the customers based on a probability distribution of these capacities, that approximates the distribution of the capacities of the Airgas customers: $\{\frac{1}{10}, \frac{1}{4}, \frac{3}{10}, \frac{1}{4}, \frac{1}{10}\}$ The corresponding probability distribution is visualized in Figure 6.2.

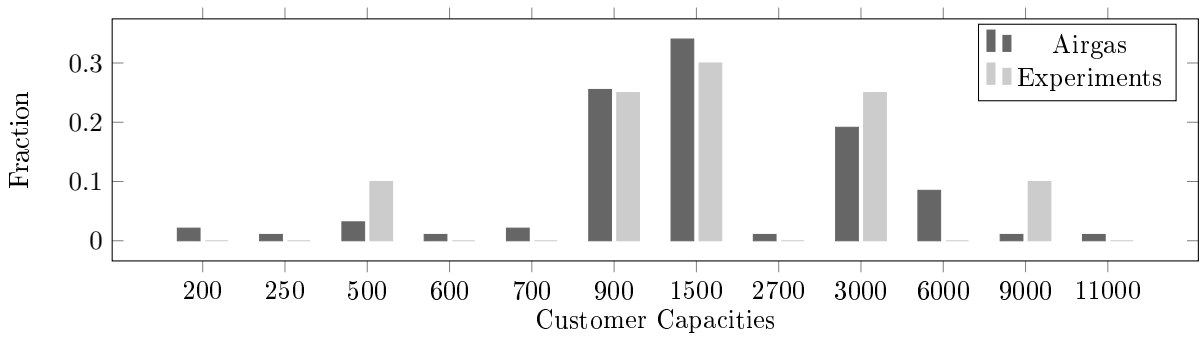
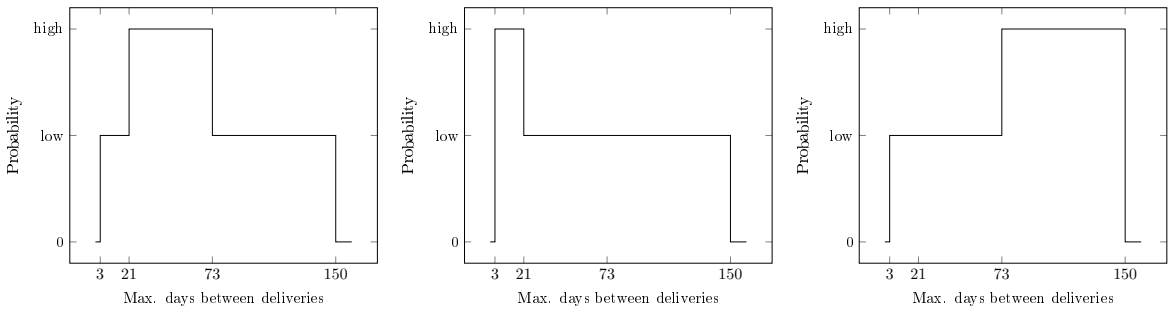


FIGURE 6.2: Distribution of the capacities of the Airgas customers and the customers in the experiments.

Customer Usage Rates

The usage rate of a customer is often related to the capacity of a customer. Customers that are known to have a high usage rate will often also have a higher storage space in order to limit the visit frequency of customers. Therefore we will calculate the maximum number of days between two deliveries to a customer based on the Airgas data and use this to calculate the usage rate. The maximum number of days between two deliveries is the ratio between the capacity and the usage rate of a customer. We divide the Airgas customers into three groups based on the number of days between deliveries to this customer being small, average or high. The boundaries between these groups are determined such, that half of the customers belongs to the 'average' group, a quarter to the 'small' group and a quarter to the 'high' group. To determine the maximum number of days between two deliveries to the customers in our experiments, we assume that the days in the groups is uniformly distributed.

We consider three options within our experiments, a dataset with medium frequency, high frequency and low frequency, indicating that the majority of customers either has a medium, high or low visit frequency. A sketch of the three corresponding distributions for the maximum number of days between two deliveries is displayed in Figure 6.3. The medium frequency set, where the majority of customers has a medium visit frequency, has the same distribution as the Airgas customers. In the high frequency option, 50% of the customers has a small number of days between deliveries, which means that the majority of the customers has a high visit frequency. In the low frequency option, 50% of the customers has a high



(a) Medium visit frequency (b) High visit frequency (c) Low visit frequency

FIGURE 6.3: Shape of the probability distribution of the maximum number of days between two deliveries, corresponding to the three frequency options of our data, where the majority of customers has either a medium, high or low frequency level.

number of days between deliveries. We calculate the usage rate of customer as the capacity divided by the maximum number of days between deliveries.

Vehicle Capacities

Airgas has a heterogeneous fleet with different capacities for the vehicles. The distribution of the capacities is shown in Figure 6.4. We consider the two most common values, a small capacity of 4800 and a big capacity of 7000 and construct instances that have either all small vehicles or all big vehicles.

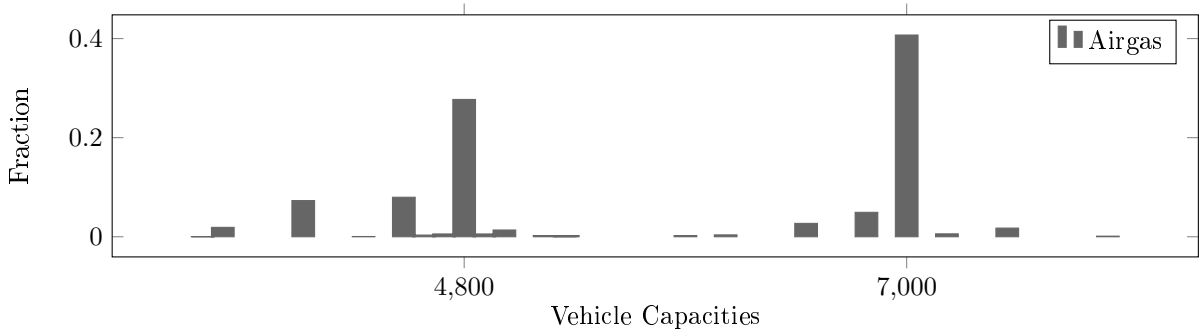


FIGURE 6.4: Distribution of vehicle sizes of the Airgas fleet.

Planning Horizon

We consider a planning horizon of 150 days. The highest value for the maximum of days between two deliveries to a customer is also 150 days, which means that every customer will be delivered at least once in our experiments.

Instances

We vary the distribution of the frequency levels and the vehicle sizes. Besides this we also vary the random seed that is used to have multiple instances with the same characteristics. We use two sizes for the customer set (all or a subset), three frequency level profiles (medium, low and high), two vehicle sizes

(small and large) and three random seeds for every options, resulting in a total of 36 instances. The characteristics of each instance can be found in Table 6.1.

Instance	Customers	Frequency	Vehicle Size
0-2	all	middle	small
3-5	all	middle	large
6-8	all	high	small
9-11	all	high	large
12-14	all	low	small
15-17	all	low	large
18-20	subset	middle	small
21-23	subset	middle	large
24-26	subset	high	small
27-29	subset	high	large
30-32	subset	low	small
33-35	subset	low	large

TABLE 6.1: Characteristics of all 36 instances used in the experiments.

6.3 Setup of Experiments

We design experiments to evaluate the performance of our solution methods and to compare different strategies as suggested in our short-term approaches in Chapter 4 and our long-term approaches in Chapter 5. This section explains two types of experiments that we will execute: the daily experiment and the horizon experiment.

Daily Experiment

The first solution methods we explained in Chapter 4 are short-term solution methods. When evaluating these approaches we are first of all interested in how much we can improve the daily cost per volume by using different order selection criteria. To examine this we propose a daily approach, where we execute our methods for a set of the same days. To obtain this set of days, we execute our benchmark solution approach for T days and save all days that were encountered. Subsequently, we solve these days with the methods we want to compare. In this experiment we can evaluate the effect of our short-term strategies on our daily cost per volume and we can examine the quality of our estimation methods. We execute this experiment for the methods as described in Sections 4.2 and 4.3.

Horizon Experiment

Besides the daily performance we are also interested if improving the daily cost per volume also improves our total cost per volume over the horizon. The second experiment examines the effect on the long-term cost, when we execute the different methods for T days starting from the same begin situation. After T days we compare the total cost per volume between different solution methods. Depending on whether the days contain easy or difficult customers, our total cost per volume can vary between the days, especially when T is not very big. We take this into account by taking the average cost per volume of the last three days in order to smooth the results.

We first execute this experiment to compare the short-term methods from Sections 4.2 and 4.3 to the benchmark method. Next we use it to compare different short-term objectives as mentioned in our long-term solution approaches. We start with examining the effect of minimizing the cost per volume of all days so far or minimizing the cost per volume of each route compared to minimizing the daily cost per volume, as suggested in Section 5.2. Besides this we consider the effect of optimizing each separate delivery to a customer as opposed to a day or a route as explained in Section 5.3.

7 Results

We present and analyze the results of the experiments as proposed in the previous chapter. Table 7.1 shows the notation of the examined solution methods. Section 7.1 starts with the results of the short-term solution approaches FSIO and CMST and compares these methods to the benchmark solution BM. Section 7.2 presents the results of the long-term solution approaches, where we make a distinction between the methods related to the influence of the state day (LR_d, LR_c, LR_r and LR_f) and the methods related to customer optimality (CLR_Q, CLR_q and CLR_H).

Notation	**Solution Method**
BM	Benchmark Solution of Section 6.1.3.
FSIO	Fixed, Stem and Inter-Order Costs of Section 4.2.
CMST	Capacitated Minimum Spanning Tree of Section 4.3.
LR_d	Daily logistics ratio of Equation 5.1.
LR_c	Cumulative logistics ratio of Equation 5.2.
LR_f	Fixed logistics ratio of Equation 5.3.
LR_r	Logistics ratio per route of Equation 5.4.
CLR_Q	DLR based on a full truckload Q of Equation 5.7.
CLR_q	DLR based on the $q_{i,\max}$ of customers $i \in V$ of Equation 5.8.
CLR_H	DLR based on the historical allocated costs of Equation 5.9.

TABLE 7.1: Notation of the solution methods as used throughout this chapter.

7.1 Short-Term Solution Approach

The short-term objective that is used in this approach is the daily logistics ratio. In FSIO fixed, stem and inter-order costs are used to estimate the costs of deliveries to customers and define recursive probabilities to select the best orders. In CMST a capacitated minimum spanning tree is constructed on a subset of the customers in order to find the best orders for the upcoming day. The goal of the experiments for these solution approaches is twofold:

1. What is the quality of both methods with respect to the short-term performance?
2. What is the effect of minimizing the short-term objective on the long-term objective?

To answer question 1 we start by examining the quality of the delivery cost estimation in Section 7.1.1 and continue by comparing the results with respect to the short-term objective to the benchmark solution

in Section 7.1.2. Question 2 is analyzed in Section 7.1.3, by comparing the results with respect to the long-term objective to the benchmark solution and to the results with respect to the short-term objective. We close with a comparison of the methods on some interesting characteristics in Section 7.1.4, in order to explain the differences.

7.1.1 Quality of Estimation Methods

Sections 4.2.1 and 4.3.1 describe two different methods to estimate the costs of delivering to customers without constructing the actual delivery routes. We analyze the quality of these methods by comparing the estimated costs to the costs of the routes as found by ORTEC Cloud Services. Remember that the requirement for our estimation methods was not to obtain an estimate as close to the actual costs as possible, but rather to have a method that provides a realistic comparison of the costs of different sets of orders. For this reason we do not calculate the absolute error, but calculate the relative deviation of the estimation with respect to the actual costs instead. This gives us the following formula for the deviation of every day t:

$$\text{dev}(t) = \frac{\text{estimate}(t) - \text{cost}(t)}{\text{cost}(t)}.$$

A negative value means that we obtained an underestimation of the actual costs, while a positive deviation indicates an overestimation. We execute the daily experiment for 100 days in order to obtain the estimation and the actual costs and calculate the deviation for every day. Figure 7.1 shows for a boxplot both methods of the deviation of all days. Each box contains the lower and the upper quartile values of the days, with a line at the median. The whiskers extend from the box to show the variability outside the quartiles and the points past the end of the whiskers show the outliers.

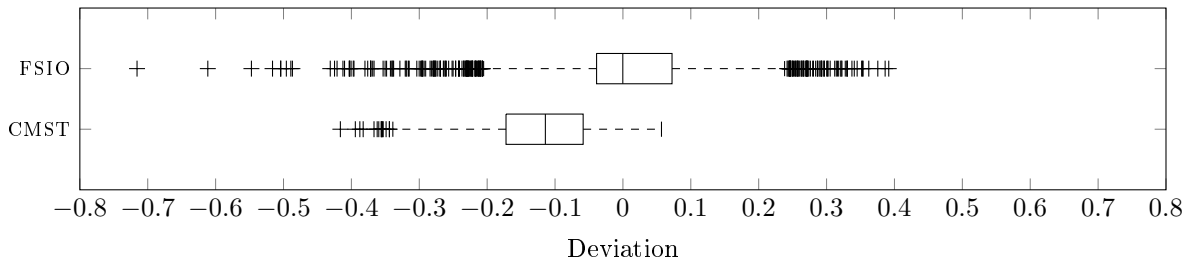


FIGURE 7.1: Boxplot of the difference between the estimation of costs and the actual costs of the routes, using either FSIO or CMST.

We see that the deviation of FSIO is centered around zero, while the deviation of CMST is centered around -0.12, indicating that it is an underestimation in most cases. The variance in the deviation is bigger in FSIO than in CMST and this method also has many more outliers. This means that although FSIO on average provides a value close to reality, it is not very suitable to compare different sets of orders, since the variance is big and it results alternately in an underestimation or an overestimation. CMST on the other hand has a smaller variance and always provides an underestimation, making the method more suitable to compare the cost of delivering to different customers, even though the estimation is not very precise. We do not observe big differences in the quality of estimation methods when different data characteristics are used.

Besides the overall quality of the estimation methods, we are also interested in how well the branches that are formed in the CMST correspond to the actual routes that are constructed. To investigate this aspect we compare the number of branches to the number of routes. We see that the routing algorithm results in one route less in 3.7% of the days. In instances with a low frequency level this even happens in 6.8% of the days. In 0.8% of the days, an extra route is constructed in the routing phase. In most days the number of branches is equal to the number of routes, indicating that the branches of a CMST are suitable to represent the routes.

7.1.2 Effect on Short-Term Objective

The daily experiment allows us to examine the performance with respect to the short-term objective, by solving the same days with different solution methods and comparing the daily logistics ratio. A visualization of this approach is shown in Figure 7.2, where 20 days of instance 0 are solved with BM, FSIO and CMST. We see that the new methods have a lower cost per volume than the benchmark for most days, which is an improvement in performance. However there are also days for which one of the new methods performs worse (days 5 and 19), or where both methods perform worse (day 13). If we compare the daily logistics ratios for the first 100 days of all 36 instances, we see that FSIO improves the LR in 67% of the days and CMST in 65% of the days. There are also days for which the new methods find a solution with a higher cost per volume, which is in 27% and 26% of the days respectively. The rest of the days the methods find the same solution as the benchmark.

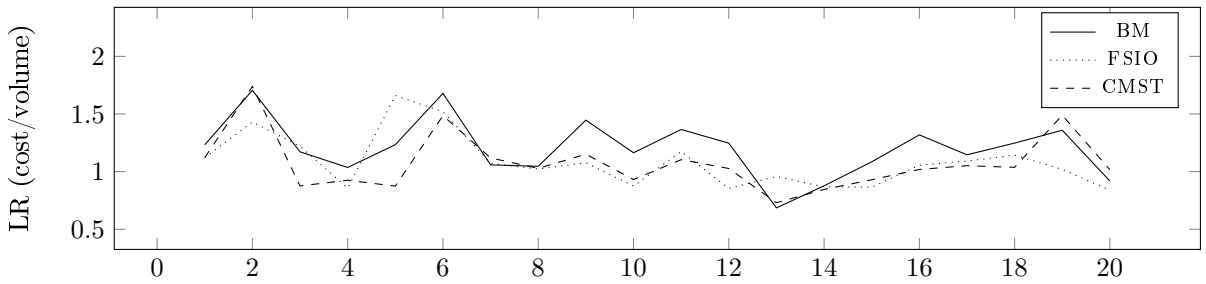


FIGURE 7.2: Daily cost per volume of the first 20 days of instance 0.

To evaluate the short-term performance we execute the daily experiment for all three methods on each instance z. For each day of the horizon t we calculate the improvement of the logistics ratio of the new methods M (where we evaluate M=FSIO and M=CMST) and we take the average over all days to calculate the average short-term improvement:

$$\text{improvement}_{ST}(z, M) = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{LR_{\text{BM}}(z,t) - LR_M(z,t)}{LR_{\text{BM}}(z,t)} \right).$$

If we aggregate over all instances we see that both methods result in an improvement. The average improvements are 7.3% and 7.1% for FSIO and CMST respectively. A boxplot of the improvement within the instances in shown in Figure 7.3. It shows that FSIO performs better in most cases, but has some instances for which the performance is very bad. CMST shows deterioration with respect to the benchmark solution for only one instance, but the median of the improvement is lower than that of FSIO. To summarize we can say that the average performance of FSIO is better, but that there are more outliers in the results.

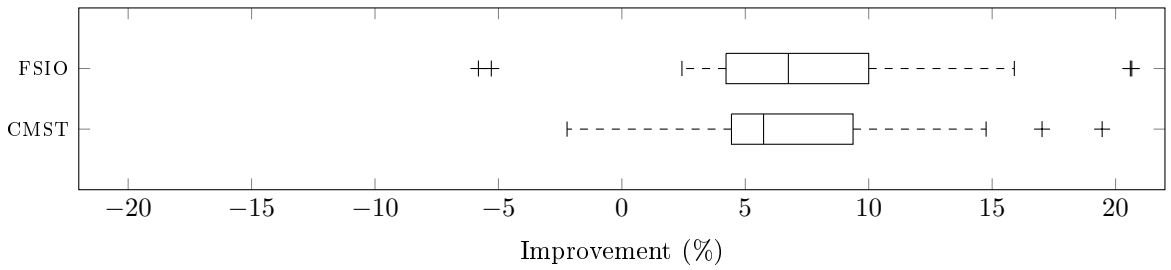


FIGURE 7.3: Boxplot of the percentual improvement for each instance of the daily logistics ratio with respect to the benchmark.

7.1.3 Effect on Long-Term Objective

The horizon experiment gives us an overview of the performance with respect to the total cost per volume. Figure 7.4 visualizes this approach for BM, FSIO and CMST, for the first 50 days of instance 0. We notice that the total cost per volume fluctuates strongly in the beginning, but gets more smooth after some days. For this instance the lower bound on the logistics ratio is 0.72 and the practical upper bound is 1.72. This means that the logistics ratios of the solution methods fall between these bounds, validating the approaches. The same holds for the logistics ratios of the other instances.

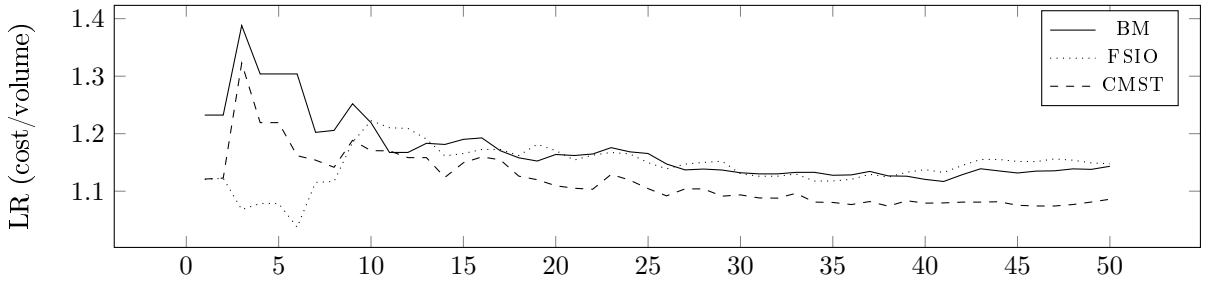


FIGURE 7.4: Total cost per volume over 50 days when solving instance 0.

To evaluate the long-term performance we first note that the difference in performance between the solution methods depends on the moment in time that is chosen for comparison, as can be seen in Figure 7.4. In order to smoothen the functions we therefore take the final total cost per volume as the average of the last three days, to compare the solution methods. We do this for $T = 50, 100$ and 150 to observe the change of the long-term performance over time. This gives us the following formula to calculate the improvement for our methods for each instance z:

$$\text{improvement}_{LT}(z, M) = \frac{\sum_{t=T-2}^{T} LR_{BM}(z,t) - \sum_{t=T-2}^{T} LR_{M}(z,t)}{\sum_{t=T-2}^{T} LR_{BM}(z,t)}. \tag{7.1}$$

We aggregate the improvement over all instances to get an overview of the average performance of both methods. The corresponding results for the improvement of FSIO and CMST with respect to the benchmark solution are shown in Table 7.2. The short-term performance of both methods is also included, to make a comparison between the effect of the methods on the short-term performance and the long-term performance.

	Short-term	**Long-term**		
	1 day	50 days	100 days	150 days
FSIO	7.3	-6.0	-8.0	-8.2
CMST	7.1	4.5	4.1	4.2

TABLE 7.2: Average improvement (%) with respect to BM, aggregated over all instances.

We see that, while the effect of both methods on the short-term performance is comparable, the effect on the long-term performance is very different. Even though FSIO improves the daily results with around 7%, the long-term results are much worse with a deterioration of 8% compared to the benchmark solution. The long-term improvement of CMST is around 4%, which is not as high as the short-term improvement of 7%, but indicates that CMST is a good solution method. We conclude that the performance with respect to the short-term objective is not a good indication of the performance with respect to the long-term objective.

7.1.4 Comparison on Characteristics

We saw that FSIO and CMST on average perform similar with respect to minimizing the daily cost per volume, but that results the for the long-term performance are very different. To examine this difference of both methods, we look at some other performance measures besides the logistics ratio. An overview is given in Table 7.3, where the best short-term solution approach is highlighted. A big difference can be seen in the number of may-go customers that is added by the different solution methods. This percentage is similar for the BM and CMST methods, but much lower in the FSIO method. The reason that this percentage is much lower in the FSIO method is that this method does not consider the clusters of customers that will be formed and for which the fixed costs hold. For this reason, half empty vehicles might actually seem cheaper in this method, while the other methods will try to fill the vehicles.

	Logistics Ratio (cost/volume)			**May-go's**	**Utilization**	**Routes**
	Average	Maximum	Minimum	(%)	(%)	(#)
BM	1.06	1.52	0.72	61	97	216
FSIO	1.14	1.62	0.76	45	80	255
CMST	1.02	1.45	0.70	62	84	245

TABLE 7.3: Overview of performance indicators of the short-term solution approaches.

Another interesting difference can be found in the vehicle utilization. The utilization for the benchmark is 97%, which can be explained by the fact that the benchmark will add all may-go customers that fit in the vehicle, whether this increases or decreases the logistics ratio. The utilization in FSIO is only 80%, which is a result of the low percentage of may-go customers that is added, as discussed before. Most interesting is the fact that, although the percentage of may-go customers in CMST is similar to that of BM, the vehicle utilization is only 84%. This also indicates that high vehicle utilization is not always desired in order to optimize the long-term performance.

So far we aggregated all data over the instances, however we are also interested in the performance of our methods for different data input. Therefore we examine some results based on the data characteristics of the instances. In this case we aggregate all instances that meet a certain characteristic, as was shown

in Table 6.1. If, for example, we examine the effect of small vehicles compared to large vehicles, both sets contain 18 instances, however if we compare the effect of middle frequency to high frequency, both sets contain 12 instances. The short-term improvements with respect to BM are shown in Figure 7.5 and the long-term improvements with respect to BM in Figure 7.6.

The short-term results show that both methods are an improvement for all data characteristics that are examined. The long-term results indicate that CMST improves the results for all data characteristics, while FSIO always results in a deterioration. The relative improvement between the different data characteristics is similar for the short-term and long-term, indicating that within the same solution method, the short-term performance is related to the long-term performance. We see that both methods perform best when the frequency level of the dataset is high and the majority of customers needs many visits in the planning horizon. In this case, there are more possibilities for combining customers, resulting in better solutions. Both methods perform worse when the frequency level is middle. An overview of the performance for all instances separately can be found in Appendix C, in order to highlight the extreme solutions.

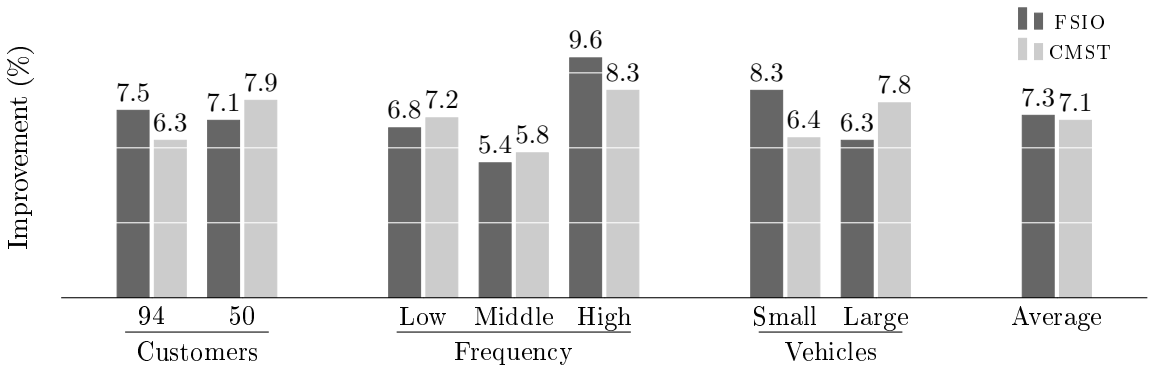


FIGURE 7.5: Average short-term improvement (1 day) of FSIO and CMST with respect to BM, for different characteristics of the instances.

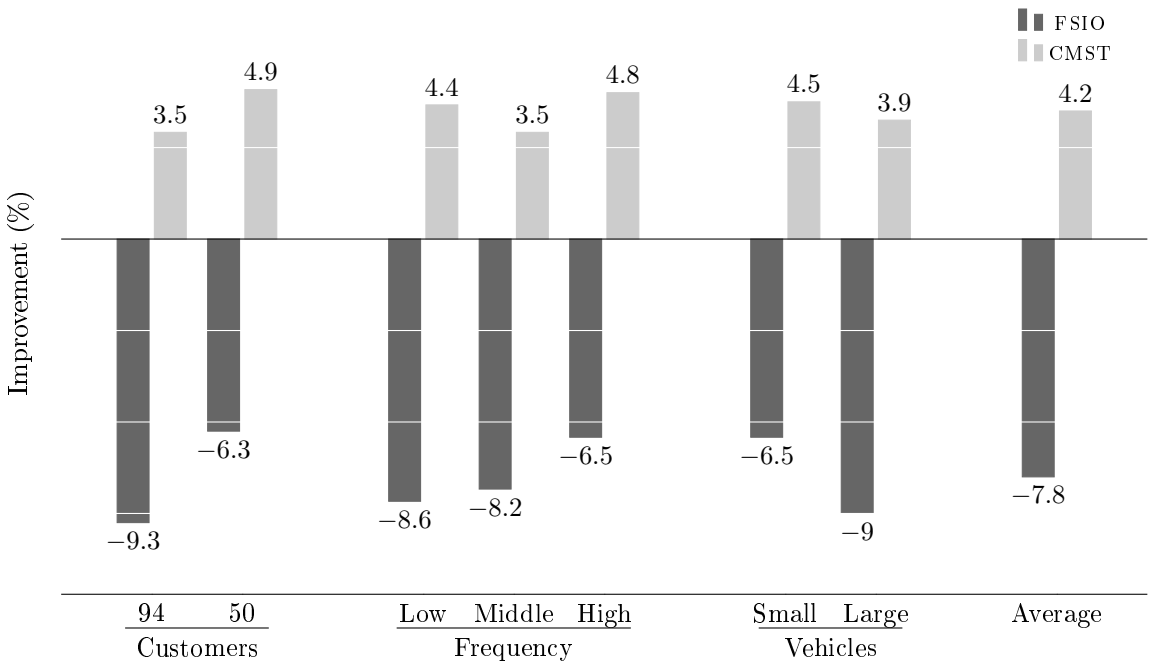


FIGURE 7.6: Average long-term improvement (150 days) of FSIO and CMST with respect to BM, for different characteristics of the instances.

7.2 Long-Term Solution Approach

So far we have seen that minimizing the current short-term objective (daily logistics ratio), is not always the best strategy to minimize the long-term objective (total logistics ratio). The goal of the long-term solution approach is therefore to examine different short-term objectives and their effect on the long-term objective. We identified that the logistics ratio of the must-go customers of a certain day influences the decisions that are made on this day, which might have a negative influence on the long-term performance. Besides this, we noticed that there is currently no distinction in selection criteria for 'easy' or 'difficult' customers. In this section we therefore focus on two questions:

1. How does the logistics ratio of the must-go customers influence the decisions regarding the may-go customers?
2. Can we improve the long-term performance by focusing on customer optimality instead of daily optimality?

Question 1 is addressed in Section 7.2.1, where we compare strategies that use the logistics ratio per day (LR_d), per route (LR_r), over all days cumulative (LR_c) or with respect to a fixed value (LR_f). We address question 2 in Section 7.2.2. We start by identifying the best desired logistics ratio by comparing methods CLR_Q, CLR_q and CLR_H, and afterwards we compare customer optimality to day optimality by using the best DLR versus the daily logistics ratio. Again we close with an overall comparison of the mentioned methods in Section 7.2.3.

7.2.1 Influence Must-Go Customers

We want to examine if decreasing the influence of the logistics ratio of the must-go customers in our short-term objective by using LR_c or LR_f improves the long-term performance. Besides this we want to examine if the short-term objective that is used by ORTEC for practical reasons (LR_r), is worse than using LR_d and if there is room for improvement on this aspect. Figure 7.7 shows an example where we executed the horizon experiment for instance 0 over 100 days, comparing the short-term objectives that minimize the cost per volume per day (LR_d), cumulative (LR_c), per route (LR_r) and with respect to a fixed value (LR_f) respectively. We see that the performance of the strategies is very similar for this instance, especially when the planning horizon is small.

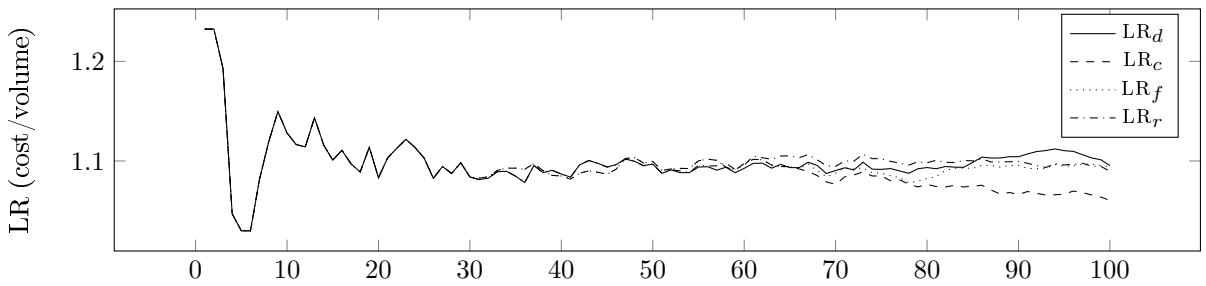


FIGURE 7.7: Total cost per volume over 100 days when solving instance 0.

First, we examine how often the use of a different short-term objective has an effect on the daily decision of which customers to deliver to. This means that we compare whether the solution methods find a

different solution when solving the same day. In order to do this we execute the daily experiment and compare the daily solutions found with LR_c, LR_r and LR_f to the solutions found with LR_d. We calculate how often using a different short-term objective, has an influence on the orders that are selected. The percentage of days for which a different solution is found, is presented in the first column of Table 7.4. These results show that the influence is smallest for LR_c and LR_f, where a different solution is found in only 6.3% and 7.3% of the days respectively. For LR_r this percentage is somewhat higher, namely 9.4%, which can be explained by the fact that the logistics ratio per route differs more from the daily logistics ratio than the cumulative and the fixed logistics ratio.

After this experiment we execute the horizon experiment with the four methods for all 36 instances for 150 days and again compare the performance with respect to LR_d. Examining the percentual improvement of the long-term objective shows that the differences are very small for all instances, in most cases less than 1%. Depending on the instance that is solved, the methods result alternately in an improvement or a deterioration. The average percentual improvement of the long-term objective is shown in the second column of Table 7.4, however since we only solved 36 instances, the differences can be the result of the statistical variation in our data. To evaluate if one method is better than another, we therefore perform a paired samples t-test. In this test we use the solutions of our methods to each instance as a paired sample and test whether the difference between the methods is significant. We require a confidence interval of 95%, which means that the difference between the methods is significant if the two tailed p-value is smaller than 0.05. The p-values corresponding to the improvement of the long-term objective are displayed in the last column of Table 7.4, where the highlighted p-values indicate a significant difference within the instances.

	Short-term	**Long-term**	
	Days with a different solution	Improvement of total logistics ratio	p-value of improvement
LR_c	6.3%	-0.3%	0.26
LR_f	7.3%	-0.4%	0.17
LR_r	9.4%	-0.7%	0.03

Table 7.4: Short-term and long-term differences of the solution methods with respect to LR_d. Highlighted values indicate a significant difference.

Performing the t-tests show that LR_r is the only solution method for which the difference in the total logistics ratio is significant from LR_d. This corresponds with the fact that this method also showed the most differences in short-term performance. The differences in short-term performance of LR_c and LR_f with respect to LR_d, do not result in a significant difference in long-term performance.

To conclude, we see that the influence of the logistics ratio of the must-go customers on our decisions is relatively small. As a result, using the cumulative logistics ratio or the fixed logistics ratio does not improve the long-term performance nor does it deteriorate it. Besides this we observe that minimizing the logistics ratio per route instead of per day has a negative effect on the long-term objective, however this effect is on average only 0.7%.

7.2.2 Customer Optimality

In this section we examine the effect of using the Customer Logistics Ratio (CLR) as described in Section 5.3, where we aim to optimize each delivery to a customer, instead of optimizing the days or the routes. We compare the effect of using three different desired logistics ratios (DLRs), based on full truckload deliveries (CLR_Q), based on maximum volume deliveries (CLR_q) and based on the logistics ratio of previous deliveries(CLR_H). Figure 7.8 shows an example where we solved instance 0 with respect to customer optimality, using three different desired logistics ratios (DLR).

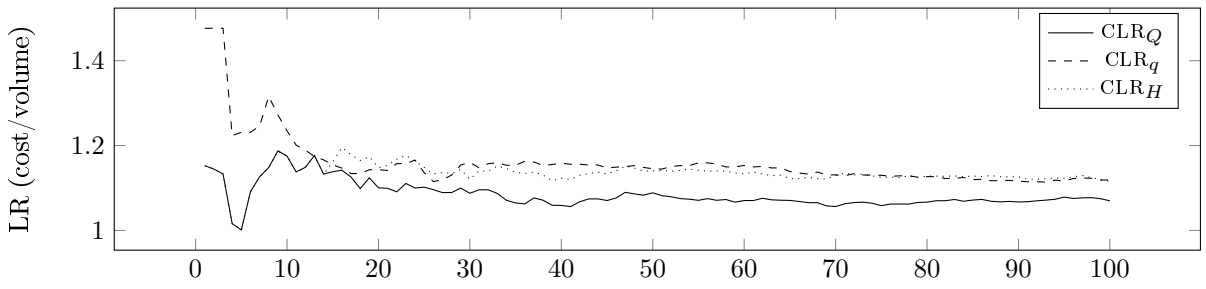


FIGURE 7.8: Total cost per volume over 100 days when solving instance 0.

First, we identify which DLR results on average in the best solutions. We consider CLR_Q as the basic version, since we only make a distinction in the geographical locations of customers when determining their desired logistics ratios. In CLR_qwe tried to improve this value by including the volumes of customers in the calculation and in CLR_H by updating it based on historical information. We use Equation 7.1 to calculate the improvement of the total logistics ratio of CLR_q and CLR_H with respect to CLR_Q. Figure 7.9 shows a boxplot of this improvement for all instances. We see that both methods perform worse than the basic DLR on almost all instances. The average deterioration of both methods is approximately 3%.

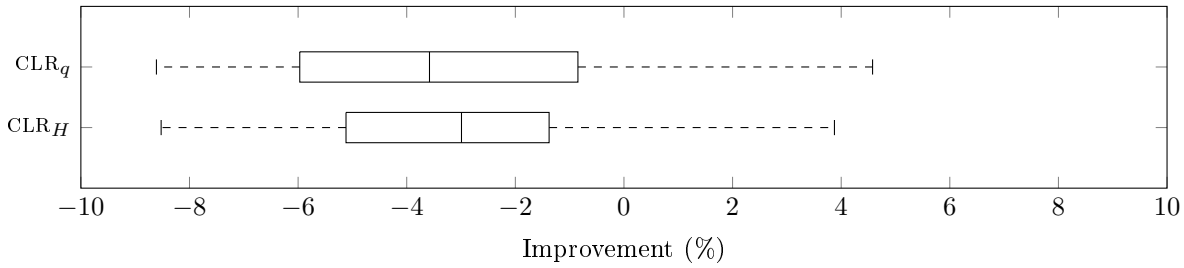


FIGURE 7.9: Boxplot of the percentual improvement for each instance of the total logistics ratio with respect to CLR_Q.

Given that CLR_Q is the best approach if we want to incorporate customer optimality, we are interested in the effect of optimizing the delivery to each customer compared to optimizing the routes of each day, which we examine by comparing CLR_Q to LR_d. Examining the short-term effect shows that this objective has a higher impact on the daily decisions than the objectives discussed in the previous section (LR_c, LR_f and LR_r). In 57% of the days we find a different solution with CLR_Q than we find with LR_d. To evaluate the effect on the long-term performance we again execute the horizon experiment and find that customer optimality improves the total logistics ratio in 75% of the instances. The average improvement aggregated over all instances is 0.9%. Performing a paired samples t-test with a confidence interval of 95% shows that this is a significant improvement over using the daily cost per volume as short-term objective. The short-term and long-term differences can be found in Table 7.5.

	Short-term	**Long-term**	
	Days with a different solution	Improvement of total logistics ratio	P-value of improvement
CLR_Q	57%	0.9%	0.03

TABLE 7.5: Short-term and long-term differences of the solution methods with respect to LR_d. Highlighted values show a significant difference.

To summarize we found that CLR_Q is the best strategy to aim for customer optimality. Using this short-term objective has a higher impact on the daily decisions than the logistics ratio of the must-go customers has on the decisions. The average improvement of using the logistics ratio per customer compared to using the daily logistics ratio is 0.9%.

7.2.3 Comparison on Characteristics

The results of this section so far showed that the logistics ratio of the must-go customers in the short-term objective has little influence on performance with respect to the long-term objective. Besides this, we saw that the customer logistics ratio that uses the full-truckload strategy to determine the DLR is best and also outperforms the use of the daily logistics ratio. In this section we compare some characteristics of the long-term solution approaches in order to explain the differences. Table 7.6 gives an overview of some characteristics for all methods, with the best long-term approach highlighted.

	Logistics Ratio (cost/volume)			**May-go's**	**Utilization**	**Routes**
	Average	Maximum	Minimum	(%)	(%)	(#)
LR_d	1.05	1.50	0.71	55	97	217
LR_c	1.05	1.57	0.71	55	97	217
LR_f	1.05	1.53	0.73	55	96	218
LR_r	1.05	1.58	0.71	55	96	218
CLR_Q	1.04	1.49	0.70	51	93	223
CLR_q	1.07	1.47	0.71	62	96	219
CLR_H	1.07	1.56	0.70	59	96	218

TABLE 7.6: Overview of performance indicators of the long-term solution approaches aggregated over all instances.

Just as in our short-term solution approaches we notice that a high utilization does not guarantee good long-term performance. The best solution has the lowest utilization of 93%. We also see that it adds the lowest percentage of may-go customers and has the highest number of routes. However, the differences between the solution methods are a lot smaller than the differences we have seen between the short-term approaches in Section 7.1.4. It is interesting to notice that the percentage of may-go customers is much higher in CLR_q than in CLR_Q, while the performance is much worse. This might indicate that the DLR of the first method might be too high, therefore allowing too many may-go customers to be added. It might be interesting to scale this measure or to adapt the cost sharing protocol in order to obtain better results for CLR_q.

Last, we again compare LR_c, LR_r LR_f and CLR_Q to LR_d as we did in Sections 7.2.1 and 7.2.2, but in this case we make a distinction in performance within the instances with respect to the different data

characteristics. Table 7.7 shows the differences in total logistics ratio when the different solution methods are compared to LR_d. The highlighted values indicate the differences that are significant, these significant differences are also visualized in Figure 7.10. This shows that there are big differences in performance within the instances. When the set of customers is big, there is no significant difference in minimizing the logistics ratio per route or per day, however using the logistics ratio for customer optimality results in an improvement of 2%. When the set of customers is small, there is no significant improvement when using CLR_Q, while the results show a deterioration of 1.4% when LR_r is used instead of LR_d. An overview of the performance for all instances separately can be found in Appendix C, in order to highlight the extreme solutions.

	Customers		**Frequency**			**Vehicles**		**Average**
	94	50	Low	Middle	High	Small	Large	
LR_c	0.2	-0.8	-0.4	-0.2	-0.2	-0.7	0.2	-0.3
LR_f	-0.1	-0.8	-0.3	-0.7	-0.3	-0.2	-0.6	-0.4
LR_r	0.1	-1.4	-1.2	-0.1	-0.7	-0.6	-0.8	-0.7
CLR_Q	2.0	-0.2	1.3	-0.1	1.4	1.2	0.6	0.9

TABLE 7.7: Percentual improvement in total logistics ratio of the solution methods when compared to LR_d. The significant improvements and deteriorations are highlighted and are also visualized in Figure 7.10.

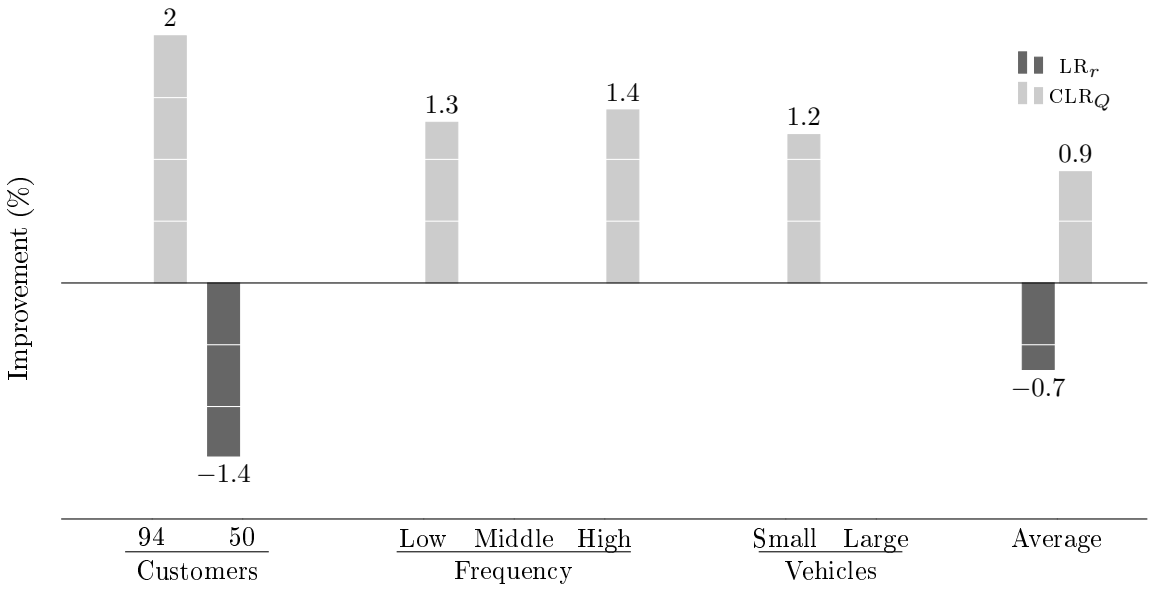


FIGURE 7.10: Average long-term improvement when compared to LR_d, for different characteristics of the instances. Only the significant differences are shown, all results can be found in Table 7.7.

8 Conclusions and Recommendations

The goal of this research was to identify opportunities to improve the long-term performance of ORTEC Inventory Routing by using different algorithmic approaches to select the orders for the upcoming day. In Section 8.1 we summarize the results of our research and discuss the conclusions. Based on these conclusions, we formulate some recommendations in Section 8.2, which consist of practical recommendations for ORTEC and recommendations for further research.

8.1 Conclusions

In this research we developed multiple algorithmic approaches to select the orders for the upcoming day in OIR. We examined the effect of these methods on both the short-term objective and the long-term objective and in this section we discuss our findings.

We developed two solution methods that both focus on minimizing the daily cost per volume. The approaches select the orders without constructing the routes, in order to avoid that the initial routes of the must-go customers and the sorting of the may-go customers have an influence on the selection. We found that both methods outperform the benchmark on the short-term objective, but that this does not give any guarantees for the long-term performance. Even though the fixed, stem and inter-order cost method obtains the lowest daily cost per volume, the total cost per volume is much worse than that of the benchmark. The method using a capacitated minimum spanning tree shows better results for the total cost per volume and outperforms the benchmark on average with 4%. Note that the benchmark solution is not an accurate depiction of OIR, meaning that we cannot guarantee the same improvement for OIR. This means that our conclusion for the short-term solution methods is twofold. First, we should be careful when designing short-term solution methods that minimize the daily cost per volume, since this does not necessarily result in optimal total cost per volume. Second, the capacitated minimum spanning tree methods shows promising results.

For this reason we also examined the effect of using different short-term objectives that focus on long-term optimality instead of short-term optimality. First, we examined the influence of the logistics ratio of the must-go customers on the decisions regarding the may-go customers. We crafted three alternative short-term objective functions that either increase or decrease this influence. This showed that the daily decisions are influenced by the logistics ratio of the must-go customers, however decreasing this influence has no significant effect on the long-term performance. Using the cumulative cost per volume instead of the daily cost per volume is therefore not an improvement. Increasing this effect by minimizing the cost per volume per route is however significantly worse. The improvement of using the LR per day instead

of per route has in the meantime been incorporated in the newest version of OIR and this research shows that this should results in better solutions.

The last method takes the difficulty level of individual customers into account when deciding whether or not to deliver to them. This way we optimize the cost per volume of each delivery instead of the cost per volume of each day. We calculated a desired logistics ratio for every customer, depending on its characteristics. The best results are found when we use a desired logistics ratio that only takes geographical aspects into account and is based on a direct delivery with a full truckload. Using this value in the customer logistics ratio results in a significant improvement of approximately 1%, compared to using the daily logistics ratio. This might seem like a small difference, however when the product is used by big companies where the distribution costs are very high, this makes a big difference. We also find that the effect very much depends on the data characteristics of the customers.

8.2 Recommendations

The best two solutions are CMST and CLR_Q, however both have their drawbacks. The CMST method minimizes the daily cost per volume, while we showed that this short-term objective is not necessarily optimal in the long-term. Moreover, this approach does not directly take long-term performance into account and neglects the difference between easy and difficult customers. Even though this method showed the best overall results in our research, it is more difficult to predict the performance when used in OIR, since the benchmark solution is not a very accurate representation. Therefore we recommend further research of this strategy before implementing it into OIR, for example by finding a better benchmark solution.

The CLR_Q approach is more focused on long-term optimality and makes a distinction between easy and difficult customers. On the other hand, it still uses the initial solution of must-go customers and a fixed order in which to consider may-go customers, two aspects of which we identified that they might have a negative influence on the order selection. We can however predict the effect of this method better, since we were able to compare similar strategies that only differ in the use of the short-term objective. Moreover, adapting the short-term objective is a less invasive change to OIR than implementing the capacitated minimum spanning tree approach. Therefore we recommend ORTEC to start by implementing a customer logistics ratio instead of a daily logistics ratio.

Besides this, we think there is still improvement possible in the desired logistics ratio that is used. We listed multiple customer characteristics that determine the difficulty level of a customer, however we only tested a few characteristics in the desired logistics ratio. We should especially consider the relation between the DLR and the cost sharing protocol. To find the best combination it is interesting to look at the percentage of may-go customers that is added by different methods, since this might indicate whether the DLR is too strict or too flexible.

Since the results of both approaches seem very promising, we also recommend further research in combining these methods. This would entail the construction and pruning of a capacitated minimum spanning tree to select the orders, while minimizing the customer logistics ratio. The fact that CMST outperforms CLR_Q might also be the result of the fixed order in which may-go customers were considered for addition by CLR_Q. This order might be different than the order currently used in OIR and we did

not examine which would be the best strategy for sorting the customers. This indicates that there are also gains in researching different sorting algorithms or using an order selection method like CMST to give priorities to certain customers.

Last, we notice that we simplified the approach of ORTEC Inventory Routing, by omitting some constraints. It is certainly necessary to perform successive experiments where we take more aspects into consideration, for example a heterogeneous fleet, multiple depots, opening hours of customers and flexible volumes that change based on the exact moment of delivery during the day. Besides this we assumed that our forecasting is hundred percent accurate, however experiments including stochasticity might provide more realistic results. More realistic results might also be obtained by using data of real users or OIR, which was not available in this research.

Appendices

A Abbreviations

Abbreviation	Description
CLR	Customer Logistics Ratio
CMST	Capacitated Minimum Spanning Tree
DLR	Desired Logistics Ratio
FPCST	Fractional Prize-Collecting Steiner Tree
FSIO	Fixed, Stem and Inter-Order Costs
ILP	Integer Linear Program
IRP	Inventory Routing Problem
LR	Logistics Ratio
LTLR	Long-Term Logistics Ratio
MST	Minimum Spanning Tree
OIR	ORTEC Inventory Routing
PCST	Prize-Collecting Steiner Tree
STLR	Short-Term Logistics Ratio
VMI	Vendor Managed Inventory
VRP	Vehicle Routing Problem
VRPO	Vehicle Routing Problem with Outsourcing
VRPPC	Vehicle Routing Problem with Private Fleet and Common Carrier

Table 1: Explanation of the abbreviations used throughout this thesis.

B Validation

This appendix contains the results of the validation as explained in Section 6.1.2. To validate whether the results of the solution methods are properly scaled, we compare the value of the total logistics ratio of the methods to the lower and upper bounds of Equations 6.2 and 6.3. Each instance has different bounds depending on its characteristics, as is visualized in Figure 1. Table 2 shows for each instance i the deviation of the lower bound LR_{LB} when compared to the value of solution method M:

$$deviation_{LB}(i, M) = \frac{LR_{LB}(i) - LR_M(i)}{LR_M(i)} \tag{1}$$

Table 3 shows for each instance the deviation of the upper bound LR_{UB} when compared to the value of the solution method:

$$deviation_{UB}(i, M) = \frac{LR_{UB}(i) - LR_M(i)}{LR_M(i)} \tag{2}$$

We see that all values in Table 2 are negative and all values in Table 3 are positive, indicating that the logistics ratios lie within the lower and upper bounds and are realistic as a solution to the instances. We also notice that the gap between the bounds is substantial and that the solution values are closer to the lower bound in most cases.

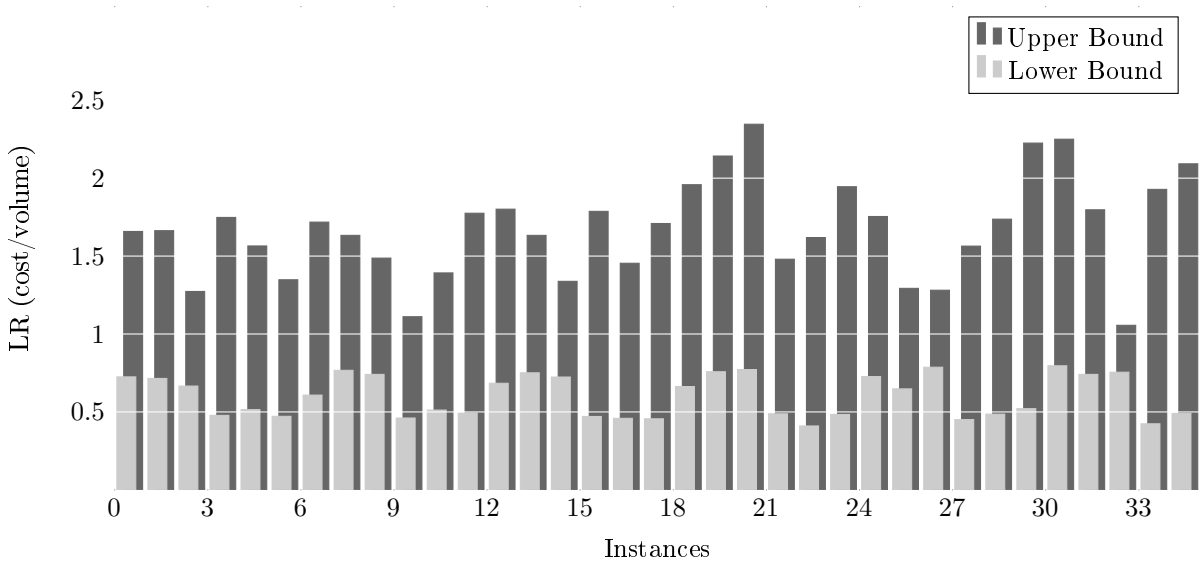


FIGURE 1: Lower and upper bounds of the logistics ratio for all instances

Instance	**Solution Method**									
	BM	FSIO	CMST	LR_d	LR_c	LR_f	LR_r	CLR_Q	CLR_q	CLR_H
0	-0.34	-0.38	-0.32	-0.33	-0.32	-0.33	-0.33	-0.32	-0.36	-0.35
1	-0.36	-0.38	-0.30	-0.32	-0.33	-0.33	-0.33	-0.32	-0.36	-0.35
2	-0.37	-0.41	-0.35	-0.36	-0.36	-0.36	-0.36	-0.35	-0.39	-0.36
3	-0.36	-0.43	-0.35	-0.38	-0.37	-0.37	-0.37	-0.36	-0.38	-0.37
4	-0.46	-0.51	-0.46	-0.46	-0.45	-0.45	-0.45	-0.43	-0.47	-0.45
5	-0.42	-0.52	-0.42	-0.43	-0.42	-0.44	-0.42	-0.42	-0.43	-0.45
6	-0.33	-0.34	-0.30	-0.32	-0.32	-0.32	-0.33	-0.31	-0.36	-0.33
7	-0.34	-0.37	-0.30	-0.34	-0.34	-0.34	-0.33	-0.33	-0.34	-0.34
8	-0.31	-0.38	-0.29	-0.32	-0.33	-0.32	-0.32	-0.31	-0.34	-0.33
9	-0.46	-0.46	-0.42	-0.43	-0.43	-0.44	-0.43	-0.41	-0.46	-0.44
10	-0.28	-0.33	-0.27	-0.28	-0.28	-0.29	-0.28	-0.27	-0.29	-0.30
11	-0.37	-0.42	-0.36	-0.38	-0.38	-0.37	-0.38	-0.37	-0.38	-0.37
12	-0.37	-0.45	-0.32	-0.38	-0.38	-0.36	-0.37	-0.36	-0.38	-0.39
13	-0.34	-0.41	-0.34	-0.35	-0.35	-0.33	-0.34	-0.34	-0.37	-0.36
14	-0.32	-0.38	-0.31	-0.32	-0.33	-0.33	-0.32	-0.31	-0.34	-0.34
15	-0.43	-0.50	-0.41	-0.41	-0.42	-0.41	-0.41	-0.41	-0.43	-0.42
16	-0.50	-0.57	-0.49	-0.51	-0.50	-0.51	-0.51	-0.49	-0.51	-0.52
17	-0.43	-0.51	-0.41	-0.43	-0.42	-0.44	-0.44	-0.42	-0.46	-0.45
18	-0.44	-0.49	-0.42	-0.43	-0.46	-0.43	-0.44	-0.44	-0.43	-0.45
19	-0.41	-0.46	-0.41	-0.41	-0.41	-0.42	-0.41	-0.42	-0.43	-0.42
20	-0.48	-0.52	-0.46	-0.47	-0.47	-0.47	-0.48	-0.47	-0.47	-0.48
21	-0.67	-0.67	-0.65	-0.65	-0.64	-0.64	-0.65	-0.65	-0.66	-0.67
22	-0.51	-0.56	-0.50	-0.50	-0.51	-0.51	-0.49	-0.54	-0.51	-0.53
23	-0.55	-0.57	-0.51	-0.52	-0.54	-0.54	-0.53	-0.54	-0.55	-0.54
24	-0.45	-0.47	-0.40	-0.43	-0.42	-0.42	-0.44	-0.42	-0.45	-0.47
25	-0.47	-0.46	-0.40	-0.46	-0.46	-0.46	-0.45	-0.43	-0.47	-0.47
26	-0.25	-0.25	-0.22	-0.22	-0.22	-0.22	-0.22	-0.22	-0.24	-0.23
27	-0.44	-0.49	-0.37	-0.42	-0.42	-0.42	-0.43	-0.39	-0.44	-0.42
28	-0.45	-0.53	-0.43	-0.45	-0.45	-0.45	-0.46	-0.47	-0.44	-0.47
29	-0.46	-0.54	-0.49	-0.50	-0.51	-0.51	-0.52	-0.50	-0.49	-0.50
30	-0.44	-0.48	-0.40	-0.44	-0.45	-0.44	-0.45	-0.43	-0.42	-0.44
31	-0.51	-0.54	-0.49	-0.51	-0.53	-0.52	-0.53	-0.50	-0.50	-0.53
32	-0.39	-0.42	-0.37	-0.38	-0.38	-0.39	-0.38	-0.37	-0.39	-0.40
33	-0.46	-0.46	-0.41	-0.43	-0.41	-0.42	-0.44	-0.42	-0.40	-0.40
34	-0.62	-0.62	-0.60	-0.60	-0.62	-0.62	-0.62	-0.60	-0.63	-0.60
35	-0.62	-0.63	-0.59	-0.61	-0.60	-0.60	-0.62	-0.61	-0.61	-0.62

TABLE 2: Deviation of the lower bound when compared to the solution value according to Equation 1.

Instance	**Solution Method**									
	BM	FSIO	CMST	LR_d	LR_c	LR_f	LR_r	CLR_Q	CLR_q	CLR_H
0	0.57	0.48	0.61	0.58	0.60	0.59	0.59	0.62	0.53	0.54
1	0.48	0.44	0.61	0.57	0.56	0.54	0.55	0.58	0.49	0.50
2	0.57	0.49	0.62	0.61	0.61	0.61	0.60	0.63	0.52	0.59
3	0.69	0.53	0.74	0.66	0.68	0.68	0.68	0.71	0.65	0.68
4	0.83	0.65	0.85	0.85	0.86	0.86	0.86	0.93	0.80	0.87
5	0.92	0.61	0.94	0.90	0.94	0.86	0.93	0.93	0.88	0.81
6	0.48	0.45	0.56	0.51	0.52	0.52	0.50	0.54	0.42	0.49
7	0.48	0.41	0.57	0.48	0.48	0.48	0.49	0.51	0.49	0.49
8	0.52	0.37	0.57	0.49	0.47	0.50	0.49	0.52	0.45	0.48
9	0.73	0.74	0.86	0.85	0.83	0.82	0.84	0.89	0.74	0.80
10	0.55	0.45	0.59	0.56	0.57	0.53	0.56	0.58	0.54	0.52
11	0.79	0.65	0.82	0.76	0.76	0.78	0.74	0.78	0.76	0.77
12	0.63	0.44	0.76	0.62	0.61	0.66	0.64	0.66	0.60	0.57
13	0.57	0.42	0.59	0.56	0.57	0.61	0.57	0.60	0.51	0.53
14	0.53	0.39	0.56	0.53	0.52	0.52	0.55	0.57	0.49	0.49
15	0.62	0.41	0.68	0.66	0.65	0.68	0.67	0.68	0.62	0.66
16	0.93	0.69	0.97	0.90	0.94	0.90	0.90	0.97	0.90	0.86
17	0.82	0.58	0.90	0.83	0.85	0.79	0.79	0.86	0.72	0.76
18	0.43	0.32	0.50	0.48	0.40	0.48	0.45	0.45	0.48	0.41
19	0.52	0.40	0.52	0.54	0.54	0.50	0.54	0.51	0.48	0.49
20	0.45	0.33	0.49	0.48	0.46	0.47	0.44	0.46	0.46	0.44
21	0.59	0.60	0.71	0.68	0.74	0.73	0.69	0.69	0.63	0.59
22	0.77	0.58	0.79	0.79	0.78	0.77	0.84	0.68	0.76	0.71
23	0.51	0.46	0.63	0.61	0.55	0.54	0.58	0.56	0.51	0.55
24	0.48	0.42	0.60	0.54	0.55	0.54	0.51	0.55	0.47	0.43
25	0.45	0.47	0.63	0.46	0.46	0.46	0.49	0.54	0.44	0.45
26	0.24	0.23	0.28	0.28	0.28	0.27	0.28	0.28	0.24	0.26
27	0.60	0.45	0.80	0.65	0.64	0.64	0.63	0.73	0.60	0.64
28	0.76	0.53	0.83	0.77	0.78	0.77	0.73	0.71	0.79	0.71
29	0.79	0.53	0.71	0.65	0.65	0.64	0.61	0.65	0.72	0.65
30	0.57	0.44	0.68	0.58	0.53	0.58	0.55	0.60	0.61	0.58
31	0.48	0.40	0.55	0.50	0.44	0.47	0.42	0.51	0.53	0.44
32	0.46	0.38	0.50	0.48	0.48	0.45	0.47	0.49	0.45	0.44
33	0.35	0.33	0.48	0.41	0.46	0.44	0.39	0.44	0.48	0.50
34	0.51	0.50	0.56	0.57	0.51	0.49	0.51	0.56	0.47	0.57
35	0.65	0.61	0.77	0.72	0.73	0.73	0.66	0.68	0.69	0.65

Table 3: Deviation of the upper bound when compared to the solution value according to Equation 2.

C Instances

This section provides an overview of the performance of the solution methods within the instances to find the extreme performances. An overview of the data characteristics of each instance can be found in Table 6.1. Table 4 shows the values of the total logistics ratio after 150 days. Just as in Equation 7.1 we calculate the average logistics ratio over the last three days of the horizon in order to smoothen the functions. The table indicates the best solution methods and the worst solution methods, by dark gray and light gray highlighting respectively. This shows that CMST most often finds the best solution, while FSIO most often finds the worst solution.

Instance	**Solution Method**									
	BM	FSIO	CMST	LR_d	LR_c	LR_f	LR_r	CLR_Q	CLR_q	CLR_H
0	1.09	1.16	1.07	1.09	1.07	1.08	1.08	1.06	1.13	1.11
1	1.12	1.15	1.03	1.06	1.06	1.07	1.07	1.05	1.12	1.10
2	1.06	1.12	1.02	1.03	1.04	1.04	1.04	1.02	1.09	1.05
3	0.75	0.83	0.73	0.77	0.76	0.76	0.76	0.74	0.77	0.76
4	0.96	1.06	0.95	0.94	0.94	0.94	0.94	0.90	0.97	0.93
5	0.81	0.97	0.81	0.82	0.81	0.84	0.81	0.81	0.83	0.86
6	0.91	0.93	0.86	0.89	0.89	0.89	0.90	0.88	0.95	0.90
7	1.16	1.22	1.10	1.16	1.16	1.16	1.15	1.14	1.16	1.15
8	1.07	1.19	1.04	1.09	1.11	1.09	1.10	1.08	1.13	1.11
9	0.86	0.85	0.80	0.80	0.81	0.82	0.81	0.79	0.85	0.82
10	0.71	0.76	0.70	0.71	0.71	0.73	0.71	0.70	0.72	0.73
11	0.78	0.84	0.77	0.79	0.79	0.78	0.80	0.78	0.79	0.79
12	1.09	1.24	1.01	1.09	1.10	1.07	1.08	1.07	1.11	1.13
13	1.14	1.27	1.13	1.15	1.15	1.12	1.14	1.13	1.20	1.18
14	1.07	1.17	1.05	1.07	1.07	1.07	1.06	1.04	1.10	1.10
15	0.82	0.95	0.79	0.80	0.81	0.80	0.80	0.80	0.83	0.80
16	0.93	1.06	0.91	0.94	0.92	0.94	0.94	0.91	0.94	0.96
17	0.80	0.92	0.77	0.79	0.79	0.81	0.81	0.78	0.85	0.83
18	1.19	1.30	1.14	1.16	1.22	1.16	1.18	1.18	1.16	1.21
19	1.29	1.40	1.29	1.27	1.27	1.31	1.27	1.30	1.33	1.31
20	1.48	1.62	1.44	1.44	1.46	1.46	1.49	1.46	1.47	1.49
21	1.48	1.47	1.37	1.40	1.35	1.36	1.39	1.39	1.44	1.47
22	0.84	0.94	0.83	0.83	0.83	0.84	0.80	0.88	0.84	0.86
23	1.07	1.11	0.99	1.01	1.04	1.05	1.03	1.04	1.07	1.04
24	1.31	1.37	1.22	1.27	1.25	1.26	1.29	1.26	1.33	1.36
25	1.21	1.20	1.08	1.20	1.20	1.20	1.17	1.14	1.22	1.21
26	1.04	1.05	1.01	1.01	1.01	1.01	1.01	1.01	1.04	1.03
27	0.80	0.88	0.71	0.78	0.78	0.78	0.78	0.74	0.80	0.78
28	0.89	1.02	0.85	0.89	0.88	0.88	0.90	0.91	0.87	0.91
29	0.97	1.13	1.01	1.05	1.05	1.06	1.08	1.05	1.01	1.05
30	1.42	1.54	1.32	1.41	1.45	1.41	1.44	1.39	1.38	1.41
31	1.52	1.60	1.45	1.50	1.57	1.53	1.58	1.49	1.47	1.56
32	1.23	1.30	1.20	1.22	1.21	1.24	1.22	1.21	1.24	1.25
33	0.78	0.79	0.71	0.75	0.72	0.73	0.76	0.73	0.71	0.70
34	1.27	1.29	1.23	1.23	1.28	1.29	1.28	1.24	1.31	1.23
35	1.27	1.30	1.18	1.22	1.21	1.21	1.26	1.24	1.23	1.27

TABLE 4: Total Logistics Ratio after 150 days for all instances, solved with all methods. Dark gray highlighting indicates that this solution method performs best for the instance and light gray highlighting indicates that this method performs worst.

Bibliography

[1] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers and Operations Research*, 37(9), September 2010. doi:10.1016/j.cor.2009.11.009.

[2] W.J. Bell, L.M. Dalberto, and M.L. Fisher. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, December 1983. doi:10.1287/inte.13.6.4.

[3] L.C. Coelho, J.F. Cordeau, and G. Laporte. Thirty years of inventory-routing. *Transportation Science*, 48(1):1–19, February 2014. doi:10.1287/trsc.2013.0472.

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

[5] L.C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. *Computers and Operations Research*, 40(2):558–565, 2013. doi:10.1016/j.cor.2012.08.012.

[6] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26 (1):103–120, 2014. doi:10.1287/ijoc.2013.0550.

[7] C. Archetti and M.G. Speranza. The inventory routing problem: The value of integration. *International Transactions in Operational Research*, 23(3):393–407, 2016. doi:10.1111/itor.12226.

[8] A.M. Campbell and M. Savelsbergh. A decomposition approach to the inventory-routing problem. *Transportation Science*, 38(4), November 2004. doi:10.1287/trsc.1030.0054.

[9] J. Kooijman. The inventory routing problem, a two-phase approach. Master's thesis, Vrije Universiteit Amsterdam, June 2012.

[10] C. Archetti, M.G. Speranza, and D. Vigo. Vehicle routing problems with profits. *Vehicle Routing: Problems, Methods, and Applications, Second Edition, MOS-SIAM Series on Optimization 18*, pages 273–298, 2014.

[11] B. Golden, A. Assad, and R. Dahl. Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems*, 7(2-3):181–190, 1984.

[12] M. Dror, M. Ball, and B. Golden. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):1–23, 1985. doi:10.1007/BF02022035.

[13] J. F. Bard, L. Huang, P. Jaillet, and M. Dror. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32(2), May 1998. doi:10.1287/trsc.32.2.189.

[14] J.F. Cordeau, D. Lagana, R. Musmanno, and F. Vocaturo. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55:153–166, March 2015. doi:10.1016/j.cor.2014.06.007.

[15] S. Huijink, G. Kant, and R. Peeters. An adaptable variable neighborhood search for the vehicle routing problem with order outsourcing. *CentER Discussion Paper*, 2014-062:1–30, 2014.

[16] S. Huijink, G. Kant, and R. Peeters. Determining which orders to outsource in the vehicle routing problem with order outsourcing. *CentER Discussion Paper*, 2015-049:1–17, October 2015.

[17] P. Goudvis. Door de bomen het boss weer zien. Master's thesis, Tilburg University, August 2001.

[18] Y. Hosokawa and E. Chiba. A heuristic algorithm for the prize collecting steiner tree problem. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on*, pages 99–102, Dec 2014. doi:10.1109/IEEM.2014.7058608.

[19] M. Akhmedov, I. Kwee, and R. Montemanni. A fast heuristic for the prize-collecting steiner tree problem. *Lecture Notes in Management Science*, 6:207–216, 2014. doi:10.1109/IEEM.2014.7058608.

[20] G. W. Klau, I. Ljubić, P. Mutzel, U. Pferschy, and R. Weiskircher. *Algorithms - ESA 2003: 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003. Proceedings*, chapter The Fractional Prize-Collecting Steiner Tree Problem on Trees, pages 691–702. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-39658-1. doi:10.1007/978-3-540-39658-1_62.

[21] J.H. Song and M. Savelsbergh. Performance measurement for inventory routing. *Transportation Science*, 41(1), February 2007. doi:10.1287/trsc.1060.0175.

[22] P. Jaillet, J.F. Bard, L. Huang, and M. Dror. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292–300, 2002. doi:10.1287/trsc.36.3.292.7829.

[23] T. Benoist, F. Gardi, A. Jeanjean, and B. Estellon. Randomized local search for real-life inventory routing. *Transportation Science*, 45(3):381–398, March 2011. doi:10.1287/trsc.1100.0360.

[24] T. Singh, J. E. Arbogast, and N. Neagu. An incremental approach using local-search heuristic for inventory routing problem in industrial gases. *Computers & Chemical Engineering*, 80:199 – 210, 2015. doi:10.1016/j.compchemeng.2015.05.023.

[25] A. Christofides. Worst case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, February 1976.

[26] L. Esau and K. Williams. On teleprocessing system design, part ii: A method for approximating the optimal network. *IBM Systems Journal*, 5(3):142–147, September 1966. doi:10.1147/sj.53.0142.

[27] H.E. Ruiz y Ruiz. *The Capacitated Minimum Spanning Tree Problem*. Phd thesis, Universitat Politècnica de Catalunya, BarcelonaTech, July 2013.

[28] L. Sun, A. Rangarajan, M.H. Karwan, and J.M. Pinto. Transportation cost allocation on a fixed route. *Computers & Industrial Engineering*, 83:61 – 73, 2015. ISSN 0360-8352. doi:10.1016/j.cie.2015.02.004.