Application of Adaptive Large Neighbourhood Search for a Rich and Real-World Vehicle Routing Problem

September 25, 2019

Author:
Sander Kroep
s.kroep@student.utwente.nl

University of Twente
PO. Box 217
7500AE Enschede

ORTEC BV
Houtsingel 5
2719EA Zoetermeer

Application of Adaptive Large Neighbourhood Search for a Rich and Real-World Vehicle Routing Problem

by
Sander Kroep
September 25, 2019

Master thesis
Submitted for the Master Industrial Engineering and Management

Supervisors:	
dr.ir. J.M.J. Schutten	University of Twente
dr. ir. E.A. Lalla-Ruiz	University of Twente
Drs. A. Rietveld	ORTEC B.V.
Drs. ir. L.M. Simons	ORTEC B.V.

University of Twente
Dep. Industrial Engineering and Business Information Systems

Preface

With the submission of this thesis, I finish my time as a student at the university of Twente. After completing the bachelor and master Industrial Engineering I have learned a lot of optimization techniques, met many new people and improved on my professional skills.

To complete my Masters in Industrial Engineering and Management, over the past half year, I have conducted a research at ORTEC in Zoetermeer. During my time at ORTEC, I was given the opportunity to enhance my understanding of optimization techniques and how they can be applied in reality. During my time at ORTEC, I have experienced lots of support. In particular I thank Arjen Rietveld and Laura Simons for their support and contribution to this research. I appreciated that you freed up time in your busy schedules to answer my questions, share your ideas and provide me with sharp feedback.

Moreover, I express my gratitude towards my university supervisors: Marco Schutten and Edurado Lalla. I appreciate that you spent your valuable time on proofreading my thesis and provided me with new insights for my research. Your insights have contributed largely to the report as it is now.

Without the continuous support of my family and friends, this work would not have been possible. For the past 5 years you have supported me during my time in Enschede, which allows me to now complete my studies.

With kind regards,
Sander Kroep
Enschede, August 2019

Management Summary

ORTEC is a supplier of optimization software. The ability to find efficient routes is in the core of their DNA. With their services they support esteemed customers to save costs and improve the delivery process. With larger and more complex business cases arriving, ORTEC always challenges themselves to find new and better ways to determine efficient routes. Previous research done by Simons (2017) shows that Adaptive Large Neighbourhood Search is a good improvement heuristic for complex and large cases. We are asked to investigate how the heuristic should be applied to a specific customer case. As a result of this, the research question that we answer is:

> *How can ALNS best be applied to a large Multi-Depot Vehicle Routing Problem with Heterogeneous Fleet and dynamic sourcing?*

The research started with performing a literature review. This literature provides understanding of the different variants of the VRP that are known to literature. This allows us to better understand the characteristics of the VRP of our client. The literature review also teaches us different strategies in which ALNS can be applied.

With the new information of the literature, we analysed the current situation. We identified how ALNS is currently applied at ORTEC and how our test cases look like. We identified three characteristics to the test cases that combined have not been tested in literature. These characteristics are:

- Heterogeneous fleet.
- Dynamic Assignment of orders to depot.
- Very large case sizes.

We found convincing evidence that ALNS finds good results for VRPs with heterogeneous fleet, for multi-depot VRPs and for large VRPs. ALNS has proven in literature that it can successfully deal with heterogeneous VRPs. We propose to test 5 different solution strategies to deal with the dynamic assignment of orders to depots. Each of the solution strategies uses a different depot assignment strategy. Basic ALNS and extended basic ALNS allow dynamic assignment of orders to depots. ALNS with fixed depot assignment and ALNS-FDATTP use strategies to fix the assignment of an order to a depot. The hybrid ALNS strategy uses the approach of Salhi et al. (2014)

The solution strategies have been implemented in the optimization software of ORTEC. We tuned the settings of each ALNS strategy individually before evaluating their performance. By testing the best performing configuration of each ALNS strategy, we develop a fair comparison. Based on the outcomes of our tests, we conclude that a combination of the Basic ALNS strategy, together with the ALNS with fixed depot assignment performs best. This results in an average

decrease in total costs of 0.22%. This corresponds with yearly savings estimated at [redacted] in plan costs.

Moreover, we provide an in depth analysis of the different ALNS methods and their performance on our case. We see that some methods which are included in the ALNS framework are not effective in improving the solution. We have analysed the influence of a different initial solution. We have shown that the final solution is better if the quality of the initial solution is also better. Lastly, we have shown that the computation time restriction for the retail client is for some cases too strict. We have shown that if the computation time is not restricted, for almost all cases an improvement is found.

Contents

List of Abbreviations

2E-VRP	2 Echelon Vehicle Routing Problem
ALNS	Adaptive Large Neighbourhood Search
ALNS-FDATTP	Adaptive Large Neighbourhood Search with Fixed Depot Assignment Through Solving the Transport Problem
CVRS	COMTEC Vehicle Routing Service
GIN	Greedy Insertion With New Route Openings
HVRP	Heterogeneous Vehicle Routing Problem
ITS	Iterated Tabu Search
KM	Kilometers
KPI	Key Performance Indicator
LNS	Large Neighbourhood Search
MDVRP	Multi-depot Vehicle Routing Problem
ORD	ORTEC Routing and Dispatch
VLNS	Very Large Neighbourhood Search
VNS	Variable Neighbourhood Search
VRP	Vehicle Routing Problem
VRPDP	Vehicle Routing Problem With Deliveries and Pickups
VRPTW	Vehicle Routing Problem With Time Windows

Chapter 1

Introduction

Nowadays road trucks are still a crucial transportation mode to move finished goods from production facilities to customers. Keese (2018) identifies that inefficiency is one of the largest challenges that the transport sector is facing. ORTEC is one of the companies that provides optimization solutions for the aforementioned problem. ORTEC, which was started by 5 econometric students in 1981, is now one of the largest supplier in optimization software and analytic solutions. Their competitive advantage lies in the combination of Operations Research, IT and enhanced understanding of the business processes. The company can be divided into two large business units: "consulting" and "products". The products division is responsible for implementation and development of ORTEC's main software solutions. For this research we focus only on the Fleet Routing and Dispatch solution. This software solution allows customers of ORTEC to plan transport routes, a trip executed by a driver, a truck and a trailer. A transport route always has a start location and an end location and it contains one or multiple customer orders. Such orders are always connected to at least one pickup and one delivery task.

Within the Fleet Routing and Dispatch software of ORTEC (ORD), users have the possibility to create routes manually or to use the automatized planning functionality. Optimizing transport routes means minimizing the total distance or minimizing the total costs of the transport routes. This process is done by the optimizer software (CVRS), which finds near optimal solutions to the vehicle routing problem (VRP). The relationship between ORD and CVRS is as follows. ORD is the front-end application that is directly visible for the user and it shows the transport routes and the customer orders. Within ORD, the user can call the functionality of CVRS. If the user selects the optimization button, the transport routes and customer orders are sent to CVRS. CVRS uses this information to develop optimized transport routes. Afterwards, CVRS sends these optimized results back to ORD, which displays them. Appendix A describes the functionality of ORD in more detail. Chapter 3 explains how CVRS optimizes transport routes.

Readers who are familiar with research on the VRP know that this problem is NP-hard. This means that the calculation time needed to find an optimal solution increases rapidly as the problem size grows. ORTEC faces variants of the VRP that are very large and the time available to do all the calculations is limited. As a result of this, it is not desirable to solve the VRPs using exact methods. Instead they use (meta)heuristics. A previous research by Simons (2017) has provided ORTEC with an insight on the latest state-of-the-art metaheuristics. As recommended by this research, ORTEC has embraced Adaptive Large neighbourhood Search (ALNS) as an improvement heuristic in large cases. Simons (2017) provides convincing evidence that ALNS is a metaheuristic that provides high quality solutions for the VRP in small computation time. We explain ALNS in more detail in Chapter 2. The application of ALNS is in practice not straightforward. As a result of this, we aim to provide an improved insight in the application of ALNS. To achieve this, we develop multiple solution strategies based on ALNS. We evaluate these strategies by assessing their performance on multiple cases from a customer of ORTEC.

This chapter serves as an introduction to the research that we conduct. Section 1.1 contains a description of the customer whose data we use to test our solution strategies. Section 1.2 briefly explains the research problem of this thesis. Section 1.3 and Section 1.4 explain the research goal and research question respectively. Section 1.5 provides the outline of this thesis.

1.1 Customer Description

For this thesis we consider the routing problem of a retail customer of ORTEC. A retail company is defined by the Cambridge dictionary as:

> *"a company that sells goods to the public in stores and on the internet, rather than to stores, other businesses, etc."*

This means that we possibly deal with a routing problem of a shop that sells electronics, clothes, food or furniture. The exact content of the customer orders is not important for this thesis. It is more interesting to see how the distribution network of the customer looks like. We first explain the contents of the distribution network and then provide a graphical representation. To avoid confusion, we refer to the customer of ORTEC as the retail client. In this research, customers are final customers who buy products from the retail client.

The distribution of the products starts in the warehouse. From the warehouse, the goods are distributed to smaller depots in the network by large road-trucks. We leave these routes outside the scope of our research because they are not determined by the software of ORTEC.

In the smaller depots, the products are loaded from the large trucks into smaller vehicles. Each of these depots have their own heterogeneous fleet of vehicles. A heterogeneous fleet of vehicles means that the vehicles have different characteristics in terms of capacity or capabilities. We must take these constraints into account when finding solutions to the routing problem. Chapter 3 further explores these constraints. The capacity of the depots is limited. It is restricted by the amount of products that can be transported by the vehicles that are assigned to that depot. The amount of products that can be handled on a depot is limited. In the assignment of customers to the depots, this capacity cannot be violated.

The last mile delivery to the final customer is done from these smaller depots. Each customer has a time-window in which their products must be delivered. This is an extra restriction in the generation of routes. An order must always be delivered in once to a customer. This means that when an order consists of multiple products, all products must be delivered by the same truck at the same moment. This thesis studies the routes that deliver the products from the smaller depots to the final customers. The assignment of customers to the depot is not fixed beforehand. Instead, customers can be delivered from each of the depots within the distribution network. This phenomenon is known as dynamic sourcing and imposes extra difficulty when developing solutions for the VRP of the retail client.

Figure 1.1 illustrates a simplified image of the distribution network of the retail client. From the central warehouse (dark blue), large trucks drive to the smaller depots (orange). From the depots,

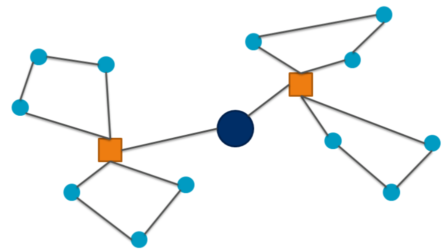


Figure 1.1: Example of Multi-Depot VRP with central warehouse(dark blue dot), depots(orange) and customers(light blue)

the last mile delivery routes are determined. The customers are represented by the light-blue dots in Figure 1.1. Confidential Appendix B shows an exact overview of the the distribution network of the retail client.

1.2 Problem Formulation

Simons (2017) provides an overview of solution algorithms for the VRP. She shows that ALNS is a promising solution algorithm for a variety of cases. ALNS has many different components that need to be configured. These configurations allow ORTEC to find good solutions to many cases, but it is challenging in the sense of finding the right configuration for a specific case.

Simons (2017) provides insight on how ALNS could be applied to large customer cases. The vehicle routing problem of the retail client contains more transport tasks than what has been tested so far. Keese (2018) mentions that a large growth is expected in the transportation sector resulting in larger and more complex VRPs. This is also a trend expected by ORTEC. As a result of this, additional insight is needed on how ALNS can be configured if the case size increases.

The case we are researching presents a number of challenges in the configuration of the ALNS framework. The first challenge is the large size of the cases. In addition to this, we find the multi-depot aspect in combination with dynamic sourcing and a heterogeneous fleet. Especially this heterogeneous fleet forms a challenge that has not been investigated yet. Furthermore, it is interesting for ORTEC to see what the influence of dynamic sourcing on the performance of ALNS is.

Finally, ORTEC is interested in knowing if ALNS can be improved further in general. The last research to ALNS that was conducted for ORTEC included literature until 2014. They believe that newer methods could be available. If we can find these methods, this could lead to an even further improved performance for existing customers of ORTEC.

1.3 Research Goal

In the previous sections we have motivated our research by explaining the practical and theoretical importance of this work. This section describes the research goals and the steps that we take to reach the goals.

For ORTEC it is important to provide high quality solutions to its customers. To this extend they always strive for improved optimization solutions. They find it crucial to have insight in how ALNS works and why it is successful or not. The first goal of this thesis is therefore to provide ORTEC with insight in the latest developments of ALNS. We aim to achieve this goal by conducting a literature study.

The second goal of this thesis is to advice ORTEC on how they can apply the ALNS framework on the very large multi-depot VRP with dynamic sourcing and a heterogeneous fleet from the retail client. To reach this goal we first conduct a literature study, which identifies what is already known on the characteristics of the VRP of the retail client. This way we aim to identify what the influence of the heterogeneous fleet, dynamic sourcing and multi-depot aspect on the applicability of ALNS is.

1.4 Research Questions

Corresponding with the previously stated research goals, we define our main research question. To answer our main research question we develop a set of sub questions. These sub questions guide us towards providing a complete answer to the research question.

The main question that we answer with this research is the following:

> *How can Adaptive Large Neighbourhood Search be best applied on a large Multi-Depot Vehicle Routing Problem with Heterogeneous Fleet and dynamic sourcing?*

We believe that this research question sufficiently covers the problem statement and research goals. The aspect of advising ORTEC on how to configure the ALNS framework is clearly present. We also have space within this research question to point out new developments of ALNS that can potentially be used by ORTEC.

As a start of our research, we must create a better understanding of the different variants of the VRP and the solution techniques that are available. Therefore we investigate what is already known in literature on the VRP variant of the retail client and how it can be solved. All in all, this leads to the first set of sub-research questions.

1. What can literature teach us on...
   - (a) The different variants of the Vehicle Routing Problem?
   - (b) The latest developments of Large neighbourhood Search techniques and their applicability for rich and real-life cases?
   - (c) Large neighbourhood Search techniques and their applicability for the customer specific VRP?

   With the findings from literature, we investigate the current practices of ORTEC. This second set of research questions increases our knowledge on how the VRP of the retail client looks like and how it is currently solved.

2. How does the current situation look like?
   - (a) How does the software of ORTEC solve VRPs in general?

(b) How does the VRP of the retail client looks like?

(c) Which solution algorithm is currently used to to solve the VRP of the retail client?

(d) How does the current solution algorithm perform?

After answering these first two sets of research questions, we have information from the literature and our case specific details. The next step would be to combine this information and develop one or multiple solution strategies. We must asses if we have developed solution strategies that are successful in providing a good solution for the VRP of the retail client. We achieve this by answering the third set of research questions.

3. What is a good solution strategy for solving the VRP of the retail client?

   (a) Which ALNS based solution strategies can we define to provide a solution to the VRP of the retail client?

   (b) How do we determine the settings of our solution strategies?

   (c) How can we compare the performance of the solution strategies?

   (d) How do we validate the results that we find?

After testing our solution design we must interpret the results. In this last research question, we also pay special attention to the usefulness of the outcomes of our research for ORTEC.

4. What insights do we gain after our tests and how can they be used by ORTEC?

   (a) What influence do we observe from the heterogeneous fleet on the ALNS performance?

   (b) What influence do we observe from the large problem size on the ALNS performance?

   (c) What influence do we observe from the dynamic sourcing on the ALNS performance?

Answering these research questions should provide us with all the information that is required for answering the main research question. This research is structured in such a way that each set of research questions is answered in a separate chapter. We provide more details on the outline of the thesis in the next section.

1.5 Outline

Since we conduct our tests with the software of ORTEC, we are naturally restricted in what we can test. Keeping that in mind we cannot simply extend the current ALNS techniques of ORTEC. The first research goal is to investigate if there are any new developments of ALNS that can be useful for ORTEC. We mainly investigate the relevance of these developments for the case of the retail client. However, they may also be applicable for general problem instances. We do not test the applicability of these ALNS developments for other cases.

The second research goal is to provide an advice how ALNS can be applied for the VRP of the retail client. To achieve this, we design multiple solution techniques that we test in our research. Moreover, we aim to discover a relationship between dynamic sourcing and the application of ALNS. Based on our findings from literature, we conduct tests to provide evidence on how well these solution techniques perform for the retail case.

Chapter 2 is an in depth analysis on what is currently known in literature. We describe different extensions of the VRP and what is known on some important solution methods. Chapter 3 provides an overview of the current situation. In this chapter we present more details on the retail client and the specifics of the case that we use for our thesis. Based on the literature study in Chapter 2, and the information on the current situation we aim to identify promising solution techniques. Chapter 4 describes which solution methods we experiment with. Chapter 5 presents our experiment design. We present the findings of these experiments in Chapter 6.

Chapter 2

Literature Review

In Chapter 1, we defined the problem statement of this thesis. In this chapter we answer the first set of research questions.

1. What can literature teach us on:
   (a) the different variants of the Vehicle Routing Problem?
   (b) Large Neighbourhood Search techniques and their applicability for the customer specific VRP.
   (c) the latest developments of Large Neighbourhood search Techniques and their applicability for rich and real-life cases.

Section 2.1, describes different variants of VRPs that are known in literature and relevant to this research. We need this information in order to investigate which solution algorithms work good for which VRP variant. Following to that, in Section 2.2 we provide an in depth analysis on the most important solution algorithms that are available in literature. This information is used to design our own solution algorithm for the retail client. The information also allows us to identify possible new algorithms that can be used by ORTEC. Section 2.3 investigates the application of Large Neighborhoud Search (LNS) techniques on different variants of the VRP.

2.1 Vehicle Routing Problem

In this section we provide an overview of the different variants of the VRP. We first discuss the basic VRP, and continue with some of its most important extensions that are relevant for our analysis. To the best of our knowledge there is no work available that describes all different variants of the VRP. For an extended overview of different variants of the VRP and their relations, we refer to Braekers et al. (2016).

2.1.1 Classical VRP

The first work done on the VRP stems from Dantzig and Ramser (1959). They present a generalization to the traveling salesman problem, which they call the Truck Dispatching Problem. They aim

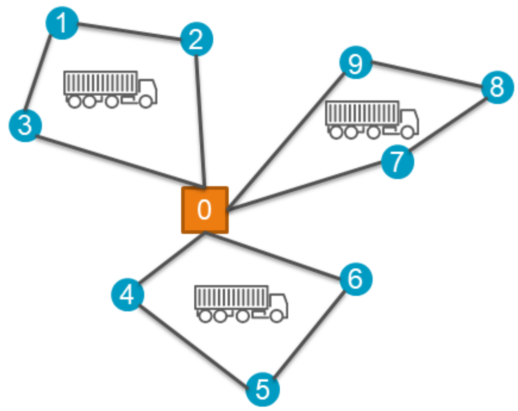


Figure 2.1: Classical VRP representation

to find a shortest-route along a set of destinations which are served by a fleet of trucks with equal capacity. Many different methods exist to represent the VRP. Since we exclude any exact solution methods, we do not present a mathematical model here. Instead, we use the graph representation as given by Cordeau et al. (2007). They define a graph $G = (V, E)$. Here V is a set of nodes in which each node i represents a customer such that $V = \{0, 1, ..., n\}$ and $i \in V$. Node 0 represents the depot in this case. All customers have a demand which is denoted by q_i. E represents a set of edges such that $E = \{(i, j), (i, j \in V)\}$. To every edge between node i and j, we assign travel costs c_{ij}. The travel costs c_{ij} are often dependent on the length of edge between node i and j (e.g. distance/time). These costs can be denoted in Figure 2.1 as a number on each line. Figure 2.1 provides an example of a solved instance of the basic VRP. This figure contains a set of 9 customer nodes. Each of the blue circles represent a customer, and the big orange square in the centre of Figure 2.1 is the depot.

2.1.2 Heterogeneous Vehicle Routing Problem

The heterogeneous vehicle routing problem (HVRP) is the first extension of the classical VRP that we discuss. In the previous section we saw the original formulation by Dantzig and Ramser (1959), who assume that all vehicles are identical. It is obvious that this assumption rarely holds in reality. In Figure 2.2, we show an example of a VRP with heterogeneous fleet. We see that there are three sub-tours which are executed by different type of trucks. Tour 1 is served by a container truck, tour 2 is served by a tanker and tour 3 is served by a cement truck. In the homogeneous vehicle routing problem, all vehicles are the same for these three routes.

This heterogeneous variant of the VRP was introduced by Golden et al. (1984). The additional

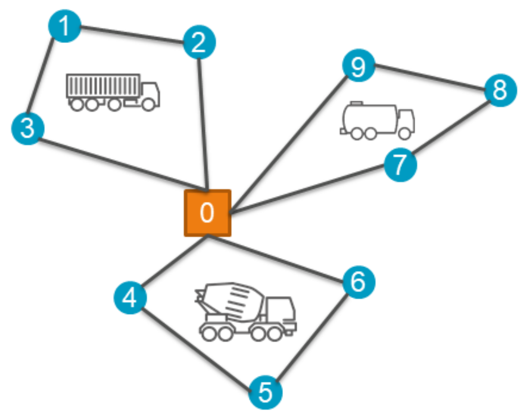


Figure 2.2: An example of a Heterogeneous Vehicle Routing Problem

assumptions allow for differentiation of capacity and cost sets over the different vehicles that are included in the fleet. Golden et al. (1984) assume that there is an unlimited number of vehicles available of each type. On the other hand, Li et al. (2018) consider a case where only a limited number of vehicles of each type can be used to solve the VRP. Li et al. (2018) take the number of vehicles of each type as a fixed input for solving the routing problem. A variant of the HVRP which does not take the fleet size as a fixed input, is known to literature as the Fleet Size and Mix Vehicle Routing Problem as described by Salhi et al. (2013); Belloso et al. (2019). Not only the capacity may be different among the vehicles. There may also be an imposed restriction on which customers can be assigned to which trucks. Penna et al. (2019), refers to this restriction as site dependency. They explain that in this situation a customer can only be visited by a subset of all the vehicles.

2.1.3 Vehicle Routing Problem with Time Windows

So far we have extended the classical vehicle routing problem by including a heterogeneous fleet of vehicles. This section discusses the vehicle routing problem with time windows (VRPTW). This means that an order must be delivered or picked-up within a certain time-interval. Figure 2.3 shows how we extend the classical VRP to include time windows. The green boxes represent in which hours of the day an order must be delivered. Since ORTEC always works with time windows, this is an important extension to understand. El-Sherbeny (2010) interprets the time window restriction as follows: A customer must be visited within the time window provided. The vehicle is allowed to arrive at the location before the start of time window but must wait service until the opening of the time window. The driver must leave the node before closing of the time window. When

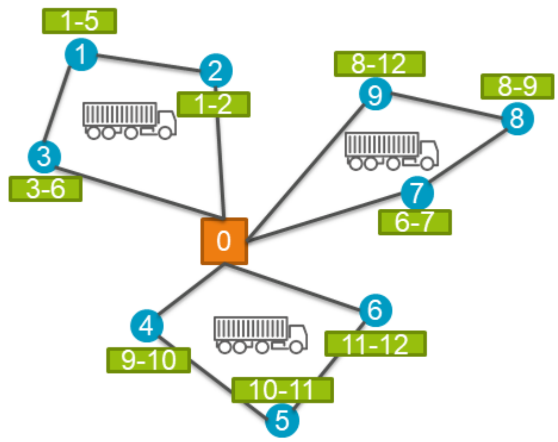


Figure 2.3: VRP with time windows

including the time window restriction in this way the service time of a customer must be stated. A result of this extension is that a solution is only feasible if every customer is visited within the designated time window. This formulation of time windows is referred to as the hard formulation. A soft formulation of the time window restriction is presented by Koskosidi et al. (1992). In this formulation it is allowed to violate the time window restriction, but this is penalized in the objective function. Imagine that the driver leaves the customer location after the time window is closed. The amount of time that the drive violates the window is multiplied with a fixed cost factor. If these costs are included in the objective function, a soft formulation of the time window restriction is realized.

2.1.4 Multi-depot Vehicle Routing Problem

In the previous sections, we have discussed VRP variants that contain only 1 depot from which all the vehicles leave. This section describes the literature concerning the multi-depot vehicle routing problem(MDVRP). To the best of our knowledge, the first variant of the MDVRP was described by Kulkarni and Bhave (1985). They assume that the demand at each depot is smaller than the total capacity of the trucks. In reality the depot may also have restricted capacity. This restriction is best described by Calvet et al. (2019). They describe a solution method to the MDVRP with limited capacity at the depots and stochastic demands. An extensive overview of all the variants of the MDVRP is given by Salhi et al. (2014) and Montoya-Torres et al. (2015). Salhi et al. (2014) provides a mathematical formulation that includes some practical considerations of the MDVRP.

- Fixed total number of vehicles.

The number of vehicles of a given type are restricted. There are also variants in which the number of vehicles to be used is unrestricted.

- Restricted number of vehicles assigned to a depot.
  There can be a restriction put on the number of vehicles that are assigned to a depot.

- Not all vehicles can be served by all depots.
  We have mentioned before that in the heterogeneous VRP there is a restriction that states that not all customers can be served by all vehicles. In the multi-depot variant we also find an extension that puts a restriction on the assignment of the vehicles to depots.

- Vehicles not required to return to start-depot.
  We find this variant in the work of Li et al. (2016). The variant where the start and finish depot of a vehicle is flexible is called the Multi-depot vehicle routing problem under shared depot resources. An advantage of this relaxation is that the vehicles can drive to the nearest depot after visiting the last customer in its route. Potentially this could reduce the total distance travelled.

- Vehicles not required to return to any depot.
  Pichka et al. (2014) describe a variant in which the vehicle does not return to a depot, but stays at the customer who was visited the last. This variant is called the multi-depot open vehicle routing problem (MDOVRP). A MIP-formulation of this variant of the VRP is given by Lalla-Ruiz et al. (2016).

2.1.5 2-Echelon Vehicle Routing Problem

The 2-echelon vehicle routing problem (2E-VRP) shows many similarities to the multi-depot variant. The first work that is known on this variant is written by Perboli et al. (2010). The model considers 2 different types of routes. First level routes which consider the delivery of products from a central depot, to multiple satellites. The second level of routes considers the distribution of products from these satellites to final customers. The simultaneous consideration of the first and second level routes makes it different from the MDVRP variant. Extensions to this model can be found in Grangier et al. (2016). In this work, a solution approach is presented for the 2E-VRP with satellite synchronization. This means that trucks for first and second level routes arrive at the same time at a satellite. This represents a quick transfer of products between the two trucks.

2.1.6 Vehicle Routing Problem with Deliveries and Pick-ups

The last extension of the VRP that we discuss is the Vehicle Routing Problem with Deliveries and Pickups(VRPDP). Products are always transported between customers and a central depot. Çatay (2010) identifies 3 different variants of the VRPDP:

1. Delivery first, pickup second
   This problem is better known in literature as the VRP with back hauls. In this variants, vehicles are only allowed to pick-up goods after they have delivered all the goods. The route often looks as following: the delivery vehicle starts full at a depot. After finishing a tour in which it delivers all its goods, it starts a second tour. In this tour, the vehicle visits a second set of customers to pickup their products.

2. Mixed pickup and delivery
   This is a more relaxed formulation of the VRPDP. The order in which the pickup and deliver tasks are executed does not matter. In this variant it can happen that the vehicle visits a customer twice in a route in order to execute the pickup and deliver actions.

3. Simultaneous pickup and delivery
   To prevent a truck from visiting a customer twice within a route, one can enforce simultaneous pickup and delivery. Here on a single customer visit, the transport vehicle delivers its products and also picks up the goods from the customer.

2.2 Solution Algorithms

The variants of the VRP that are discussed in Section 2.1 are all NP-hard problems. As a result, it is important to develop solution algorithms that find good solutions in short computation time. El-Sherbeny (2010) classifies solution algorithms for the VRP in exact methods, heuristics, meta-heuristics and artifical intelligence. For this thesis, we focus on heuristics and metaheuristics. The classification framework of solution algorithms that we use in this thesis is shown in Figure 2.4. El-Sherbeny (2010) explains heuristics and metaheuristics as follows. A heuristic is a technique which aims to find an as good as possible solution without guaranteeing optimality. Heuristics are problem specific. Metaheuristics are strategies that describe the exploration of the search space. Metaheuristic often make use of one or multiple heuristics in a intelligent manner. Toth and Vigo (2001) claim that metaheuristics are enhancements of classical heuristics.

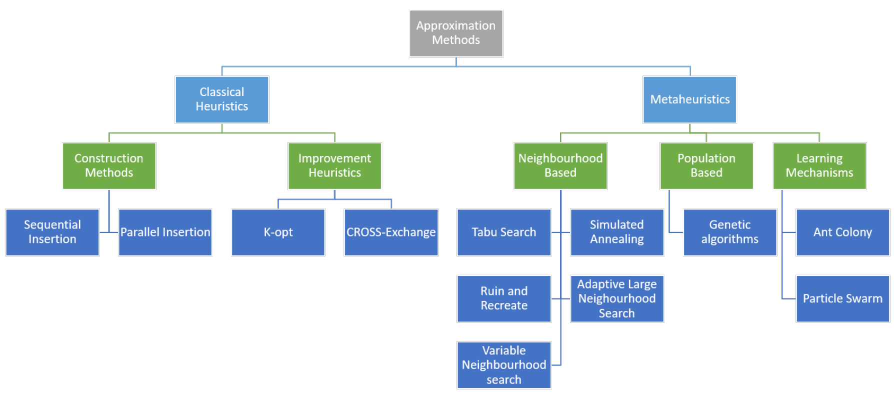


Figure 2.4: Framework for classification of approximation methods following Laporte (2009)

2.2.1 Classical Heuristics

Figure 2.4 shows that classical heuristics are divided in 2 classes. The first class contains heuristics that build new solutions from scratch. These heuristics are called construction methods. The second class of heuristics are improvement heuristics. The heuristics in this second class try to improve an existing solution. We discuss relevant methods in both classes in this section.

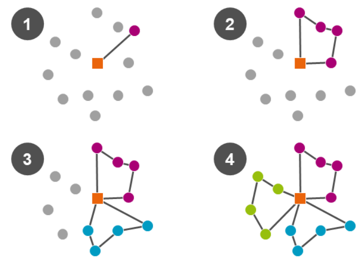


Figure 2.5: Sequential Insertion as illustrated by ORTEC (2019)

Sequential Insertion

The first construction heuristic that we discuss is sequential insertion. Joubert and Claasen (2006) explain the construction of a route using sequential insertion. Initially, using an initialization criteria, the first customer to insert in a route is selected. This customer is called the seed customer. This customer is inserted into a route. In the remainder of the process, customers are inserted into that same route until no more customers can be inserted. Next a new seed customer is selected and inserted into a new route. This route is again filled with other customers. This process is repeated until all customers are inserted into a route. Figure 2.5 illustrates the construction of routes using sequential insertion.

Parallel Insertion

Parallel Insertion is the second and last construction heuristic that we discuss. We saw that sequential insertion inserts customers into a route, one-route at a time. In parallel insertion, as described by Potvin and Rousseau (1993), routes are build in parallel. The algorithm of Potvin and Rousseau (1993) starts by selecting the customer furthest away from the depot that is unplanned. This customer is inserted into a route. Next, a second customer order is selected. This order is then inserted in the best possible location. This can be either in a new route, or in an existing route. This process is repeated until no more customers can be inserted into routes. The parallel insertion method for creating a solution to the VRP is illustrated in Figure 2.6.

K-Opt

The first improvement heuristic that we describe is k-opt. Helsgaun (2009) describes how the k-opt heuristic works. In each iteration of the heuristic, k edges of a route are replaced by k different edges. By replacing the edges the heuristic aims to find a better feasible solution. We illustrate this heuristic in Figure 2.7, where in each iteration 2 edges are replaced. This is also known as 2-opt. In the left part of the figure the original route is shown. The 2-opt operator selects first the 2 purple edges. These purple edges are removed and replaced by the green edges. From Figure 2.7 one sees that the new route is shorter than the original route.

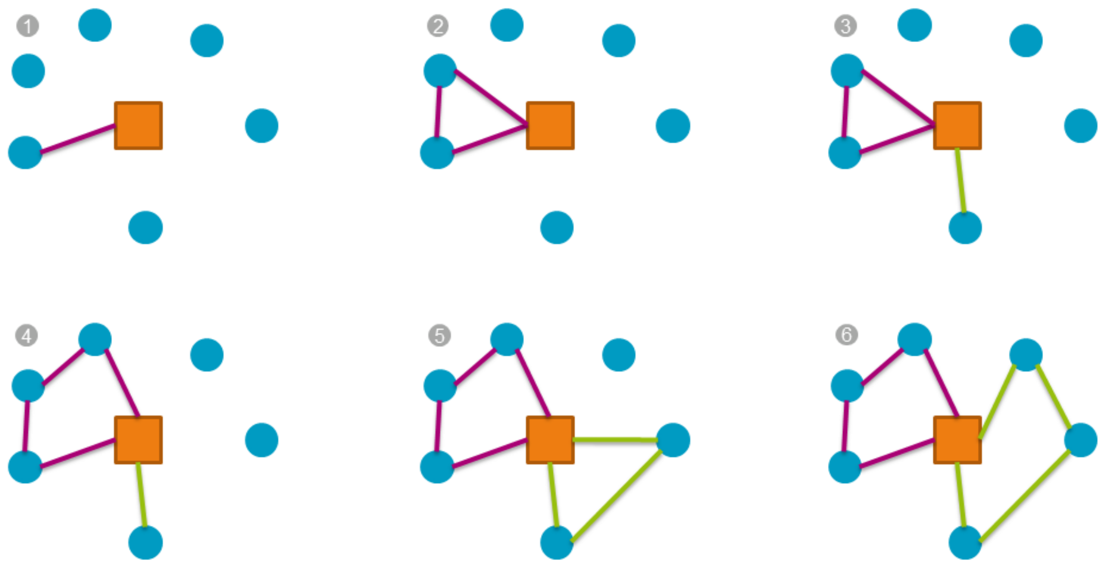


Figure 2.6: Parallel Insertion as illustrated by ORTEC (2019)

Figure 2.7: Example of a 2-opt improvement iteration as illustrated by ORTEC (2019)

CROSS-Exchange

Cross-exchange is the second improvement heuristic that we discuss in this section. Its logic is similar to the k-opt algorithm, which is described in the previous paragraph. Taillard et al. (1997) explains CROSS-exchange for the VRP. They state that the cross-opt heuristic has the possibility to select any arbitrary set of edges between customers from any route in the current solution. That makes it different to the k-opt algorithm, in which only edges inside a route are exchanged. The principle of CROSS-Exchange illustrated in Figure 2.8.

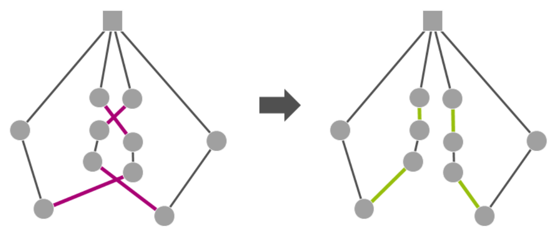


Figure 2.8: Example of a cross-opt improvement iteration as illustrated by ORTEC (2019)

2.2.2 Metaheuristics

Laporte (2009) classifies metaheuristics in neighbourhood-based algorithms, population-based algorithms and learning methods. This section only discusses some of the most important neighbourhood based metaheuristics. Appendix C describes the most important population based algorithms and learning methods. The classification of metaheuristics according to Laporte (2009) is illustrated in Figure 2.4.

Simulated Annealing

Kirkpatrick et al. (1983) was the first to describe simulated annealing in the context of combinatorial optimization problems. The basis of simulated annealing can be found in the field of statistical mechanics. The method introduces a method for accepting solution which are worse than the current solution. In each iteration, a worse solution is accepted with probability p. The probability p decreases as the process continues. The probability of accepting a worse solution is given by: $\exp\left(\frac{\Delta E}{T}\right)$. Translating this to a combinatorial optimization context, ΔE is the difference between the newly found solution and the current solution. T, is in this case the cooling parameter. After finishing a markov chain of simulated annealing, the cooling parameter reduces. T becomes $k * T_{old}$. In simulated annealing $k < 1$. This means that the value of T reduces in each iteration. A direct consequence of decreasing acceptance probability p during the process is that in the beginning, many worse solutions are accepted. At the end of the process, it is unlikely that solutions are accepted that deteriorate the objective function. Pseudocode for simulated annealing is given in Appendix

D. Wang et al. (2015) presents a parallel strategy for simulated annealing. In their procedure, calculations are spread over multiple threads. This allows to arrive faster at better solutions.

Tabu Search

Major work that introduced Tabu Search(TS) to solve VRPs was done by Osman (1993) and Gendreau et al. (1994). Just like simulated annealing, tabu search allows the acceptance of worse solutions. In particularly, TS always explores the complete neighbourhood and moves to the best solution in this neighbourhood. By accepting solutions that deteriorate the objective function, it is possible to return to solutions that have been visited already. To overcome this, solutions that have been visited are excluded from being visited again for a number of iterations. The solutions that cannot be visited again are stored on a tabu-list. To implement tabu search, Osman (1993) identifies 4 different aspects:

- **Forbidding Strategy**
  This strategy describes what is added to the tabu-list. This can be a complete solution, or a specific operator.
- **Freeing strategy**
  This strategy describes what is removed from the tabu-list.
- **Aspiration Strategy**
  This strategy describes if, and how, the restrictions of the tabu-search can be violated.
- **Stopping criteria**
  This criteria determines when the tabu-search algorithm is stopped.

Cordeau and Maischberger (2012) apply tabu search for instances of the MDVRP. They run an iterated tabu search in parallel on multiple CPU cores using different starting solutions. The algorithms, which run in parallel, exchange information after a pre-defined time. For a large number of MDVRP instances, the iterated tabu search provides new best found solutions.

Variable Neighbourhood Search

A first metaheuristic in the class of VLNS is Variable neighbourhood Search(VNS). An explanation of this metaheuristic and its applications can be found in the work of Hansen and Mladenović (2001). Variable neighbourhood search is a VLNS metaheuristic that changes the neighbourhood during the search. This means that during the search, neighbourhood search operators are used in an iterative manner to change the current solution. By changing (randomly) the local search method in each iteration, a variable neighbourhood is created. In the earlier work on VNS, only favourable solutions were accepted. In later works, simulated annealing type of acceptance criteria are applied (see Xiao et al. (2014)).

Very Large Neighbourhood Search

Very Large Neighbourhood Search (VLNS) is a sub-class of neighbourhood based metaheuristics. This is illustrated by Figure 2.4. Pisinger and Ropke (2010) explain that VLNS is a specific class of metaheuristics that use very large neighbourhoods. They explain the concept of a neighbourhood as follows. Suppose X is the set of all feasible solutions (x) to a given combinatorial optimization problem. The neighbourhood of this solution is called $N(x)$. $N(x)$ is a subset of X, which can be reached with a certain algorithm. Thus all solutions that can be created after perturbing solution x with a (local search) algorithm form N(x). The number of neighbour solutions that can be visited is dependent on the solution algorithm. The metaheuristics in this section create a large number of neighbour solutions.

Adaptive Large Neighbourhood Search The second metaheuristic which is part of the VLNS class is Adaptive Large Neighbourhood Search (ALNS). ALNS was first introduced by Pisinger and Ropke (2007). The method allows to select multiple methods for perturbing a solution. In VNS the selection of methods is done at random. ALNS tries to select the different methods a bit more clever. During the search, the probability of selecting a method is dynamic. Based on the performance of a perturbation method, the probability of selecting a certain method is increased or decreased. Stepwise, the method of Pisinger and Ropke (2007) looks as follows. First an initial solution is created. Pisinger and Ropke (2007) use a regret-2 heuristic in this construction phase. Next they use different methods to remove customer orders from a solution. Using a roulette wheel procedure, a combination of removal and recreate method is chosen. At the start of the search each combination of the removal and recreate method has equal probability of being selected. The selected combination of removal and recreate method is executed and the performance is evaluated. If the objective function is improved, the probability of selecting this combination of removal and recreate method in the next iteration, is increased. The process is repeated for a certain number of iterations.

Ruin And Recreate Ruin and recreate is the last technique within the VLNS class that we discuss. The method was first discussed by Schrimpf et al. (2000). Ruin and Recreate is often used within ALNS of Pisinger and Ropke (2007). Schrimpf et al. (2000) propose a local search metaheuristics that removes a significant amount of customer orders from an existing solution with the use of a destroy operator. After destruction, the method tries to repair the solution in a different way. This is done with the help of a repair methods.

2.3 Application of large neighbourhood search Metaheuristics

In the previous section, we gave a general overview of some of the most important metaheuristics for this study. This provides general understanding of the solution algorithms which are available to solve VRP instances. In this section we focus on the application of one class of metaheuristics: VLNS metaheuristics. More specifically, we look to the application of these metaheuristics to the VRP of the retail client. More specifically, we pay attention to three major aspects of the VRP of the retail client. These aspects are: large case size, heterogeneous fleet and multi-depot.

2.3.1 Large neighbourhood search and large Sized Cases

In general the applicability of ALNS for real-life and large sized cases is already proven by Simons (2017). New solution methods, which are developed in literature, are often tested on small scientific cases. Therefore their performance on larger real-life cases may not always be satisfying. To reduce computation times, metaheuristics often utilize ways of reducing neighbourhood sizes. This reduction comes at the cost of an increasing probability of missing out promising results. An interesting way of reducing the neighbourhood size for the MDVRP is found by Salhi et al. (2014). The authors claim to reduce computation time by 80% while finding better results with their algorithm. They do this by defining borderline customers which are located somewhere in the middle between 2 depots. Only the customers located close to these borderline customers are then considered in the improvement phase. Although the authors use this neighbourhood reduction in a VNS algorithm, it could be easily modified to apply in ALNS.

2.3.2 Large neighbourhood search MDVRP

A complete literature review on the MDVRP is given by Montoya-Torres et al. (2015). They mention that most of the available procedures for finding solutions to the MDVRP in literature consist of Tabu Search, Simulated Annealing and genetic algorithms. For the state-of-the art genetic algorithms we refer to Vidal et al. (2014) and Vidal et al. (2012). Not much metaheuristic procedures for the MDVRP make use of LNS until 2014. Also the work done on this topic after 2014 is very limited.

The survey by Karakatič and Podgorelec (2015) compares the performance of different metaheuristics. In the survey, the performance of different population search metaheuristics is compared with the performance of local search metaheuristics. The performance of these metaheuristics is evaluated agains an exact algorithm. The survey concludes that Iterated Tabu Search (ITS) is the best performing metaheuristic on 5 out of 6 benchmark cases that are used in the study. The second best known metaheuristic is known to be ALNS by Pisinger and Ropke (2007). Both methods outperform the genetic algorithms developed by Karakatič and Podgorelec (2015) and state-of-the art ant-colony optimization methods.

Li et al. (2015) claim that the standard ALNS is improved by combining it with a iterated local search. This claim is based on benchmark cases for single depot VRP. Substantiation for the MDVRP is not given. That ALNS produces robust and good solutions to the MDVRP is further substantiated by Mancini (2016). The author has adopted a matheuristic based on ALNS. The method starts by ruining part of an initial solution. This selects the neighbourhood that is evaluated. Only a limited number of tasks is selected to be removed from the solution. Since only a limited number of orders is removed, an exact method can quickly find the (local) optimal way to repair the solution. The method proofs to be effective in exploring many neighbourhoods very efficiently. It also proofs to be easily applicable to other instances of the VRP with the assignment problem. The disadvantage of the proposed matheuristic is that it requires a MILP-formulation which can be hard to develop for rich and real-world VRPs. Grangier et al. (2016) applies ALNS to the 2-echelon VRP with time windows. Due to its many restrictions, the problem formulation shows many similarities with the MDVRP. The authors propose a method of reducing the size of the neighbourhoods which has a limited impact on the solution quality, but significantly reduces the computation time.

It is obvious that that the methods for ruin and recreate which are used in ALNS have strong influence on the solution quality. In the original work by Pisinger and Ropke (2007) 7 different methods for ruining a solution were presented. These included: random removal, worst removal, related removal, cluster removal, time oriented removal, historical node-pair removal and historical request pair removal. Some additional destruction heuristics found by Grangier et al. (2016) are trip removal and least-used-vehicle removal. A route destruction operator is advised by Mancini (2016). This operator selects a defined number of customers from a route which are unplanned from the solution. The author also mentions that cluster removal is likely to be successful in the start of the search. Later random removal may be more likely to find improved results. A new removal algorithm is introduced by Emeç et al. (2016). Their route neighbourhood removal algorithm aims to select a pair of eligible transports from different routes. These transports are unplanned from the existing solution. The eligibility constraint works as following. A transport from route 1 and a transport from route 2 are considered to be removed. This removal is only eligible if it is possible to plan the transport from route 2 directly after the transport of route 1. In this check, time window constraint and capacity constraint are taken into account. From the computational study that the authors execute, this operator has the highest frequency of finding new best solutions.

The most commonly used methods for repairing a method are best insertion and regret insertion. Literature often pays more attention to removal methods rather than repair methods. Two new repair methods are introduced by Alinaghian and Shokouhi (2018). These methods use an additional 'noise function' to the two basic repair methods mentioned above. This noise function allows for more freedom in selecting the insertion location. Another point of view in repairing a solution stems from Emeç et al. (2016). The standard ALNS of Pisinger and Ropke (2007) uses a removal and a insertion phase. Emeç et al. (2016) includes an extra layer in between the removal and insertion phase. They call this the "vendor Selection/Allocation phase". This phase selects a feasible depot from the list of all possible depots. This selection is done based on a total of 6 operators. For a description of these operators, we refer to Emeç et al. (2016). This is of course only possible in a multi-depot variant of the VRP. The authors also propose 2 new insertion methods. The first insertion method, Regret-k insertion, makes use of a regret factor for selecting a customer to insert. The regret factor is calculated by the difference in objective function when inserting the transport in the best route and the k^{th} best route. The second insertion algorithm proposed by the authors is 'Greedy Insertion with New Route Openings (GIN)'.

Another aspect of ALNS which has an influence on the solution quality is the adaptive weight adjustment procedure. This procedure determines how large the probability becomes of an operator(both ruin and recreate) to be selected in the next iteration. In the original work of Pisinger and Ropke (2007) the weight updating is done based on the value of the objective function as shown in equation 2.1.

$$w_{i,j+1} = w_{ij}(1-r) + r * \frac{\pi_i}{\theta_i} \tag{2.1}$$

In equation 2.1, π_i denotes the value of the objective function which was achieved by the operator. θ_i denotes how often the operator has already been used. w_{ij} is a measure for the weight that is assigned to operator i in iteration j. These weights translate into probability of selecting operator i in iteration j as: $p_{kj} = \frac{w_{kj}}{\sum_i w_{ij}}$.

2.3.3 Large neighbourhood search and Heterogeneous Fleet

A literature review of the solution methods for the Heterogeneous Fixed Fleet VRP is given by Çağrı Koç et al. (2016). The authors do not mention any successful applications of large neighbourhood searches on this variant of the VRP. The lack of ALNS application on HVRP is part of a general trend. Çağrı Koç et al. (2016) mention that the general trend is to develop highly accurate models which lack simplicity and short computation times. Some of the best algorithms that are developed for the HVRP are a threshold accepting metaheuristic by Tarantilis et al. (2004) and a clustering based heuristic from Gencer et al. (2006). The current state-of-the art solution algorithm is a mathematical based heuristic by Naji-Azimi and Salari (2013). This method shows many similarities to the matheuristic by Mancini (2016), which we discussed earlier.

2.4 Conclusion

With this discussion, we are able to answer the first set of research questions:

1. What can literature teach us on...
   (a) The problem specific Vehicle Routing Problem?
   (b) Large Neighbourhood Search techniques and their applicability for the customer specific VRP.
   (c) The latest developments of Large Neighbourhood search Techniques and their applicability for rich and real-life cases.

We have mentioned some of the relevant extensions of the classical VRP. Based on the analysis in Chapter 1, we can conclude that not one single extension is suitable for modelling the routing problem of the retail client. We conclude that we must extend the classical VRP with the aspects of the multi-depot VRP. We prefer this extension over the 2-echelon variant, since routes from central depot to hubs are not considered in this research. Moreover, we saw a variant in which orders must always be delivered in a certain time window.

An interesting method of reducing the neighbourhood size in a VNS setting was given by Salhi et al. (2014). We could make use of this method to reduce computation time of the ALNS algorithm of ORTEC. This could be useful for solving large VRP cases. Moreover, a new type of depot-selectors is included by Emeç et al. (2016). Including this as a new adaptive layer in the ALNS framework of ORTEC could be interesting since it has shown promising results on the 2E-VRP. This VRP variant shows close resemblance to the MDVRP of the retail client. An ALNS based matheuristic which is able to quickly explore large neighbourhoods is found by Mancini (2016). It has proven to be successful in solving MDVRPs, but requires a MILP formulation. Also additional methods for destroying solutions are found. These methods provide better solutions in less computation time for the MDVRP. Solution methods for the heterogeneous VRP are not much studied. Examples of LNS algorithms for this variant are even more rare to find.

Chapter 3

Current Situation

This chapter aims to create better understanding of the vehicle routing problem of the retail client and the software of ORTEC. First, Section 3.1 provides a description of the software that we use to find solutions to the VRP of the retail client. Section 3.2 provides a detailed description of the business case of the retail client. Finally, Section 3.3 defines Key Performance Indicators (KPIs) to measure the performance of the current solution strategy.

3.1 CVRS Description

To find solutions to the VRP of the retail client, we use the CVRS software of ORTEC. We explain in this section briefly how the software works. To solve VRPs, CVRS uses a model that consists of entities and their attributes. The algorithms that are programmed in CVRS transform the attributes of these entities. By filling these attributes, or connecting the various entities, a solution to the VRP is created. Thus, to increase the understanding of CVRS and to provide a formal problem description, we describe the entities that are used as input and the entities that are used as output in Paragraph 3.1.1 and Paragraph 3.1.2 respectively. Moreover, we provide a short overview of the algorithms that are used by CVRS to create solutions to VRPs in Paragraph 3.1.3.

3.1.1 Input Entities

The first entity in the model is the *depot*. The depot is the location from which an order is picked up. The depot is also the start and end location of a trip. Each depot has 2 attributes.

- *Id*: Unique identifier by which a depot is characterised
- *Capacity*: Total amount of products that can be delivered from the depot.

The second entity of our model is the *task* entity. This entity can be either one of two types: pickup or delivery. As a result of this, an order always contains 2 tasks. One for pickup and one for delivery. We introduce the order entity below. Each task has the following attributes.

- *Id*: Unique identifier by which a task is characterised.
- *address Id*: This provides the software with the information where the task must be executed. A pickup task is located at a depot. The delivery task is located at a customer address.

- *Order Id*: This connects the tasks to the corresponding order.
- *Duration*: This contains the information on the duration of the task. A numerical value denotes how many seconds a task takes to execute.
- *Amount*: The amount of product that is delivered. Only deliver tasks have an amount specified.
- *Time Window start*: Start of the time window from which the task may be executed.
- *Time Window end*: End of the time window from which the task may be executed.

The third entity in the CVRS model is an *order*. An order always consist of a pickup and delivery task. An order forms the connection between the pickup and delivery task. All orders in the model are placed by an customer. No specific information on customers is modelled in CVRS. An order has the following attributes.

- *Id*: Unique identifier by which a order is characterised.
- *allowed Depots*: The depots from which the pickup task can be executed. If dynamic sourcing is applied, this attribute is a string that contains all depot Ids.

The next entity in the model is called the *resource*. The resource the vehicle of a given type executes a trip. We introduce the trip entity next. A resource has the following attributes.

- *Id*: Unique identifier by which the resource is characterised.
- *Capacity*: The capacity denotes the quantity of orders that can be transported by the resource.
- *Resource kind*: Number value that distinguishes a between driver/vehicle etc.
- *Home address Id*: Address of the depot at which the resource is located.
- *Forbidden addresses*: Ids of addresses that cannot be visited by the resource.
- *Task Duration*: Duration of load tasks for a vehicle. This is a fixed set-up time for loading the vehicle in addition to the task duration, which we introduced before.

The last input entity that we mention here is the *Trip*. The trip is used to create a sequence of tasks that are executed by a resource. As a result of that the trip has the following attributes.

- *Start address*: Address at which the trip starts.
- *End address*: Address at which the trip ends. Since a resource is assigned to only 1 depot, this is the same as the “from address”.
- *Earliest Start Time*: The earliest moment at which a trip can start. This can be sooner than the actual start time of a trip.
- *Latest Finish Time*: The last moment at which a trip must finish. This means the last moment at which the resource must be back at the “end address”.
- *Cost set*: Denotes the different costs that are made when the trip is executed. A cost set contains fixed set-up costs, costs per stop, costs per hour, costs per stop and costs per distance.
- *Resource id*: Unique id of the resources that are used to execute the trip.
- *Priority*: Priority of the trip. Trips with priority 1 are more desirable to be used than trips with priority 3. We explain this in more detail in Section 3.2.

These are the most important entities and attributes that are used to find solutions to the VRP of the retail client.

3.1.2 Output Entities

The first entity that is used as output by CVRS is a route. A route contains a list of trips. Each trip in its turn consist of a list of stops. We explain the stop entity next. Note that a route can contain one or more trips. As a result, a route is a place holder for one or multiple trips. The trip entity is explained in Section 3.1.1. A route contains the following attributes:

- *id*: Unique id of the route.
- *number of trips*: Denotes the number of trips that are in a single route.
- *waiting time in seconds*: The total waiting time of the route according to the planning.
- *Number of Stops*: The number of stops that are included in the route.
- *Driving time in seconds*: The total driving time of the route.
- *Distance*: The total distance in kilometers of the route.
- *Duration in Seconds*: The total duration of the route.
- *Finish time*: Moment in time at which the route finishes.
- *Start time*: Moment in time at which the route starts.

The second entity that is used as output by CVRS is a stop. A stop in the software, means a physical stop in reality. This means that the vehicle stops to execute a task. (e.g. deliver a package) A stop belongs to exactly one trip inside a route. A stop has the following attributes.

- *departureDateTime*: Start time of the stop.
- *arrival Date Time*: End time of the stop. The arrival time plus duration equals the departure time. The departure time of the previous stop plus the driving time equals the arrival time of the next stop.
- *duration In Sec*: Time to execute the task that belongs to the stop.
- *waiting Time In Sec*: Time in which no task is executed.
- *driving Time In Sec*: Time to travel to next stop.
- *distance*: Distance to next stop.
- *stop Sequence*: Integer value that denotes in which order the stops are executed.
- *address Id*: Id denoting the address where the stop takes place.
- *order Id*: Id of the order that belongs to the task that is executed at a stop.
- *type*: Denotes whether a stop contains a delivery, break, depot or pickup task.

Thus, by creating a sequence of stops which belong to a task, CVRS creates a solution to the VRP of the retail client.

3.1.3 Algorithms

In most cases, CVRS finds solutions to VRPs in 3 phases. The first phase tries to construct a feasible solution, the second phase improves this solution using classical heuristics. The last phase further improves the solution using ALNS. In the construction phase, CVRS uses empty trips, and a list of tasks as input. Usually based on some greedy method, routes are built up. Two commonly used methods for building solutions are parallel insertion and sequential insertion. These methods are explained in Section 2.2. The first improvement phase uses a number of heuristics to improve an existing solution. CVRS does this by applying changes to the existing solution. Many classical heuristics are available in CVRS. Since the main focus on our thesis is on the ALNS phase, we do not provide details on the classical heuristics.

In the last phase of finding solutions to VRPs, ALNS is applied. Chapter 2 explains how ALNS works in more detail. In this section we focus on how it is applied at ORTEC. Many configurations of ALNS at ORTEC are possible. For instance, it is possible to apply different methods for selecting the removal of customer orders from the solution and methods to repair a partial solution. It is possible to select the same methods each iteration, but one can also decide to base the selection of these methods on the roulette wheel of Pisinger and Ropke (2010). The roulette wheel adapts the likelihood of a method to get chosen based on its performance. Recall from Section 2.2.2 that ALNS works by selecting methods for destroying part of the solution and repairing it in a different way. Currently 7 removal and 3 repair methods are programmed in the software of ORTEC. Table 3.1 shows the methods that are currently available in CVRS.

Table 3.1: Removal and Recreate methods that are included for standard ALNS

Removal Method	Recreate Method
Random Removal	Sequential Cheapest Insertion
Related Removal	Parallel Cheapest Insertion
Worst Removal	Parallel Regret Insertion
Random Cluster Removal	
Worst Cluster Removal	
Trip Removal	
Multiple Trip Sets removal	

Below we briefly explain the 7 removal operators.

Random Removal: This operator randomly removes a number or percentage of tasks from the solution. Referring back to Section 3.1.1, this means that the stop that belongs to the task, is no longer connected to a trip.

Related Removal: This operator removes a number of percentage of tasks from the solution that are related. Tasks are related to each other if they are located close to each other.

Worst Removal: This operator removes the worst number of tasks from the solution. The worst task is the task that increases the objective function the most.

Random Cluster Removal: This operator removes a random cluster of tasks. A cluster is a group of tasks that are located closely to each other. It differs from related removal in the sense that multiple groups can be removed.

Worst Cluster Removal: This operator is a combination of worst removal and random cluster removal.

Trip Removal: This operator removes all the tasks from a randomly chosen trip.

Multiple Trip Sets removal: This operator removes the tasks from multiple randomly chosen trips.

Below we describe the methods that are used to repair the solution after the removal operator is executed.

Sequential Cheapest Insertion: We refer to Section 2.2 in which we already explain Sequential insertion.

Parallel Cheapest Insertion: We refer to Section 2.2 in which we already explain Parallel insertion.

Parallel Regret Insertion: Works similar to parallel cheapest insertion, but tasks are inserted in order of decreasing regret factor. The regret factor is calculated as in equation 3.1.

$$ObjectiveFuction_{2^{nd}BestOption} - ObjectiveFunction_{BestOption} \tag{3.1}$$

ALNS parameters

Each of the aforementioned removal and repair operators have a number of parameters which can be configured. For the removal operators, the following settings must be configured.

Acceptance Threshold: The software of ORTEC does not accept solutions that deteriorate the objective function. If one iteration of ALNS finds a solution that is worse than the current solution, but not worse than the allowed threshold, Local Search is applied. This setting thus decides when an additional local search phase is initiated. Only if Local Search finds a better solution, the solution is accepted.

Only do local search if improved: This setting can make the previous setting redundant. If this setting is set to true, local search is only applied to better solutions.

% of tasks to remove: This determines the percentage of tasks from the total number of tasks is removed.

Number of tasks to remove: This determines the number of tasks that is removed from the solution. If this is used in combination with the percentage of tasks to remove, then the minimum of both is chosen.

Estimator: Estimator that is used to quickly assess if a change in the solution results in an improvement or not. This does not necessary have to be the same as the optimization objective.

Re-optimize depot after removal: Re-optimize all depots after a change has been made. Using this option, increases computation time.

Only re-optimize depots changed trip: Do not change trips that belong to a depot which has not been changed by the ALNS iteration.

Use heuristic for dynamic sourcing: Use only a limited number of depot options instead of considering all options.

Always re-evaluate depots: In each step of the local search, all possibilities for depot allocation are reconsidered.

Depots per heuristic step: Can only be used if the setting: *Use heuristic for dynamic sourcing* is set to true. This limits the number of depots that are considered in each ALNS iteration.

Maximum number of depots: This limits the number of depots that are visited in each route. For our case it is not allowed to visit multiple depots in a single trip.

Number of solutions: If the first considered solution does not lead to an improvement, and this setting is equal to 2, also the second best solution is considered.

Not only the removal operators have a number of parameters that must be determined. Also for the repair operator, several parameters exist. For the repair operators, the following settings must be configured.

Maximum number of failed insertions per group: This setting reduces the number times that a task is inserted in a certain group.

Maximum number of failed insertions per trip: This setting reduces the number of times that tasks are tried to be inserted in a certain trip.

Estimator: Estimator that is used to quickly assess if a change in the solution results in an improvement or not. This does not necessary have to be the same as the optimization objective.

Only consider empty trips: Only insert tasks in empty trips.

Maximum number of failed insertions per group in a trip: Maximum number of times that an unplanned order is tried to be inserted in a certain trip.

Fix ordering load/unload tasks at address: This setting determines in which order the load and unloading tasks are inserted in a route. If this setting is true, all load tasks are planned at the start of the trip.

Only consider non-empty trips: For repairing the solution, only non-empty trips are used. This means that no extra trips are used when repairing a solution.

3.2 Case Description

So far we described how CVRS can be used to find solutions to VRPs. Now we provide more details on the business case of the retail client, which we introduced in Chapter 1. We explain additional constraints to possible solutions, the optimization objective that is used for the retail client, and the cases that we consider.

3.2.1 Restrictions

In Chapter 2, we identified many different variants of the VRP. The case of the retail client has a combination of these restrictions. We provide a discussion on the constraints that are included in the VRP of the retail client. This gives us a better understanding of the case of the retail client. These constraints are taken into account when formulating solution strategies for solving the VRP of the retail client.

Table 3.2: Fleet composition

[redacted]

In Section 1.1 we mention that we consider a VRP with a heterogeneous fleet. Table 3.2 shows the vehicle types that are available and their characteristics. The heterogeneous fleet implies the following restrictions to the solution of the VRP.

1. Type 1 vehicles must be used as much as possible. This restriction is not a hard restriction. The utilization level of type 1 vehicles is not quantified by the retail client.
2. Type 1 vehicles must be used for as long as possible. This means that once a type 1 vehicle is used, the driving time must be as long as possible.
3. Type 1 vehicle cannot be used to serve customers which have a deliver location far away from the depot at which the vehicle is stationed. No formal definition is given for what distance is acceptable.
4. After all the type 1 vehicles are used, type 2 vehicles must be used with the same restrictions as type 1 vehicles. For these type 2 vehicles the same rules apply as for type 1. It is desirable that the type 2 vehicles are always used and serve customers with a location close to the depot.
5. The type 3 vehicles are only used on a need basis.
6. Type 3 vehicles are used to drive large distances. This means that they are often used for trips to customers that cannot be served by type 1 and 2 vehicles.
7. Type 4 vehicles are undesirable to use. These vehicles are used as flexible capacity. It is very expensive to use these vehicles. As a result, they can only be used if regular capacity is insufficient to fulfil all demand.

The aforementioned restrictions are merely preferences of the retail client. It is not quantifiable how often these restrictions must be met. As a result of that, the restrictions are not modelled in CVRS. Instead, ORTEC attempts to meet these restrictions as much as possible through clever configuration of the different costs sets for each vehicle. These cost sets are explained in Section 3.2.3.

The second set of restrictions are related to the depots.

1. The capacity of the depots is restricted. We distinguish two types of capacity restrictions. First the load-capacity of the depot. Second, the capacity of the vehicles which are assigned to the depot, restricts in some cases the total capacity of the depot. If q_k is the loading-capacity of depot k and c_{lk} is the capacity of vehicle l that is assigned to depot k, then the depot capacity Q_k is given in Equation 3.2.

$$Q_k = \min(q_k, \sum_l c_{lk}), \forall k \in K \tag{3.2}$$

2. Co-loading is not allowed. Earlier we mentioned that during a trip, vehicles can visit only 1 depot. This is also known as co-loading at ORTEC. Co-loading at ORTEC means the following: A vehicle starts at a certain depot. Before visiting customers, the vehicle visits another depot. This is not allowed for the routing problem of the retail client. We must take this into account when providing solutions for the VRP of the retail client.

The fourth set of business rules concerns delivery windows. In the VRP of the retail client, a restriction is imposed on the delivery window in which the customer site or depot may be visited.

1. Each customer has a time-window in which the customer must be visited. Each customer must be visited in their preferred time-window when the order is planned on a route. As a result, a solution to the VRP is not acceptable if a customer is planned to be visited outside its stated time-window. This means that we cannot make use of the soft formulation of the time-window constraint. We refer to Section 2.1.3 for the difference between hard en soft formulation of the time-window constraint.

The previous restrictions impose constraints concerning how the solution for the VRP may look like. At last we present a restriction to the computation time of the solution algorithms. The computation time of the algorithm is restricted to 35 minutes. This norm is stated by the retail client. As a result, the solution strategies that we design must meet this requirement.

3.2.2 Optimization Objective

Each optimization problem must have an objective to optimize. In this section we describe which optimization objective is used to find a solution to the VRP of the retail client. In the case of the retail client there are multiple optimization objectives. Below we list the optimization objectives that are used for the retail client. The objectives are presented in order of decreasing importance.

1. Number of tasks planned
   This objective is measured by counting the total number of deliver tasks that are planned in a transport route. The software aims to maximize the number of tasks planned for the given resources.

2. Plan costs
   This function measures the total costs that result from all planned transport routes. A solution with the lowest costs is preferable.

3. Duration
   This objective measures the total time that all the routes need to visit all customers. The total duration is measured in hours. It is calculated as the sum of the duration of individual trips. A solution with smaller duration is preferred over a solution with longer duration.

To optimize according to different objectives works as follows in the software of ORTEC. The routes are always optimized according to the first optimization objective. In this case that is the number of tasks planned. In case multiple solutions are found that have an equal number of tasks planned, they are compared on the second objective: costs. If then still the solutions are equal, the software looks at the third objective, which is in our case total duration. It is necessary to maximize the number of tasks planned for the case of the retail client. The second objective is namely the minimization of the plan costs. If CVRS is not configured to maximize the number of tasks planned, no tasks are planned at all. This is because a solution with 0 tasks planned is cheaper than a solution with at least 1 task planned.

3.2.3 Cost sets

In the previous section, we have seen how the objective function of the VRP of the retail client looks like. We saw that the primary objective is to maximize the number of planned tasks. Next, the aim is to minimize the *plan costs.* To calculate the plan costs, each type of vehicle in the problem set is assigned a set of costs. In total, we distinguish between 4 types of costs: *costs per hour, costs per trip, costs per distance and costs per stop.* As we mentioned before, the retail client uses a heterogeneous fleet. As a result, the cost settings are different for each vehicle type. For confidentiality reasons, we do not present the cost figures here. Instead, we refer to confidential Appendix E for the different cost configurations. These figures show that the costs per trip are by far the largest for type 3 and 4 vehicles. This is done to ensure that first type 1 and type 2 vehicles are used.

3.2.4 Test cases

For testing purposes, we use 20 cases that are defined by the retail client. These cases and their characteristics are described in Table 3.3. We see that each of the cases has a different number of depots and a different number of orders that need to be delivered. We use these cases to test the performance of the solution that we design in Chapter 4. Also we use the cases to validate the results of the solution algorithm that is currently used by ORTEC.

Table 3.3: Overview of 20 test cases

[redacted]

3.3 Current Solution Algorithm

To find solutions to the VRP of the retail client, currently a two-stage approach is used. This approach consist of a construction step and a local search step. Since we do not intend to make large adjustments to this method later in our research, we do not discuss this method here. For a detailed explanation of the current construction method, we refer the reader to confidential Appendix F. Instead, here we focus on the performance of this construction method. Later we compare the outcomes of our solution strategies with this current method. The performance of these solution techniques is assessed on 20 cases. We introduced these cases in Table 3.3. Table 3.4 shows the key performance indicators for each of the 20 cases. The table shows the total costs, total driving time, total distance and number of vehicles used of each type. In Section 3.2, we mention that vehicle type 1 must always be used. From Table 3.4, we see that not every case contains type 1 vehicles. For the cases in which vehicle 1 types are available, (cases 16, 17 and 18) we see that not all type 1 vehicles are used. The table shows for each case the plan costs, number of tasks planned and the total duration. Finally, Table 3.4 shows the runtime of the solution algorithm, that was described in the previous section, for each of the 20 cases. It can be seen that for some cases the computation time exceeds the 35 minutes threshold, which was set by the retail client. This is caused by the fact that CVRS completes executing the method it is currently executing before reporting on the KPIs.

3.4 Conclusion

This section provides more details on the VRP of the retail client for who we develop a solution strategy. We have discussed the restrictions that are imposed on the solution to the VRP of the retail client. The section was completed by an explanation on how ORTEC currently solves the case of the retail client. This gives us enough insight to answer the second set of research questions:

1. How does the current situation look like?
    (a) How does the VRP of the retail client looks like?
    (b) Which solution algorithm is currently used to to solve the VRP of the retail client?
    (c) How does the current solution algorithm perform?

Our discussion of the restrictions pointed out a number of limitations that are imposed to possible solutions. In the discussion of the restrictions we have seen that we work with a heterogeneous fleet. Also the calculation time that is available for finding a solution to the VRP is restricted to 35 minutes. Moreover, we have seen that the delivery time window in which customers are visited is restricted. All these real-life restrictions must be taken into account when finding a solution to the VRP of the retail client. We have described the characteristics of 20 cases that we use for testing our solution design. Each of these cases has a different number of depots, orders or routes. In Section 3.3, we present an overview of the performance of the current solution method on 20 benchmark cases. We expand the performance measures beyond the performance indicators from the objective function. By including these additional performance indicators, we assess the performance of the solution algorithm on the business rules of the retail client.

Table 3.4: Overview of the performance of the current solution algorithm on 20 customer cases

[redacted]

Chapter 4

Solution Design

This chapter provides a description of the solution strategies that we test. Based on the context analysis and literature study, we propose five different ALNS based solution strategies. Each of the strategies addresses the order-to-depot assignment differently. The first strategy is a standard ALNS application in the software of ORTEC. Section 4.1 describes this strategy. The second strategy extends this first strategy with additional repair operators. Section 4.2 provides a description of this strategy. In both strategies, the assignment of orders to depots can still change. Solution strategy 3 does not allow the depot assignment to change during the execution of ALNS. This means that after the construction phase, the order-to depot assignment does not change. Section 4.3 contains a description of this solution strategy. The fourth strategy first solves an assignment problem. The solution to this assignment problem determines the order-to-depot assignment. During the construction and improvement phase, this depot assignment does not change. Section 4.4 discusses how this strategy works. The fifth solution strategy is a hybrid combination of the previously mentioned strategies. In the fifth strategy, the depot-assignment of a subset of all orders is allowed to change during the solution algorithm. Section 4.5 illustrates this idea. Each of the aforementioned strategies uses the already existing construction algorithm. We refer to this construction algorithm that we describe in Section 3.3 as the *ORTEC construction algorithm.* Table 4.1 presents an overview of each of the aforementioned strategies, in which sections they can be found and some of the most important characteristics.

Table 4.1: Overview of Solution Strategies that are included in this thesis

Strategy	Construction Method	Depot Assignment Fixed	Set of ALNS operators	Section
Basic ALNS	ORTEC Construction	Dynamic Assignment	Standard ORTEC operators	4.1
Extended Basic ALNS	ORTEC Construction	Dynamic Assignment	Standard ORTEC operators + New Route Selector	4.2
ALNS with Fixed Depot Assignment	ORTEC Construction	After Construction	Standard ORTEC operators	4.3
ALNS-FDATTP	Adjusted ORTEC Construction	Before Construction	Standard ORTEC operators	4.4
Hybrid Method	Adjusted ORTEC Construction	Before Construction	Standard ORTEC operators	4.5

4.1 Basic ALNS Strategy

This section describes the first solution strategy. We call this strategy the *Basic ALNS Strategy*. This strategy uses ALNS algorithms that are currently available in CVRS. This strategy is based on the work of Pisinger and Ropke (2010). However, the implementation in CVRS is done differently. In order to understand how the basic ALNS strategy works, let us introduce the following notation.

R^-: Set of all removal operators, $R^- = \{1, 2, ..n\}$. Table 4.2 shows the removal operators that are included in the Basic ALNS strategy.

R^+: Set of all repair operators, $R^+ = \{1, 2, ..m\}$. Table 4.2 shows the repair operators that are included in the basic ALNS strategy.

P_r^-: probability with which removal operator r^+ is selected.

P_r^+: probability with which removal operator r^- is selected.

s_0: Initial solution. This is the solution that is found by the ORTEC solution algorithm.

s_n: Neighbour solution.

s^*: Best found solution.

Algorithm 1 shows the pseudo code for the basic ALNS strategy. In line 1 the probabilities P_r^- and P_r^+ are initialized. CVRS initializes the probabilities different from the work of Pisinger and Ropke (2010). Next in Line 2, the best found solution is initialized. Next, ALNS is run for a number of iterations. In each iteration a removal method r^- and a repair method r^+ are selected. Using these removal and repair methods, a new solution is created. This is done in line 4-8 of Algorithm 1. Lines 9 -20 describe the logic of accepting solutions. We mentioned before that CVRS does not accept worse solutions. If the new solution s_n is better than the best found solution s^*, a local search is applied to see if further improvements can be achieved. Since the solution was improved, the probability that the methods r^+ and r^- are selected increase. If the new solution s_n is worse than s^*, but within the acceptance threshold, local search is applied to improve s_n. If s_n is now better than s^*, it is accepted as the new s^*. The probability that the methods r^+ and r^- are selected increases. If s_n is still not better than s^*, the new solution is not accepted. The probability that methods r^- and r^+ are selected decreases.

Table 4.2: List of removal and repair operators that are included in the first ALNS strategy

Removal Operator	Repair operator
Worst Removal	Cheapest Insertion
Random Removal	Parallel Cheapest Insertion
Related Removal	Parallel Regret Insertion
Random Cluster Removal	
Worst Cluster Removal	
Remove Multiple Trips Sets Removal	
Trip Removal	

Algorithm 1: Pseudocode of the basic ALNS strategy

Initialize: $P_r^- = P_r^+ = \frac{1}{n*m} \forall r$;
set $s* = s_0$;
while *Maximum number of iterations is not reached* **do**
  Select a method r^- using probabilities P_r^+;
  Select a method r^+ using probabilities P_r^+;
  According to method r^- remove k orders from the solution;
  According to method r^+ re-plan the k orders;
  Call the newly found neighbour solution s_n;
  if s_n *better than* $s*$ **then**
    Apply local search to improve s_n;
    Accept neighbour solution as best solution and set $s* = s_n$;
    Update $p_r^-, p_r^+ \forall r$;
  if s_n *worse than* $s*$*, but within Acceptance Threshold* **then**
    Apply local search to improve s_n;
    if s_n *better than* $s*$ **then**
      Accept neighbour solution and update $s* = s_n$;
      Update $p_r^+, p_r^- \forall r$;
    else
      do not accept s_n as new solution and go to next iteration;
    end
end

4.2 Extended Basic ALNS

This section describes the second solution strategy that we test. We call this the *Extended Basic ALNS Strategy*. This strategy extends the basic ALNS strategy with additional ALNS operators. Based on the research of Emeç et al. (2016), we have reason to believe that their newly proposed operators can improve the current ALNS framework of ORTEC. Emeç et al. (2016) propose an ALNS application, in which they also face the decision of assigning customers, to what they call satellite facilities. These satellites correspond to the various depots of the retail client. Emeç et al. (2016) use this aspect of their case to develop an additional set of operators. They refer to these operators as vendor selection/allocation algorithms. We propose to transform these operators to route selection/allocation algorithms. We incorporate this into the existing ALNS methods of ORTEC as follows. The ALNS algorithm first selects a method to remove a number of tasks from a solution. Afterwards a repair method is chosen to try and plan these tasks at a certain location in the partial solution. If no route selector is applied, this means that potentially a task can be planned in a route that belongs to any of the depots. If we now include a route selection method within the repair operator, this is no longer possible. Instead, the route selector will make sure that the software only tries to plan a task on a sub-set of all routes. Below we list the route selector that we propose to add to the first set of ALNS operators, which are shown in Table 4.2. From the operators that are proposed by Emeç et al. (2016), this is the only operator that we can implement within the technical possibilities of CVRS.

- Select all routes from the depot that is closest to current task.

We include this selector to reduce the number of assignments that are tried that do not result in improvement. Inserting tasks in any routes that do not belong to their closest depot is likely not to lead to an improvement. This depot selector is based on the Node Neighbourhood Vendor Selection of Emeç et al. (2016). It only includes the selector that selects the nearest depot. This operator selects a sub-set of routes that belong to the depot that is closest to the task that was removed. Then, using any of the aforementioned repair operators, the software only tries to plan this task on any of these routes. We refer to Algorithm 1 for an explanation of this strategy.

Each of the ALNS operators has a number of settings that we can configure. We already mention these settings in Section 3.1.3. We describe in Section 5.2 how we determine the best settings for these ALNS operators.

4.3 ALNS with fixed depot assignment

This section describes the third solution strategy that we include in this research. In the previous 2 solution strategies, the customer to depot assignment can change in the ALNS procedure. Although we cannot conclude from literature what the influence of this is on the performance of ALNS, it is likely to have an effect. This third strategy completely forbids any changes in the customer to depot assignment during the ALNS algorithm. This means that we need to provide CVRS with a order to depot assignment as input before it runs the ALNS algorithm. It is crucial that we provide a good depot assignment during the construction phase. The depot-assignment strategy that we propose uses the *ORTEC construction method* to determine the customer to depot assignment. The strategy starts by applying the algorithm that ORTEC currently uses on the complete problem. This step results in a feasible solution to the MDVRP of the retail client. Next, the case is split into single depot problems. This is done in the following way. The solution algorithm from ORTEC provides a set of trips on which tasks are planned. Each of these routes has a start and end location at one of the depots of the case. Thus we create sub-sets that contain all trips (and planned tasks) that are assigned to the depot. This is the second step in this strategy. Afterwards, we apply the basic ALNS strategy on each of these single depot problems. We refer to Algorithm 1 for the pseudo code of this strategy. In the ALNS step, we take the planned trips, which were determined earlier, as input for the ALNS algorithm. This means that s_n is now the solution to each of the single depot problems. This means that we do not execute a construction algorithm on each of the single depot problems. Lalla-Ruiz and Voß (2019) show competitive results of such a partial optimization approach for the multi-depot cumulative capacitated vehicle routing problem.

4.4 Depot Assignment by solving the Transport Problem

The fourth ALNS strategy that we propose is based on the work of Tansini et al. (2002). They show that a two-stage approach for solving the MDVRP, in which first an assignment problem is solved to determine the order to depot assignment, obtains high quality results. The assignment problem that is described by Tansini et al. (2002), is called the transport problem. Section 4.4.1 provides the formal definition of this problem. For further reference, we call this strategy ALNS-FDATTP. This ALNS based strategy starts by solving the transport problem. The transport problem determines how much goods are shipped between a depot and a customer. The solution to the transport problem results in a order to depot assignment. We formulate a mathematical model for the transport problem in Section 4.4.1. With the solutions to the transport problem, we divide the case into multiple single depot VRPs. For each of these single depot VRPs we apply the Construction algorithm that is currently used by ORTEC. After this step, we have a feasible solution for each individual depot. Per depot, we apply an ALNS algorithm in order to further improve the current solution. We apply the basic ALNS strategy, which is described in Algorithm 1. Afterwards, we combine the single depot VRPs into one MDVRP. We evaluate the final solution by summing the objective functions of the single depot problems. For testing purposes, we propose to include the removal and repair operators that are mentioned in Table 4.2.

4.4.1 Transport Problem

According to Tansini et al. (2002) the transport problem aims to identify how many items are supplied to a customer from a certain depot. We have seen in Chapter 3 that a customer can only be supplied from a single depot. Inspired by Tansini et al. (2002) we describe a mathematical model of the transport problem. The objective function in Equation 4.1 shows that we aim to minimize the total costs of distributing goods from depots to customers. Here $x_{i,j}$ is a binary variable that is equal to 1 if customer i is assigned to depot j. $c_{i,j}$ denote the costs of delivering customer i from depot j. Since it is not straightforward to assign delivery costs for this problem, we take $c_{i,j}$ as the distance between customer i and depot j. Constraint 4.2 and 4.4 together ensure that every order is assigned to exactly one depot. Constraint 4.3 ensures that the capacity constraint of the depot is not violated. The total quantity of goods that are delivered to customers from a certain depot may not be larger than the capacity at the depot. Here d_i denotes the demand of order i. q_j denotes the total capacity of depot j.

$$\min \sum_i \sum_j x_{i,j} \cdot c_{i,j} \tag{4.1}$$

$$\sum_j x_{ij} = 1, \forall i \tag{4.2}$$

$$\sum_i x_{ij} \cdot d_i \leq q_j, \forall j \tag{4.3}$$

$$x_{ij} \in \{0,1\} \tag{4.4}$$

4.5 Hybrid Approach

The fifth ALNS based strategy, is a hybrid form between the previous four strategies. This hybrid strategy is based on the work of Salhi et al. (2014). Instead of fixing the depot assignment for all customers, we only fix the depot assignment for so-called non-borderline customers. The notations needed to illustrate this idea, which follow from Salhi et al. (2014), are mentioned below.

I: Set of all customers, where $I = \{1, 2, ..., n\}$

D: Distance

p_i: Nearest depot to customer i

$p_i^{'}$: Second nearest depot to customer i

B: the set of borderline customers

η: Parameter that is used to distinguish between borderline customers and non borderline customers. $(0 < \eta < 1)$

To determine which customers belong in set B, we take the following steps.

1. For each customer i, find p_i and $p_i^{'}$. Set $B = \emptyset$.
2. For each customer i, compute $\rho = \frac{D_{ip_i}}{D_{ip_i^{'}}}$. If $\rho \leq \eta$, allocate customer i to its nearest depot. If $\rho > \eta$, assign customer i to set B.

After following these steps for all customers, all $i \notin B$ are assigned to their nearest depot. For the $i \in B$, no depot assignment is fixed yet. These are the so-called border line customers. Following the work of Salhi et al. (2014) we assign these border line customers to the nearest and second nearest depot. This means that in the ALNS phase a task can be removed from a route on its nearest depot, and placed in a route that belongs to the second nearest depot during the repair phase. Using the aforementioned order to depot assignment, we run the ORTEC construction method. This provides us with an initial solution. Afterwards the basic ALNS strategy is applied to this solution. We refer to Algorithm 1 for a description of the basic ALNS strategy.

4.6 Conclusion

In this chapter, we described 5 different strategies for applying ALNS on the VRP of the retail client. This description provides us with enough insight to answer research question 3a, *Which solution strategies can be defined to provide a solution to the VRP of the retail client?* This chapter describes 5 different solution strategies. We summarize these strategies below in Table 4.3. Each of these strategies uses a different amount of freedom in the assignment of tasks to depots. Including these strategies in our tests should provide us with enough insight in the influence of dynamic sourcing on ALNS. This tackles the most difficult aspect of our case. Moreover, by including new ALNS operators, we believe that the proposed strategies are able to tackle all other aspects of the VRP of the retail client sufficiently.

Table 4.3: Summary of Solution Strategies

#	Strategy	Summary
1	Basic ALNS	This strategy uses current techniques and practices of ORTEC.
2	Improved Basic ALNS	This strategy extends the Basic ALNS strategy by including new repair operators.
3	ALNS with fixed depot assignment	This strategy uses the current solution algorithm to determine depot assignment. ALNS is then applied on each depot individually while using pre-determined routes.
4	ALNS-FDATTP	This strategy solves the transport problem to determine the depot assignment. Per depot the current construction and local search finds an initial solution. Next, ALNS is applied for each depot
5	Hybrid ALNS strategy	The hybrid approach uses the approach of Salhi et al. (2014) to determine the depot assignment. Afterwards ALNS is applied per depot and for the complete case iteratively

Chapter 5

Computational Experiments

In the previous section, we defined five different solution strategies. As explained in Chapter 1, the aim of this thesis is to apply ALNS in such a way that it produces good solutions for the VRP cases of the retail client. This chapter explains how we arrive at the best possible configuration for each of the solution strategies that are defined in Chapter 4. Section 5.1 explains the test approach that we employ in this thesis. Afterwards, Section 5.2 describes the parameter tuning process that we follow to define the settings for each of the solution strategies. Section 5.3 analyses the operators that ALNS uses. Finally, Section 5.4 provides an analysis of the number of iterations needed in the various ALNS strategies. We devote a separate chapter to the comparison of the solution strategy. This allows us to better evaluate and explain the results.

5.1 Test Approach

In this section, we describe the test approach that we use in this research. The process that we follow to test the 5 solution strategies can be divided into 2 steps. The first step is to determine the parameter settings for each of the strategies. Section 5.2 explains in more detail how we determine the parameter values for each of the strategies. The second step of the test approach is to compare the performance of the different solution strategies. We use the outcomes of the parameter tuning step to evaluate the performance of the solution strategies.

5.1.1 Test Data

In total we have access to 20 VRP cases of the retail client. Chapter 3 introduces these cases, but for completeness we list them below in Table 5.1. For our test approach, we divide these 20 cases in two sets. The parameter tuning part uses the first set, consisting of 4 cases. This set consists of cases 5,7,10 and 18. These cases are selected at random. We are aware that it is desirable to use more cases to perform the parameter tuning approach. The parameter tuning process requires running many configurations. Since each test run takes between 15-30 minutes, we choose to perform the parameter tuning on only 4 cases. Moreover, by choosing a limited amount of cases to perform the parameter tuning, we prevent over fitting of the solution strategies to the cases. To test the ALNS strategies, we use the algorithms that are available in CVRS. These algorithms are programmed in

$C++$. The tests are performed on a Windows 10 PC with eight-core Intel $I7$ processor and $8gb$ of RAM.

Table 5.1: Overview of 20 test cases

[redacted]

5.1.2 Incorporating Depot Assignment Strategies

Three of the solution strategies, which we propose in Chapter 4, cannot be implemented directly in CVRS. As a result of that, we design our own tools that allow us to test the proposed strategies. First of all, we design a tool that allows us to solve the transport problem before starting an optimization run in CVRS. To understand the data-structure of CVRS, we refer to Section 3.1. From this it can be seen that CVRS uses a number of entities as input. We see that an order has an attribute that is called "allowed depots". For the retail case, this attribute takes on the value of all depots in the problem set. To translate the outcome of the transport problem to CVRS, we manipulate this attribute. We choose to model the transport problem in Excel since it contains good solvers that can quickly find good solutions for the transport problem. More importantly, Excel is a tool that is readily available in many companies. Moreover, the outcomes can easily be analysed afterwards. To import and transform all the data in order to solve the transport problem, we make use of VBA. After running the application, the orders have only 1 allowed depot from which they can be delivered. This information is used by CVRS to find a solution to the VRP. We use the opensolver of Excel with CBC engine by Lougee-Heimer (2005), to solve the transport problem. We refer to Appendix G for an explanation of the code that is used to solve the transport problem and the code used for changing the depot assignment in the input files of CVRS.

5.2 Parameter Tuning

This section describes how to determine the parameter settings for each of the solution strategies. First, Subsection 5.2.1 introduces the parameter tuning process that is followed. Next, Subsections 5.2.2 to 5.2.6 show the results of the parameter tuning process for each solution strategy.

5.2.1 Parameter Tuning Process

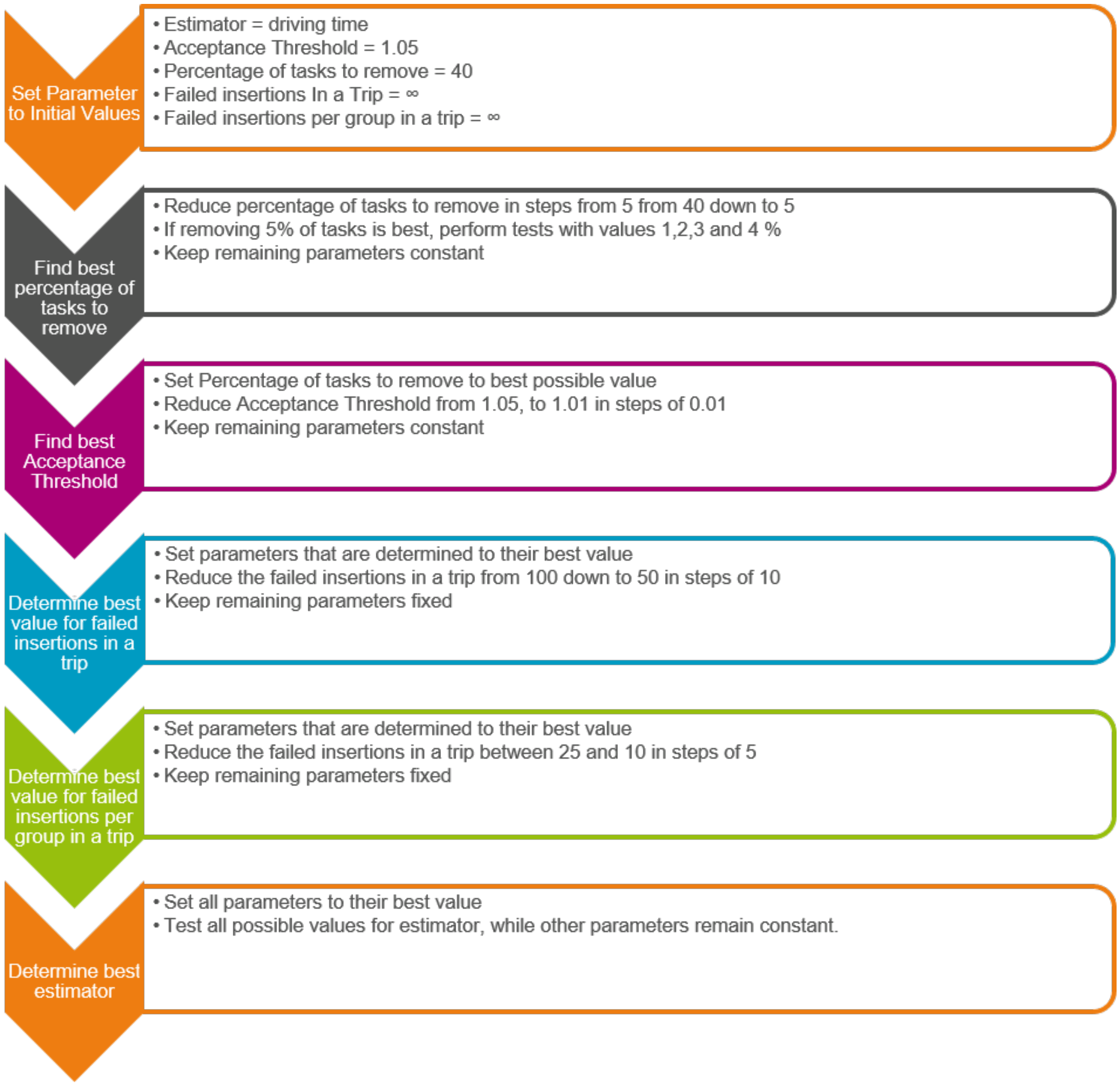
In Section 3.1.3, we mention all parameters that can be configured for the ALNS algorithms. However, not all settings are relevant for the VRP of the retail client.Table 5.2 presents the relevant parameters.

Table 5.2: The different settings, which are determined for all ALNS strategies, shown with range of possible values.

Setting	Possible values	Step-size
% of tasks to remove	$1, 2, 3, 4, 5, 10, 15, ..40$	
Acceptance Threshold	1.01-1.05	0.01
Estimator	“*Costs*”, “*drivingtime*”, “*Waittimeandcosts*”	
Maximum Number of Failed Insertions in a trip	50-100	10
Maximum Number of Failed Insertions per group in a trip	10-25	5

Figure 5.1, illustrates the process that we follow to determine the optimal values for the parameters in Table 5.2. To tune the settings, we first set each of them to the highest possible value. This allows the strategies to potentially explore the largest neighbourhoods possible. Moreover, since we assume no interaction effects exist the estimator can be set to a random value from the set. As a result, we expect ALNS to find very good solutions, but at the cost of a large computation time. Next we reduce each of the parameters one by one. Figure 5.1 better illustrates this process. We first reduce the *% of tasks to remove*. This parameter is likely to have the largest impact on the solution quality. The original work of Pisinger and Ropke (2007) shows that it is best to remove 40% of tasks. Simons (2017) shows that it is more desirable to remove 15% of tasks for large cases. As a result, we decrease the *% of tasks to remove* in steps of 5% from 40%, down to 5%. This gives an indication for interesting area to explore further. With this setting, we investigate promising values for *acceptance threshold*. Next, we reduce the value for *Maximum number of failed insertions per trip*. After fixing this setting to the best performing value, we reduce the *Maximum number of failed insertions per group in a trip*. Lastly, we fix all previously determined parameters and test different values for the *estimator*. We repeat this process for each of the five solution strategies. Subsections 5.2.2 to 5.2.6 contain the results of the parameter tuning process for each of the five solution strategies. Ideally, we would also investigate interaction effects between the various parameters. However, running each configuration takes on average 20 minutes (including construction). As a result, investigating interaction effects is computationally too intensive for this research.

Figure 5.1: Illustration of the process that is followed to determine the parameter values



5.2.2 Parameters Basic ALNS strategy

In this section, we determine the parameter settings for the Basic ALNS strategy. Following the parameter tuning approach, which we describe in Section 5.2, we first determine the *% of tasks to remove*. To determine this, we test values of 5% to 40%, in steps of 5%. The remaining settings are configured as explained in Figure 5.1.

Figure 5.2 shows the relationship between the average improvement in costs, the computation time and the % of tasks removed from a solution. We observe that only an improved solution is found if 5% of all tasks in the solution is removed or less. Destroying a larger part of the solution does only lead to an increase in computation time. We conclude that it is desirable to remove 2% of tasks in the Basic ALNS strategy. Before proceeding to determine the values for the remaining parameters, we fix the % of tasks to remove at 2% for all upcoming tests with this strategy. This is in line with the strategy that we describe in Section 5.2. Comparing this with the findings of Simons (2017), then we find contradictory results. Simons (2017) indicates that ALNS performs best if approximately 15% of tasks is removed. Although the percentage of the solution to be destroyed is smaller, the number of tasks is similar. This is due to the smaller case sizes used by Simons (2017)

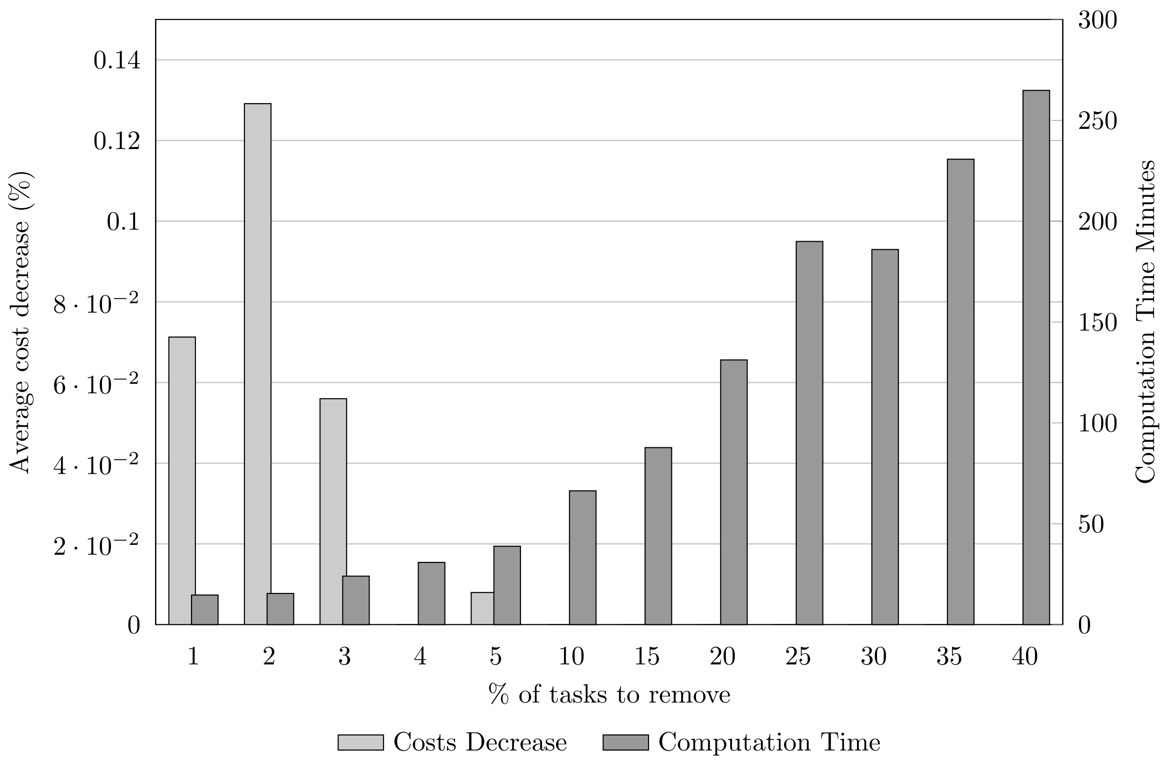


Figure 5.2: Average costs decrease and computation time shown for different values of % of tasks to remove.

At this point, the percentage of tasks to remove is determined. Next we perform tests to find the best value for *Acceptance Threshold*. Figure 5.3 illustrates the results of these tests. We see that the solution quality is not affected by the range of values that we test. Moreover, the computation time increases if we increase the value of the threshold. This is explained by the increased number of times that local search is tried. We conclude that it is possible to find the same solutions in less computation time, if we fix the Acceptance Threshold to 1.01.

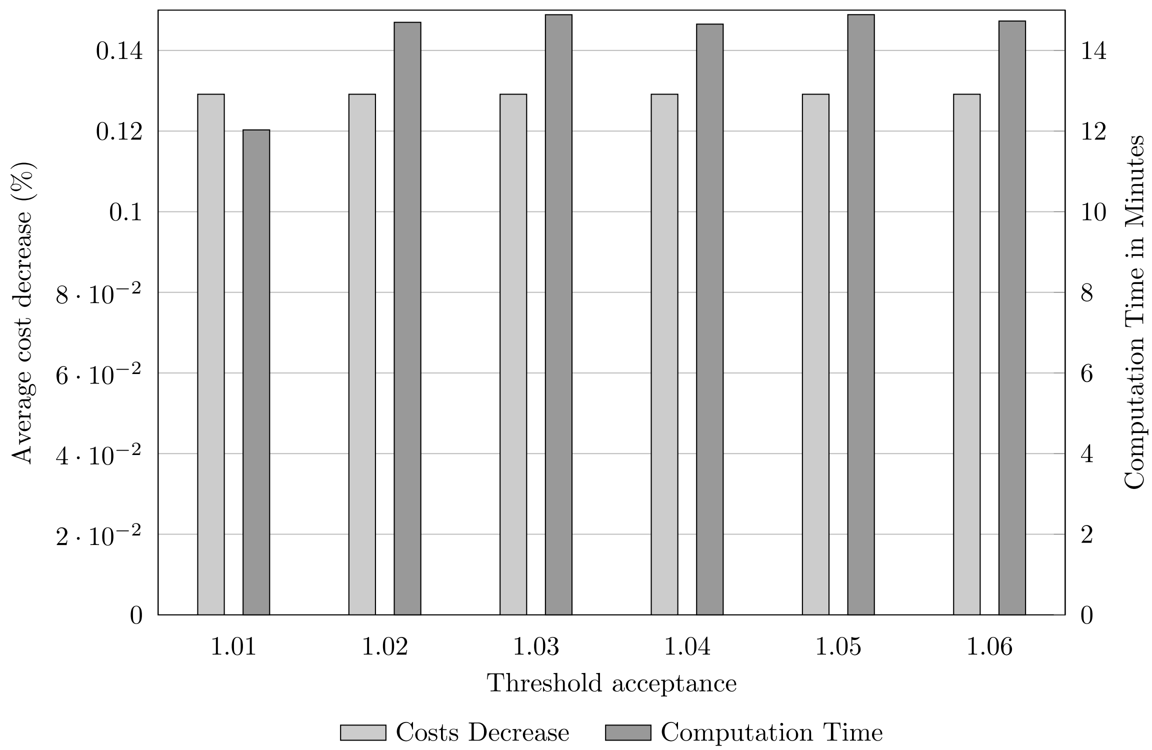


Figure 5.3: Average cost decrease and computation time shown for different values of acceptance threshold

Next, we fix the *% of tasks to remove* at 2% and the *Acceptance Threshold* at 1.01. With these values, we determine the *Failed insertions in a trip* and following the *failed insertions per group in a trip*. Figure 5.4 and Figure 5.5 show the results of these tests. First we change the number of failed insertions in a trip. Figure 5.4 shows that reducing the maximum number of failed insertions in a trip below 60 leads to a decrease in the quality of the solution. Furthermore, the setting does not affect the calculation time too much. For the maximum number of failed insertions per group in a trip, we see that the solution quality and computation time remains constant, for the different values that we tested.

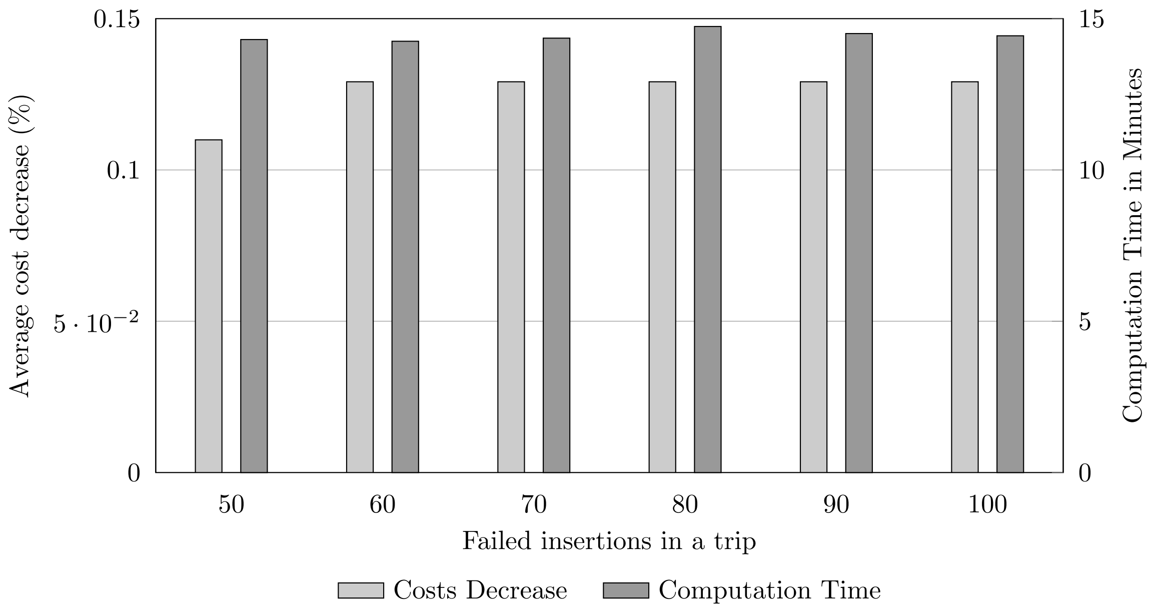


Figure 5.4: Cost decrease and computation time for different values of failed insertions in a trip

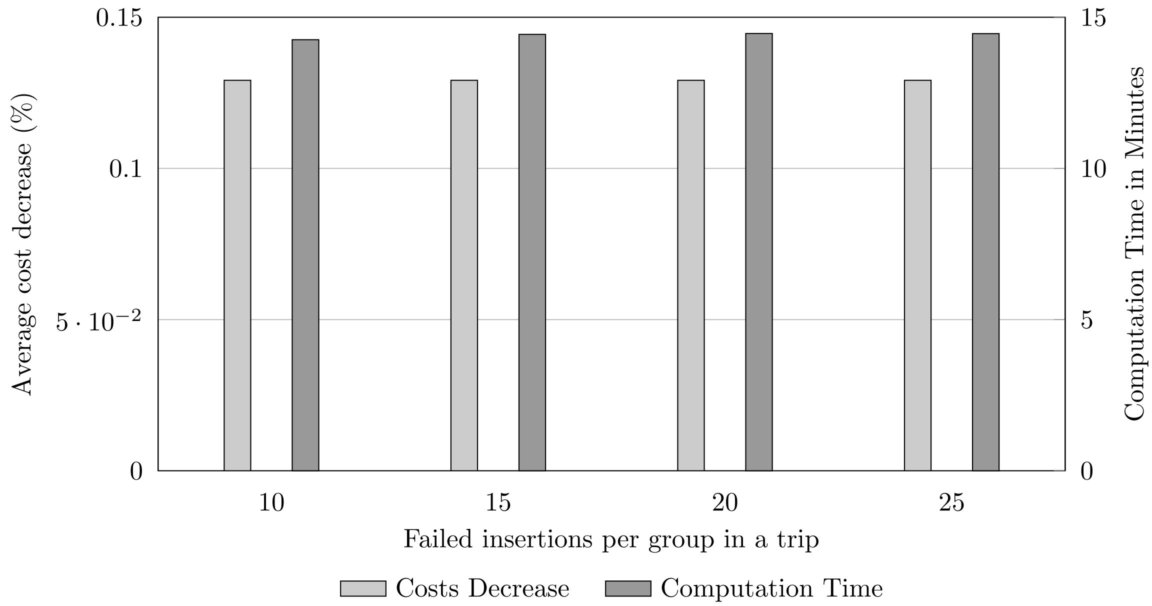


Figure 5.5: Cost decrease and computation time for different values of Failed Insertions per group in a trip

Lastly, we decide which *estimator* needs to be used. After fixing the aforementioned parameters to their best possible values, we compare the three different estimators. Based on the results, which are shown in Figure 5.6, we conclude that *driving time* is the best estimator to use. The best performing configuration for the basic ALNS strategy is shown in Table 5.3.

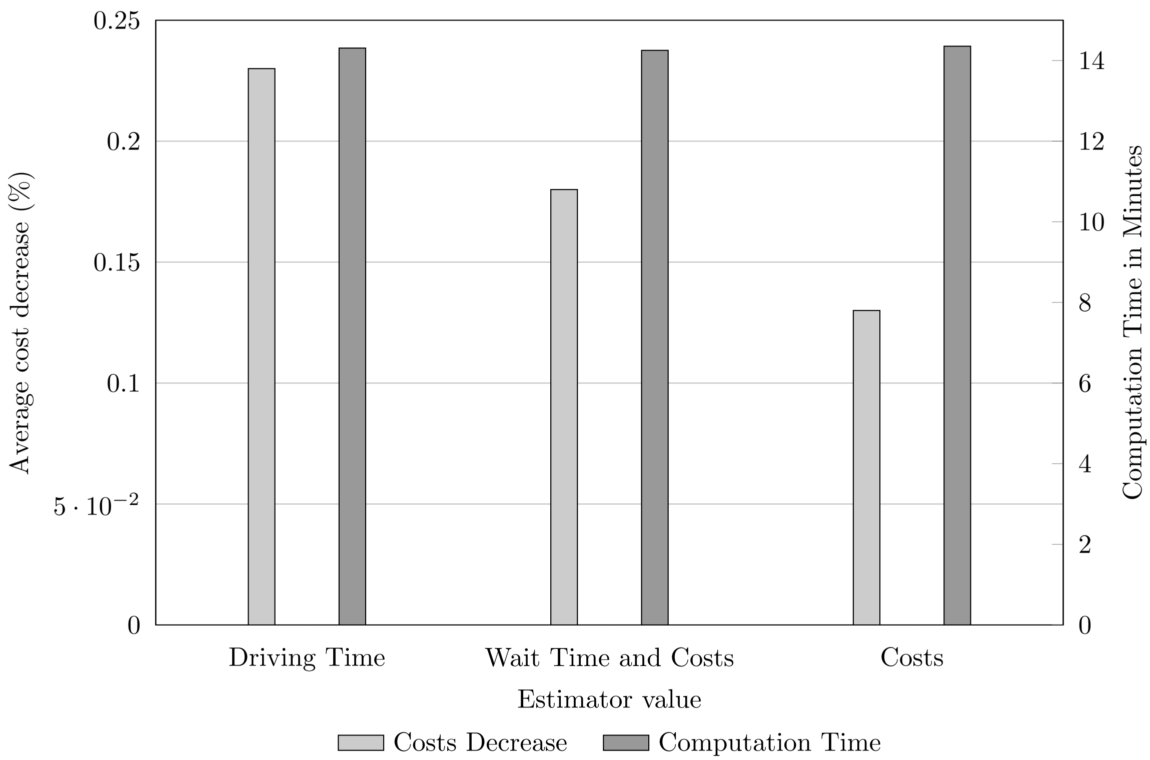


Figure 5.6: Average cost decrease and computation time shown for different values of acceptance threshold

Table 5.3: Best performing configuration for the basic ALNS strategy

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in minutes)
2	DrivingTime	1.01	60	10	0.13	14.21

5.2.3 Parameter Tuning Extended Basic ALNS Strategy

In this section we describe how we derive the parameter values for the Extended Basic ALNS strategy. This strategy includes an additional route selector. We follow the same approach to determine the values for the parameters as we did for the basic ALNS strategy. Based on the outcomes in the previous section, we conclude that removing more than 5% of tasks increases the computation time largely, while the solution is not improved much. Since we include a more powerful recreate method, we believe that it is possible to remove a larger part of the solution. As a result, we perform tests with *% of tasks to remove* between 1 and 7(step-size of 1). Based on the average results, we conclude that it remains desirable to remove 1% of tasks from the solution. Following the same approach as for the Basic ALNS strategy, we tune the remaining parameters. We refer to Appendix H for all the results of the parameter tuning process. Table 5.4 shows the best performing configuration.

Table 5.4: Best performing configuration for the extended ALNS strategy

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in minutes)
1	DrivingTime	1.01	50	15	0.23	9.34

5.2.4 Parameter Tuning ALNS with fixed depot assignment

This section describes how we tune the parameters for the extended basic ALNS strategy. We follow the same parameter tuning approach that we explained in Section 5.2.1. As a result, we first tune the % of tasks to remove from a solution. Appendix H shows the outcomes of these tests. The results show that the best solution is found if 4% of tasks is removed when using the driving time estimator. This is 4% of the tasks that are included in a single depot problem. Compared with the previous strategies, we must remove a larger % of the tasks from the solution. This is due to the fact that the cases are smaller when using this third strategy. Using the best values for the aforementioned parameters, we determine the acceptance threshold. Based on the results in Appendix H, we conclude that the best results are achieved for an acceptance threshold of 1.01. Accepting a lower number of worse solutions, does not result in a worse final solution, but the computation time decreases. Lastly, we can reduce the computation time of the algorithm together with a small improvement in the costs by setting values for *number of failed insertions in a trip* and *number of failed insertions per group in a trip*. Appendix H contains the results of all these tests. Table 5.5 shows the best performing strategy.

Table 5.5: Test results to determine the % of tasks to remove for the ALNS strategy with fixed depot assignment

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in minutes)
4	DrivingTime	1.01	50	5	0.29	8.75

5.2.5 Parameter Tuning Fixed depot assignment through Transportation Problem

This section describes how we determine the parameters for the fourth strategy. To find good parameter settings, we use the OFAT approach. Based on the experience from the parameter tuning process of the aforementioned 3 strategies, we believe that removing large number of tasks from a solution does not lead to improvements. As a result, we only test with values for % of tasks to remove that is less than 5%. Appendix H shows the results of all other tests that we conducted to determine the parameter settings. We see that the best results are achieved when 5% of tasks is removed. This gives us reason to believe that removing larger parts of the solution can be beneficial. After performing extra tests, we conclude that destroying 6% of a solution in each iteration gives the best results. Appendix H contains the results of all tests. The best performing configuration is shown in Table 5.6 This configuration improves the solution the most, within the least amount of time. In the calculation of the computation time we neglect the time needed for solving the transport problem. Additional tests show that the transport problem can be solved using in less than 1 second for some of the largest cases. This is so small compared to the allowed computation time of 35 minutes that we neglect it here. Since the depot assignment is strict in this strategy, on average only 93.5% of the tasks are planned. It is likely that the time-windows of the orders are too strict and as a result it is not possible to find a complete schedule. As a result, for the test cases, this strategy is not guaranteed to find a feasible solution.

Table 5.6: Test results to determine the % of tasks to remove for the ALNS strategy with fixed depot assignment

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in minutes)
6	Costs	1.04	80	15	0.37	10.38

5.2.6 Parameter Tuning hybrid ALNS strategy

This section describes the parameter tuning of the hybrid ALNS strategy. Appendix H shows the results of the tests that we conduct to determine the parameters for this strategy. Based on the outcomes of these tests, we conclude that the best solutions are achieved when 1% of the solution is destroyed. Next, the threshold of accepting worse solutions should be set to 1.03. We conclude that lowering this threshold further increases the computation time, while the total costs of the solution do not change. Next, we further reduce the computation time by reducing the maximum number of iterations in a trip and the maximum number of iterations per group in a trip. We refer to Appendix H for the outcomes of the remaining tests. Table 5.7 shows the best performing strategy.

Table 5.7: Test results to determine the % of tasks to remove for the ALNS strategy with fixed depot assignment

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in minutes)
1	DrivingTime	1.03	90	15	0.57	12.58

5.3 Analysis of ALNS Components

In this section, we analyse the different components of the strategies. We aim to identify which combination of operators work well in combination with what solution strategy. In this section we dive into the performance of the methods that ALNS uses to destroy and repair a solution. We focus on identifying which methods are successful and which are less successful. Moreover, we try to explain why certain methods are effective and others not. Figure 5.7 and Figure 5.8 together with Appendix I show the performance of the ALNS methods for each strategy for the 4 parameter tuning cases. Based on these results, we conclude that some methods contribute more to improving the solution than others. In general we see that the removal methods which remove tasks according to some relationship function, for example cluster removal, related removal or trip removal, are generally successful. Moreover, we conclude that for the basic strategy and extended basic strategy, worst removal and random removal are not effective. We argue that this is caused by the size of the cases. If two tasks are removed from the solution at random, it is likely that these tasks are located far away from each other. As a result, not much improvement in the solution can be achieved during the repair phase. As a result, it may be more effective to destroy a localized part of the solution for large sized cases. We base these conclusions on Figure 5.7 and Figure I.

Appendix I concludes that related removal is by far responsible for the largest amount of improvements to the solution. However, the improvements that are found are generally relatively small. As a result, the total decrease in transport costs is similar to the random removal operator. It is reasonable to assume that if a localized part of the solution is destroyed, the possible improvement found is also small. The figures in Appendix I show the performance of the ALNS methods for the hybrid strategy. We find that the most improvement in transport costs is found by the trip removal method. Trip removal methods also find these improvements in less iterations. Thus, for the hybrid ALNS strategy the trip removal method is most effective. We argue that this method removes multiple trips from different depots. If this happens, the depot assignment of the borderline customer can change. This results in better solutions that are not found with the other operators. In this strategy, small improvements are found by random removal methods. This method accounts for 21% of total number of improvements, but only improves the solution with 2%. We argue that the large size of the cases is the cause of the ineffectiveness of random removal. Randomly removing tasks from the solution that lay in different parts of the distribution network, limits the number of feasible repair options. Resulting, randomly removing tasks is less effective. With concerns to the repair operators, we see that a large part of the improvements is found by the regret insertion methods. This method is especially successful in combination with related removal, cluster removal and trip removal. We argue that this observation makes sense. The aforementioned destroy methods remove tasks from a solution that are usually located closely together. During the repair phase, the insertion options of these tasks are often overlapping. Thus the regret factor is of influence on the solution quality. If tasks are removed in a random fashion, these tasks are often located far away from each other. This means that the repair options for these tasks are usually non-overlapping. As a result, the order in which the tasks are inserted is not so important. Hence, regret insertion is not necessarily better in combination with aforementioned removal method.

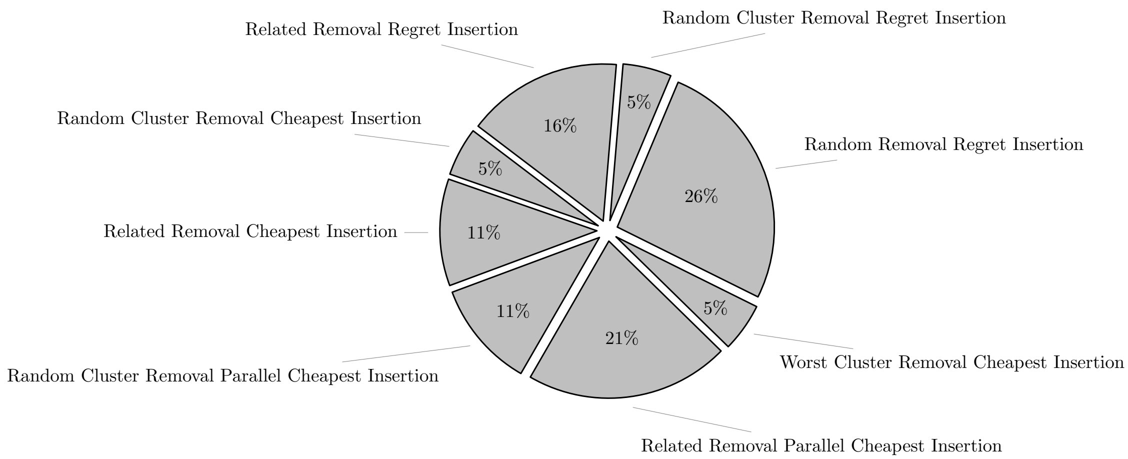


Figure 5.7: Analysis of effectiveness of destroy and recreate methods for the standard ALNS strategy. Expressed in the % of times that a method improved the solution as part of total number of improvements.

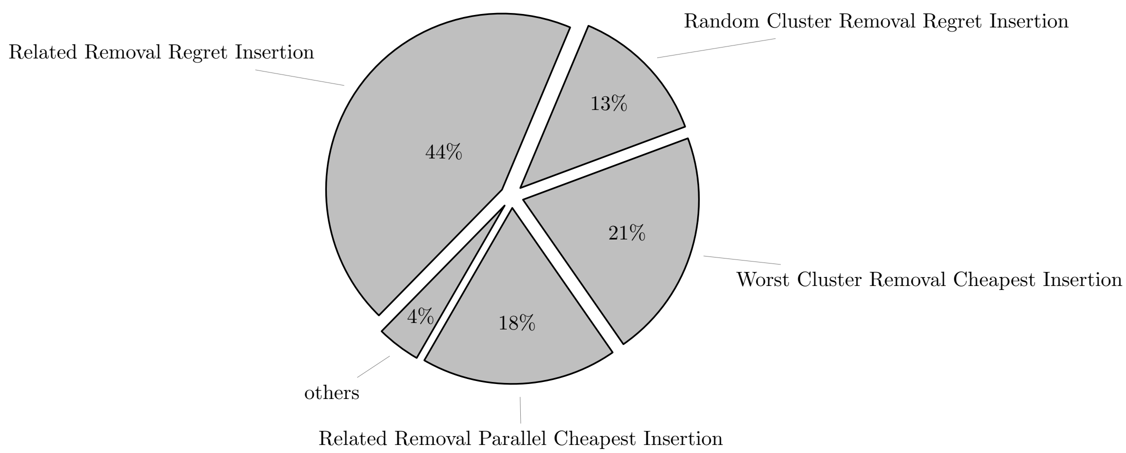


Figure 5.8: Analysis of effectiveness of destroy and recreate methods. Measured in % of costs decrease as part of total costs decrease found by ALNS.

5.4 Analysis of the number of iterations needed for ALNS

In this section we analyse the total number of iterations for each of the proposed solution strategies. Figure 5.9 and Figure 5.10 show the relationship between the number of iterations, the Computation Time and the costs decrease. Figure 5.10 shows that after 250 iterations, ALNS finds hardly any improvements. We see that the computation time for 3 of the solution strategies is approximately 5 minutes. For most of the cases, this will be feasible. For the extended basic ALNS strategy and the hybrid ALNS strategy, the computation time exceeds 20 minutes. Since for a large part of the cases the construction heuristic already takes close to 35 minutes, it is not possible to run so many iterations within the available computation time. To overcome the problem, we advice ORTEC to do the following. We set the number of iterations to 400. This far exceeds the number of iterations that is usually used by ORTEC. Although for the test cases, no more than 250 iterations are needed for each strategy, future unknown cases may require more iterations. As long as the maximum computation time is not exceeded, this is allowed. We present the results of tests with this setting in Chapter 6. For practical implications, we advice ORTEC to use a setting that stops the ALNS algorithms as soon as the maximum computation time is reached.

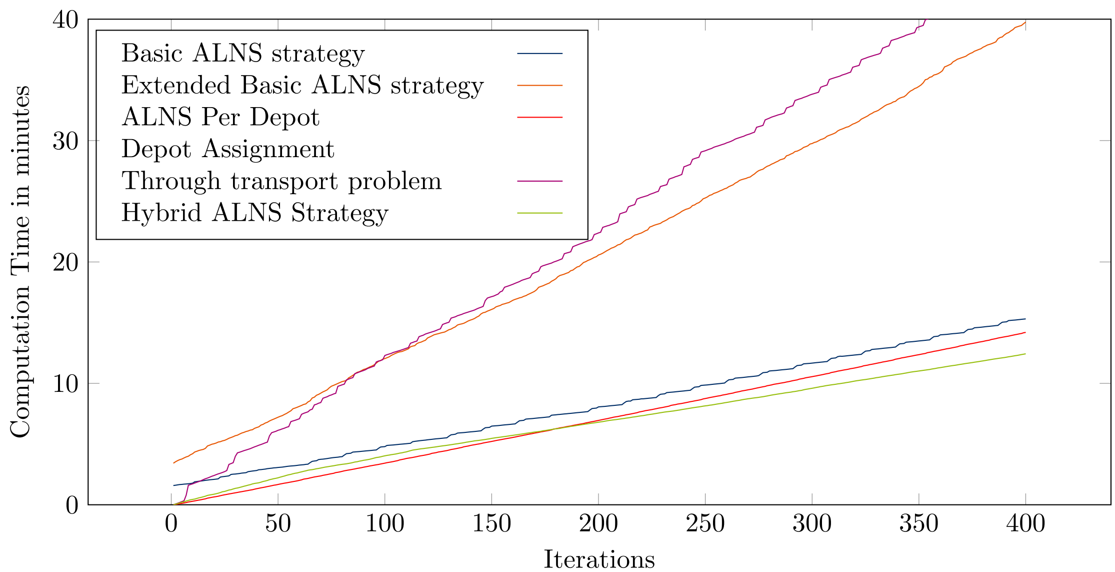


Figure 5.9: Development of the average computation time for the number of iterations

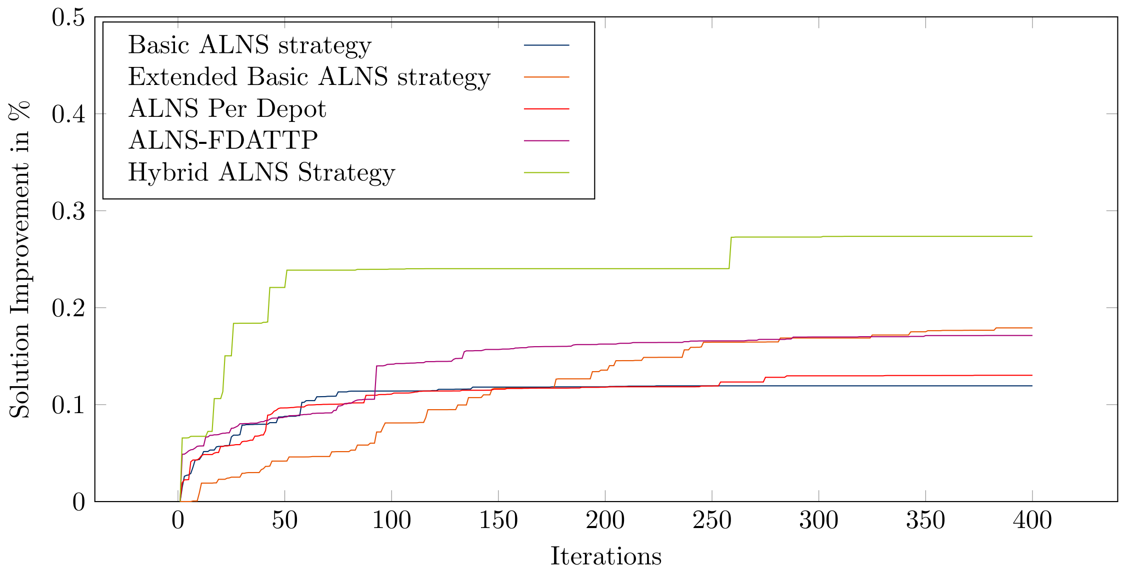


Figure 5.10: Development of the average transport costs for the number of iterations

5.5 Conclusion

In this section, we presented the results of the computational experiments for each of the five solution strategies. Section 5.1 showed the test approach described how we incorporate the depot assignment strategies in CVRS. The test approach allows us to compare the performance of the different solution strategies.

To compare the solution strategies, we must determine the best possible configuration for each of the solution strategies. Section 5.2 describes which steps are followed to determine the best performing configuration for each solution strategy. To prevent over fitting of the strategies on the individual cases, we separated the test cases in two groups. For each of the strategies, we found their best performing configuration. Table 7.1 shos the best performing configuration for each strategy.

Table 5.8: Overview of best performing configuraitons for each solution strategies

Strategy	% of tasks to remove	Estimator	Threshold Acceptance	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip
Basic ALNS	2	DrivingTime	1.01	60	10
Extended Basic ALNS	1	DrivingTime	1.00	50	15
ALNS With Fixed Depot Assignment	4	DrivingTime	1.01	50	5
ALNS-FDATTP	6	Costs	1.04	80	15
Hybrid ALNS	1	DrivingTime	1.03	90	15

Furthermore, Section 5.3 analysed the performance of the different ALNS components. We found that the effectiveness of the ALNS methods is different for each of the solution strategies. For the large cases, the ALNS methods that destroy tasks which are located close to each other are more effective.

Finally, Section 5.4 provides a description of the number of iterations needed for each of the solution strategies. With the experiments that are conducted in this section, we found for each of the solution strategies the best performing configuration. The configurations that are found in this chapter are used in Chapter 6 to compare the performance of the different solution strategies.

Chapter 6

Strategy Comparison

With the information of the previous chapter, we know the best performing configuration for each of the solution strategies. This chapter compares the outcomes of the different solution strategies. Section 6.1 discusses the outcomes of the solution strategies on the 16 cases of the retail client. Section 6.2 discusses for each solution strategy how the results are substantiated and how our findings relate to existing research. We identify the influence of the computation time restriction to the effectiveness of ALNS. Section 6.3 provides this discussion. Lastly, Section 6.4 discusses what the influence of the initial solution is on the effectiveness of ALNS.

6.1 Results Compared

This section describes the performance comparison of the different ALNS strategies. Table 6.1 shows the performance of the five solution strategies on the 16 cases of the retail client. For each strategy the computation time and improvement in costs is shown. Equation 6.1 shows how the costs improvement is calculated.

$$Costs\quad Improvement = \frac{Costs_{afterconstruction} - Costs_{new}}{Costs_{afterconstruction}} * 100\% \tag{6.1}$$

Ranking the solution strategies based only on costs improvement is not fair. Equation 6.1 shows that the cost improvement is a difference between the solution after construction and the final solution. The hybrid ALNS strategy and the ALNS-FDATTP strategy have a different initial solution than the remaining 3 solution strategies. Due to stricter depot assignment strategies, the number of planned tasks is much lower for the hybrid ALNS and the ALNS-FDATTP. Table 6.2 shows how many tasks are planned for the aforementioned solution strategies. We see that the ALNS-FDATTP strategy plans on average 93.5% of the tasks. The Hybrid ALNS strategy plans on average only 81.9% of the tasks. We conclude that this is the cause for the higher efficiency of the ALNS strategies. If less tasks are planned, vehicles have a lower occupation. This allows ALNS to find changes to the solution that cannot be found when more tasks are planned. The remaining 3 solution strategies all use the same initial solution. As a result of this, the improvements, which are found by ALNS, are measured in similar ways. We conclude that the Basic ALNS strategy is on average the best performing strategy, followed by the ALNS strategy with fixed depot assignment and the extended Basic ALNS strategy.

ORTEC OPTIMIZE YOUR WORLD

Table 6.1: Average Costs decrease(%) and computation time(minutes) for the 5 solution strategies over the remaining 16 cases.

	Solution Strategy									
	Basic ALNS		Extended Basic ALNS		ALNS With Fixed Depot Assignment		ALNS with Fixed Depot Assignment Through TP		Hybrid ALNS	
Case	Costs Decrease	Computation Time	Costs Decrease	Computation Time	Costs Decrease	Computation Time	Costs Decrease	Computation Time	Costs Decrease	Computation Time
1	0,00	15,23	0,00	675,25	0,11	19,86	0,00	38,51	0,00	9,22
2	0,00	6,88	0,01	282,09	0,13	8,37	0,06	9,34	0,01	6,09
3	0,00	16,92	0,00	1049,13	0,30	21,14	0,06	18,88	0,85	11,25
4	0,00	16,06	0,01	602,07	0,01	14,87	0,14	17,59	0,00	9,57
6	0,12	24,01	0,08	256,37	0,02	27,73	0,00	64,78	0,07	22,46
8	0,01	22,64	0,01	294,95	0,00	30,71	0,00	35,16	0,00	21,51
9	0,04	17,94	0,00	115,51	0,01	27,59	0,01	28,05	0,00	15,51
10	0,00	23,93	0,00	526,93	0,02	27,50	0,01	46,02	0,00	23,58
12	0,46	18,39	0,54	66,10	0,09	5,25	0,00	14,13	0,79	3,93
13	0,00	11,82	0,39	902,15	0,01	23,16	0,18	23,50	1,10	18,34
14	0,99	16,27	0,56	301,07	0,04	15,90	0,77	16,17	1,20	9,08
15	0,02	13,31	0,04	794,41	0,05	21,17	0,58	17,84	0,00	11,97
16	0,99	11,72	0,01	199,34	0,49	26,09	0,12	23,22	0,01	11,64
17	0,85	19,55	0,00	363,35	0,03	10,86	0,02	26,28	0,77	6,58
19	0,00	14,15	0,00	281,32	0,48	12,82	0,42	20,28	0,33	10,39
20	0,01	25,06	0,00	397,97	0,04	18,88	0,04	19,64	0,00	7,83
Average	0,22	16,95	0,10	444,25	0,11	19,49	0,15	26,21	0,32	12,44
Variance	0,14		0,04		0,03		0,05		0,20	

Table 6.2: Percentage of tasks planned for the Hybrid ALNS strategy and ALNS-FDATTP strategy

Case	ALNS with Fixed Depot Assignment Through TP	Hybrid ALNS
1	95.12	87.1
2	99.3	93.8
3	88.0	58.3
4	97.2	70.0
6	96.5	95.5
8	94.0	97.1
9	98.4	97.2
10	94.6	96.2
12	95.3	87.6
13	79.5	75.3
14	97.7	81.8
15	78.3	65.3
16	90.8	81.4
17	99.0	62.1
19	100	78.3
20	97.6	83.9
Average	93.5	81.9
Variance	49.7	1.7

6.2 Results Explained per Strategy

We are now familiar with the average performance of each of the solution strategies. In this section we elaborate in detail on the performance of each strategy. We substantiate the results for each solution strategy for the individual cases.

6.2.1 Results explained for the Basic ALNS strategy

Table 6.3 shows the results that are obtained by the basic ALNS strategy. The numbers in Table 6.3 represent reductions in respectively number of trips, total distance, total duration and total costs. A positive number means a reduction, and a negative number means an increase in the respective KPI. The results show that the largest costs reductions are obtained for cases 12, 14, 16 and 17. We immediately observe that these large improvements are found by reduction in the number of trips. The results show for only case 6 a visible cost reduction that is not caused by a reduction of the number of trips. For this case, the total distance and total duration is reduced after applying ALNS. We refer back to Section 3.2.3 where we explain how the costs are substantiated. From here it can be seen that the costs per hour, distance and stop are much smaller compared to the costs per trip. As a result of this, reductions in each of the objectives does not count equally.

Table 6.3: Difference in distance, total duration, number of trips and costs that are realized by the basic ALNS strategy.

Case	Reduction In Trips	Reduction In Distance (in KM.)	Difference In Total Duration (in hours)	Cost Reduction in %
1	0	0.24	0.00	0.00
2	0	0.03	0.00	0.00
3	0	0.00	0.00	0.00
4	0	0.25	-0.16	0.00
6	0	19.47	2.15	0.12
8	0	-1.45	0.11	0.01
9	0	-17.27	0.9	0.04
10	0	0.00	0.00	0.00
12	1	-137.89	-4.45	0.46
13	0	2.03	-0.02	0.00
14	2	-55.18	-2.99	0.99
15	0	45.62	-0.2	0.02
16	2	-97.91	-3.11	0.99
17	1	22.46	-2.06	0.85
19	0	3.54	-0.36	0.00
20	0	4.90	0.16	0.01

We include Table 6.4 to explain why in some cases the total number of trips can be reduced and why in others not. We see that the utilization is the highest for cases 6-10. With the exception of case 6, the basic ALNS strategy hardly improves the solution for these cases. For cases 12, 14, 16 and 17, the utilization of the vehicles is 94.5% and less. Based on this analysis it is expected that the basic ALNS strategy is also effective for cases 1 until 4. However, the structure of these cases is significantly different. We refer to confidential Appendix B for an overview of the distribution network per case. The appendix shows that the distance between depots is much larger for cases 1 until 4, than for cases 12,14,16 and 17. As a result, the basic ALNS strategy attempts to re-assign orders to another depot. If the distance between the depots is large, these changes do not result in improvements.

Table 6.4: Additional characteristic of the cases, which are needed to better understand the effectiveness of the ALNS strategies.

[redacted]

6.2.2 Results Explained for the Extended Basic ALNS strategy

This section explains the results of the extended basic ALNS strategy. This strategy extends the aforementioned strategy with a new route selector. As a result, it is interesting to compare the outcomes of this strategy to the basic ALNS strategy. Table 6.5 shows the reductions in total number of trips, total distance, total duration and total costs. The basic ALNS strategy finds negligible improvements for cases 1 until 4. The improvements found by the extended basic ALNS strategy are a little better. For instance, the reduction in distance for case 4 is 10 times larger with the additional route selector, while the total duration does not become worse. For cases 6 until 10, the additional route selector is not beneficial. Table 6.4 shows that these cases have a high utilization of vehicles. As a result of this it is infeasible to assign orders to the closest depot. Only for case 13, the improvement found by the extra route selector is significantly beneficial. It manages to find a solution with 1 route less. This solution is not found by the Basic ALNS strategy.

Table 6.5: Difference in distance, total duration, number of trips and costs that are realized by the extended basic ALNS strategy.

Case	Reduction In Trips	Reduction In Distance (in KM.)	Difference In Total Duration (in hours)	Cost Reduction in %
1	0	0.71	0.03	0.00
2	0	-1.59	0.14	0.01
3	0	0.00	0.00	0.00
4	0	24.37	0.00	0.01
6	0	-12.48	1.99	0.08
8	0	-2.21	0.21	0.01
9	0	0.00	0.00	0.00
10	0	0.00	0.00	0.00
12	1	-118.17	-3.95	0.54
13	1	12.78	-1.3	0.39
14	1	9.71	-0.27	0.56
15	0	30.31	0.67	0.04
16	0	-16.59	0.27	0.01
17	0	0.00	0.00	0.00
19	0	0.47	0.05	0.00
20	0	0.00	0.00	0.00

6.2.3 Results Explained for the ALNS strategy with fixed depot assignment

Table 6.6 shows the change in number of trips used, change in duration and change in total distance. Corresponding with the outcomes shown in Section 6.2.1, the largest improvements are found when the number of trips can be reduced. This is the case for cases 3, 16 and 19. Unlike the previous 2 strategies, we observe that this strategy finds significant reductions in total distance and total duration. Especially for cases 1 until 4. We saw earlier that the Basic ALNS strategy is unable to find improvements for these cases. This strategy finds reductions in total distance and total duration for cases 1,2 and 4 and finds a reduction in the number of trips for case 3. This is explained by the structure of the cases. Appendix B shows that the distance between depots is much larger for cases 1 until 4. As a result, it is not efficient to change the depot assignment after construction. This results in improvements found by the strategy that forbids any depot assignment changes during the ALNS phase. Also for the remainder of the cases, this strategy is generally better in finding improvements in distance and total duration than the previously 2 discussed strategies.

Table 6.6: Difference in distance, total duration, number of trips and costs that are realized by the ALNS strategy with fixed depot assignment.

Case	Difference In Trips	Difference In Distance	Difference In Total Duration	Cost Reduction in %
1	0	6.92	1.81	0.11
2	0	38.69	1.69	0.13
3	1	-62.33	-3.30	0.30
4	0	-5.19	0.18	0.01
6	0	-1.17	0.36	0.02
8	0	0.16	0.01	0.00
9	0	0.20	0.17	0.01
10	0	9.04	0.49	0.02
12	0	-14.4	1.56	0.09
13	0	3.88	0.3	0.01
14	0	9.60	0.19	0.04
15	0	-5.71	1.26	0.05
16	1	-29.06	-1.55	0.49
17	0	3.14	0.73	0.03
19	1	-19.48	-0.98	0.48
20	0	-9.37	0.98	0.04

6.2.4 Results Explained for the ALNS-FDATTP strategy

This section discusses the results that are obtained by the ALNS-FDATTP strategy. This strategy has a different initial solution compared to the strategies that are discussed so far. This is caused by a stricter depot assignment. As a result of this, it is difficult to compare the results with the aforementioned solution strategies. The results show that the largest improvements are obtained for the cases in which the number of trips is reduced. This is in line with the previous findings. The results also show that large improvements are obtained in terms of distance and total duration. In the previous section, we show that the reductions in distance and total duration are larger when ALNS is applied per depot. The results from this strategy confirm these findings. The magnitude of improvements with this strategy is a little higher. This is caused by the fact that less orders are planned in this strategy.

Table 6.7: Difference in distance, total duration, number of trips and costs that are realized by the ALNS-FDATTP strategy.

Case	Difference In Trips	Difference In Distance	Difference In Total Duration	Cost Reduction in %
1	0	0.00	0.00	0.00
2	0	20.94	0.7	0.06
3	0	59.34	0.53	0.06
4	0	105.71	1.81	0.14
6	0	0.00	0.00	0.00
8	0	1.35	0.01	0.00
9	0	13.27	0.17	0.01
10	0	5.07	0.19	0.01
12	0	0.00	0.00	0.00
13	0	42.31	2.84	0.18
14	1	69.46	2.52	0.77
15	1	20.4	1.25	0.58
16	0	38.55	1.58	0.12
17	0	6.18	-0.07	0.02
19	1	-82.55	-1.59	0.42
20	0	7.12	0.7	0.04

6.2.5 Results Explained for the Hybrid ALNS strategy

This section discusses the outcomes of the hybrid ALNS strategy. Table 6.8 shows the difference in number of trips used, the difference in total distance and the difference in total duration. In line with the previous observations, we see that the largest improvements are found when the number of trips is reduced. The number of planned tasks is different from the aforementioned strategies. As a result of this, it is difficult to compare this strategy with the aforementioned strategies. However, we can say something about the differences in performance on each case. The results show that for cases 4,8,9 and 10 the costs improvement is negligible. We see that cases 8,9 and 10 have both a high utilization and a low number of depots. The low number of depots, together with the small distance between the depots reduce the effectiveness of the hybrid strategy. If the distance between the depots is small, the separation between borderline customers and non-borderline customers is too strict. The hybrid strategy in this case forbids too many changes. As a result of this, the effectiveness of ALNS is reduced.

Table 6.8: Difference in distance, total duration, number of trips and costs that are realized by the hybrid ALNS strategy.

Case	Difference In Trips	Difference In Distance	Difference In Total Duration	Cost Reduction in %
1	0	0.39	0.01	0.00
2	0	0.60	0.26	0.01
3	1	-9.05	-0.32	0.85
4	0	0.00	0.00	0.00
6	0	-2.19	1.4	0.07
8	0	0.12	0.00	0.00
9	0	0.00	0.00	0.00
10	0	0.19	0.00	0.00
12	1	-7.91	-1.77	0.79
13	2	-30.36	-0.92	1.10
14	2	-62.17	-2.29	1.20
15	0	0.00	0.00	0.00
16	0	5.58	0.04	0.01
17	2	19.38	1.28	0.77
19	1	-87.62	-4.63	0.33
20	0	0.00	0.00	0.00

6.3 Results after restricted computation time

Thus far we have ignored the computation time restriction. This way have better shown the potential of the ALNS strategies. This section presents the results if we include the computation time restriction. Table 6.9 shows the results that each strategy obtains when the computation time is limited to 35 minutes. Based on this, we see that the hybrid strategy is on average no longer the best strategy. Moreover, we see that this strategy does not find any improvement within the allowed computation time of 35 minutes. This is due to the fact that the ORTEC construction heuristic in combination with this strategy already takes the full 35 minutes. This is due to a construction in the software of ORTEC. Also the extended basic ALNS strategy no longer finds any improvements within the limited computation time. This is caused by the large computation time needed for the proposed route selector. To make this selector work in the current capabilities of ORTEC, a sorting action is executed. Since the cases are large and sorting orders is a computationally intensive operation, this requires a lot of time. One of the main benefits of the ALNS strategies with fixed depot assignment, is that the construction heuristic can be applied for each depot individually. Each single depot problem is smaller than the combined multi-depot problem. As a result, the time needed for the construction heuristic can decrease for many cases. Although the basic ALNS strategy performs worse when the computation time is restricted to 35 minutes, on average it is still best. The results show that on cases 1-4, the ALNS strategy with fixed depot assignment is best. On cases 6-10, hardly any improvements are found by any of the strategies. Recall that the utilization of these cases were higher than 96%. We recommend to use the basic ALNS strategy for the cases 12-18. Case 19 and 20 can best be optimized using the ALNS strategy with fixed depot assignment.

Table 6.9: Overview of performance of ALNS strategies with restricted computation time

Case	Basic ALNS	Extended Basic ALNS	ALNS With Fixed Depot Assignment	ALNS-FDATTP	Hybrid ALNS
1	0.00	0.00	0.11	0.00	0.00
2	0.00	0.00	0.13	0.06	0.00
3	0.00	0.00	0.30	0.06	0.00
4	0.00	0.00	0.01	0.14	0.00
6	0.00	0.00	0.02	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00
9	0.04	0.00	0.01	0.01	0.00
10	0.00	0.00	0.02	0.01	0.00
12	0.46	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.18	0.00
14	0.99	0.00	0.00	0.77	0.00
15	0.00	0.00	0.00	0.58	0.00
16	0.58	0.00	0.49	0.12	0.00
17	0.85	0.00	0.03	0.02	0.00
19	0.00	0.00	0.48	0.42	0.00
20	0.01	0.00	0.04	0.04	0.00
Average	0.18	0.00	0.10	0.15	0.00
Variance	0.12	0.00	0.03	0.05	0.00

6.4 Impact of initial solution

This section discusses the impact of the initial solution to the effectiveness of ALNS. Simons (2017) indicates that ALNS at ORTEC works better if the initial solution is already very good. To assess if this is valid for the VRP of the retail client, we test the basic ALNS strategy with a different initial solution. Chapter 3 shows that in the ORTEC includes a local search phase in its Construction Heuristic. The tests in this section use the ORTEC construction heuristic, but without the local search phase. As a result, the total costs of the initial solution will be higher.

We only perform this test for the Basic ALNS strategy for the following reason. This Basic ALNS strategy guarantees to find an initial solution for all cases and shows the best average performance. The extended ALNS strategy and the ALNS strategy with fixed depot assignment show on average a worse performance. As a result, we exclude the latter two strategies from these tests. The hybrid ALNS strategy and the ALNS-FDATTP do not find feasible solutions. These strategies require tailored new construction heuristics to find feasible solutions. This is outside the scope of this research. As a result, these strategies are also not included in these tests.

Table 6.10 shows the best performing configuration for the basic ALNS strategy. We follow the procedure that is described in Section 5.2 to determine the best values of the parameters. If we compare the configuration in Table 6.10 with the configuration of Table 5.3, then we conclude that

Table 6.10: Test results to determine the % of tasks to remove for the basic ALNS strategy with worse initial solution

%Tasks To Remove	Estimator	Acceptance Threshold	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip	Costs Decrease (in %)	Computation Time (in Minutes)
15	WaitTime-AndCosts	1.01	50	5	0.26	65

after a worse initial solution, a larger part of the solution must be removed. Table 6.11 contains the results of the basic ALNS strategy after a worse initial solution. The results show that the strategy remains effective on only a subset of all cases. However, we do see that the improvement that is found increases. When we start with a worse initial solution, we see that the average improvement is 0.63%. For a better initial solution, this improvement was only 0.25% on average. However, since we destroy a much larger part of the solution, the computation time is also much larger. We must make clear that, although the improvement compared to the initial solution is larger, the final solution quality is worse. Thus, if a worse initial solution is used, the final solution is also worse.

Table 6.12 shows how the results of the Basic ALNS strategy are obtained after a different initial solution. The results show that not for each case the solution can be improved by ALNS. This is in line with the findings of Simons (2017). She shows that the initial solution is globally so bad, that even ALNS cannot find an improvement. For the cases that do find an improvement, we see that this is caused by differences in distance and total duration. These improvements are much larger than those obtained when the original initial solution is used. This indicates that the solution found by the ORTEC construction heuristic is stuck in a local optimum. Normally, ALNS should be able to escape from such a local optimum. However, due to the unique design of the cost sets and the high utilization of the vehicles ALNS only finds small improvements to the total distance and total duration.

6.5 Conclusion

This chapter compares the performance of the five solution strategies. The results show that there is not a unique solution strategy that is suitable for all cases. In particular, the results show that the basic ALNS strategy is most effective for the cases where the distance between the depots is not too large. If the distance between the various depots is very large, it is most effective to apply the ALNS strategy with fixed depot assignment. We have also shown that the computation time currently is too restrictive for many cases. As a result of this, ALNS does not find improvements for the cases where the construction heuristic already takes 35 minutes. Lastly, we have shown what the influence is from the initial solution on the quality of the final solution. The tests show that if the initial solution is worse, ALNS is more effective. On the contrary, the final solution is better when the initial solution is already very good.

Table 6.11: Results of the Basic ALNS strategy that are achieved after a worse initial solution

Case	Costs Decrease	Computation Time (minutes)
1	0.00	131.46
2	0.89	45.36
3	0.00	140.39
4	0.00	124.71
6	0.00	149.37
8	0.00	291.83
9	0.00	200.12
10	0.00	318.04
12	2.67	52.44
13	0.00	178.00
14	3.97	132.52
15	0.00	300.94
16	0.00	104.58
17	0.26	88.00
19	2.21	60.90
20	0.00	140.93
Average	0.63	
Variance	1.49	

Table 6.12: Difference in distance, total duration, number of trips and costs that are realized by the Basic ALNS strategy after a different initial solution.

Case	Difference In Trips	Difference In Distance	Difference In Total Duration	Cost Reduction in %
1	0	0.00	0.00	0.00
2	1	142.47	3.42	0.89
3	0	0.00	0.00	0.00
4	0	0.00	0.00	0.00
6	0	0.00	0.00	0.00
8	0	0.00	0.00	0.00
9	0	0.00	0.00	0.00
10	0	0.00	0.00	0.00
12	0	1461.18	25.33	2.67
13	0	0.00	0.00	0.00
14	1	2053.52	44.64	3.97
15	0	0.00	0.00	0.00
16	0	0.00	0.00	0.00
17	2	156.33	5.59	0.26
19	2	563.91	28.14	2.21
20	0	0.00	0.00	0.00

Chapter 7

Conclusions and Recommendations

This last chapter presents the most important outcomes of this research. To achieve this, Section 7.1 presents the conclusions of our research. Moreover, Section 7.2 provides recommendations for ORTEC. This chapter concludes by providing suggestions for further research and limitations of the research in Section 7.3.

7.1 Conclusion

The VRP of the retail client contains many characteristics that make it interesting to investigate the application of ALNS. In Chapter 3 we have seen that in this research we face a vehicle routing problem with a heterogeneous fleet, in which each vehicle type has a different cost set. Moreover, we have seen that the delivery time window in which customers are visited is restricted. All these real-life restrictions must be taken into account when finding a solution to the VRP of the retail client. We also saw that the depot assignment of the orders is a planning decision that is not predetermined. Finally, we saw that the computation time is restricted to 35 minutes.

With this information in mind, we conducted a literature review. Based on the outcomes of the literature review, we proposed to test 5 different solution strategies. The basic ALNS strategy tests the current ALNS framework of ORTEC. Emeç et al. (2016) show that including an additional ALNS method is beneficial for multi-depot VRP. Moreover, a partial optimization approach, which is developed by Lalla-Ruiz and Voß (2019), finds good solutions to the multi-depot VRP. We test two different partial optimization approaches. Lastly, we test the hybrid approach of Salhi et al. (2014). Their neighbourhood reduction technique is specifically designed to work well on multi-depot vehicle routing problems.

- Basic ALNS strategy
- Extended Basic ALNS strategy
- ALNS with fixed depot assignment
- ALNS with fixed depot assignment through solving the transport problem
- Hybrid ALNS strategy

Table 7.1: Overview of best performing configuraitons for each solution strategies

Strategy	% of tasks to remove	Estimator	Threshold Acceptance	Failed Insertions In A Trip	Failed Insertions Per Group In A Trip
Basic ALNS	2	DrivingTime	1.01	60	10
Extended Basic ALNS	1	DrivingTime	1.00	50	15
ALNS With Fixed Depot Assignment	4	DrivingTime	1.01	50	5
ALNS-FDATTP	6	Costs	1.04	80	15
Hybrid ALNS	1	DrivingTime	1.03	90	15

To determine the settings for each of the solution strategies, we developed a parameter tuning process. After following this approach, we found the following optimal settings for each of the solution strategies.

In order to compare the performance of the different solution strategies, we designed a test approach. This test approach consists two steps. First, we found the best parameter settings for each of the strategies. Afterwards, by using the best performing settings of each strategy, we produce the results for the cases of the retail client. The performance of each of the solution strategies is summarized in Table 7.2. Based on our research, we conclude that by looking at the average performance the basic ALNS strategy performs best. However, many differences are observed between the different cases. The results in Table 7.2 are obtained within the allowed computation time of 35 minutes.

The case of the retail client, has 3 big aspects whose influence on ALNS has not been studied together before. With the outcomes of our tests, we shed a light on the relationship between the performance of ALNS and these aspects.

1. Heterogeneous fleet
2. Dynamic order-to-depot assignment
3. Large case-size

We described that the different vehicles have different cost sets. We saw that the costs per trip are very large compared to the costs per distance, or duration. This cost structure makes it more difficult for ALNS to find improvements to the solution. Namely, it is desirable to have occupation of vehicles. If the vehicles are highly occupied the feasible options of planning an order are limited.

Moreover, we discovered a relationship between the dynamic order-to-depot assignment and the effectiveness of ALNS. That the performance of the basic ALNS strategy is inversely proportional to the distance between the depots. If the distance between depots increases, the improvements found by the basic ALNS decrease. In this case a partial optimization approach is more effective. Our research shows that the order-to-depot assignment is crucial. If the assignment is done to strict it is possible that no feasible solution is found.

Lastly, we provided new insight with regards to the application of ALNS on very large cases. Previous research by Simons (2017), shows that it is desirable to remove approximately 15% of tasks from the solution. We found that for the retail client it is more desirable to remove 2% of

Table 7.2: Overview of performance of ALNS strategies with restricted computation time

Case	Basic ALNS	Extended Basic ALNS	ALNS With Fixed Depot Assignment	ALNS-FDATTP	Hybrid ALNS
1	0,00	0,00	0,11	0,00	0,00
2	0,00	0,00	0,13	0,06	0,00
3	0,00	0,00	0,30	0,06	0,00
4	0,00	0,00	0,01	0,14	0,00
6	0,00	0,00	0,02	0,00	0,00
8	0,00	0,00	0,00	0,00	0,00
9	0,04	0,00	0,01	0,01	0,00
10	0,00	0,00	0,02	0,01	0,00
12	0,46	0,00	0,00	0,00	0,00
13	0,00	0,00	0,00	0,18	0,00
14	0,99	0,00	0,00	0,77	0,00
15	0,00	0,00	0,00	0,58	0,00
16	0,58	0,00	0,49	0,12	0,00
17	0,85	0,00	0,03	0,02	0,00
19	0,00	0,00	0,48	0,42	0,00
20	0,01	0,00	0,04	0,04	0,00
Average	0,18	0,00	0,10	0,15	0,00
Variance	0,12	0,00	0,03	0,05	0,00

tasks. The average case size is equal to 2100 orders. This means that on average we remove 41 tasks from the solution. This is in line with the original work of Pisinger and Ropke (2007).

7.2 Recommendations

Following the results that we have shown in the previous section, we recommend ORTEC to implement two ALNS based strategies. For cases 1 until 4, and 19, the ALNS strategy with fixed depot assignment is best. This strategy is very well able to reduce the total distance and duration of the trips, and can improve the solution effectively within 35 minutes. For cases 12-17, we recommend ORTEC to implement the basic ALNS strategy. The Basic ALNS strategy is able to reduce the number of trips used and reduce the total costs significantly for the aforementioned cases. Each of the cases represents a specific part of the distribution network, on a specific day. Assuming each of the cases is representable for its future occurrence, ORTEC can implement the aforementioned recommendations for each case individually. Before implementing the ALNS strategies, we recommend ORTEC to negotiate on this with the retail client. The savings that can be realized on a yearly basis are significant for the retail client. However, we have not looked at the benefits for ORTEC yet. The development costs of implementing these strategies must at least be covered by the retail client. Although our research does not show that, the development costs for ORTEC, which are paid for by the retail client, reduce the benefits of the retail client.

Moreover, we recommend ORTEC to develop software that is able to split cases in multiple subproblems. If larger cases arise in the future, which is very likely, they can be solved more effectively. We have shown that it is effective for some cases of the retail client, dividing a multi-depot VRP in multiple single-depot VRPs. In our research we used an external tool to divide the cases into sub-cases. For practical applications, this is not desirable. The data set that needs to be imported in order to separate the case is very large. The optimization is currently done in CVRS, which also loads all the data. As a result of this, it would be most efficient if the capabilities of splitting a case were programmed within the existing framework of CVRS.

At last we recommend ORTEC to challenge the retail client with concerns to the restricted computation time. In our research, we have shown that running ALNS for 400 iterations, takes longer than the available 35 minutes. However, the results of these tests also show that the total costs of the retail client can be reduced further. The results presented in this thesis can be used to show the retail client the benefits of allowing longer computation time.

7.3 Limitations and Further Research

In this research we briefly investigated the impact of initial solution to the effectiveness of ALNS. Our findings were in line with the work of Simons (2017). Our research found that the improvement found by ALNS is larger when the initial solution is worse. However, the final solution is not better compared to those obtained after a good initial solution. Both the work by Simons (2017) and our research are conclusive on this matter. Although we give reasons why this may be, no evidential proof for the cause is given. Future research at ORTEC could be devoted to the relationship between the initial solution and the effectiveness of ALNS.

Generally literature on ALNS includes many different type of methods which should in their own way be capable of improving the solution. In this research, we have shown which methods are successful in improving the solution and which methods are less successful. If the less successful methods are excluded, this could increase the effectiveness of ALNS. This is in line with a current working paper by Christiaens and Berghe (2018). The paper advocates to use only one powerful recreate and repair method. The paper shows high quality results for rich vehicle routing problems. However, computation times are not explicitly mentioned. A first step to investigate this would be to test ALNS with only ruin and repair methods that are generally successfully in improving the solution.

Our literature review pointed out a number of metaheuristics besides ALNS. Since the purpose of this research was to investigate the applicability of ALNS on the VRP of the retail client, we have not included a comparison to other metaheuristics. Our literature research shows that some of the most recent developed genetic algorithms are able to outperform ALNS in most benchmark cases. A known disadvantage of genetic algorithms is that they are difficult to program. Implementing these algorithms in CVRS where they need to be reusable for all sorts of cases can therefore be difficult. Future research may investigate the applicability of genetic algorithms for rich and real-world cases and how ORTEC can use them.

We have tested a partial optimization method that first solves the transport problem in order to determine the initial depot assignment. The current construction method of ORTEC did not find a feasible solution. As a result, it can be interesting to investigate different construction methods that do find a feasible initial solution. The work of Lalla-Ruiz and Voß (2019) indicates that a partial optimization approach is effective in solving multi depot VRPs. This can be validated for the cases of the retail client.

Bibliography

Alinaghian, M. and Shokouhi, N. (2018). Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, 76:85 – 99.

Çağrı Koç, Bektaş, T., Jabali, O., and Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1 – 21.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 37(10):6809 – 6817.

Bell, J. E. and McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1):41 – 48.

Belloso, J., Juan, A. A., and Faulin, J. (2019). An iterative biased-randomized heuristic for the fleet size and mix vehicle-routing problem with backhauls. *International Transactions in Operational Research*, 26(1):289–301.

Braekers, K., Ramaekers, K., and Nieuwenhuyse, I. V. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300 – 313.

Calvet, L., Wang, D., Juan, A., and Bové, L. (2019). Solving the multidepot vehicle routing problem with limited depot capacity and stochastic demands. *International Transactions in Operational Research*, 26(2):458–484.

Christiaens, J. and Berghe, G. V. (2018). Slack induction by string removals for vehicle routing problems. Technical report, KU Leuven.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., and Vigo, D. (2007). Chapter 6 vehicle routing. In Barnhart, C. and Laporte, G., editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 367 – 428. Elsevier.

Cordeau, J.-F. and Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033 – 2050.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

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

Emeç, U., Çatay, B., and Bozkaya, B. (2016). An adaptive large neighborhood search for an e-grocery delivery routing problem. *Computers & Operations Research*, 69:109 – 125.

Gencer, C., İsmail Top, and Aydogan, E. K. (2006). A new intuitional algorithm for solving heterogeneous fixed fleet routing problems: Passenger pickup algorithm. *Applied Mathematics and Computation*, 181(2):1552 – 1567.

Gendreau, M., Hertz, A., and Laporte, G. (1994). Tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290. cited By 672.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49 – 66.

Grangier, P., Gendreau, M., Lehuédé, F., and Rousseau, L.-M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1):80 – 91.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449 – 467.

Helsgaun, K. (2009). General k-opt submoves for the lin–kernighan tsp heuristic. *Mathematical Programming Computation*, 1(2):119–163.

Joubert, J. and Claasen, S. (2006). A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem. *ORiON*, 22.

Karakatič, S. and Podgorelec, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, 27:519 – 532.

Keese, S. (2018). Shifting up a gear. *Roland Berger Focus*.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Koskosidi, Y. A., Powell, W. B., and Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science*, 26(2):69–85.

Kulkarni, R. and Bhave, P. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1):58 – 67.

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

Lalla-Ruiz, E. and Voß, S. (2019). A popmusic approach for the multi-depot cumulative capacitated vehicle routing problem. *Optimization Letters*.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Li, J., Li, Y., and Pardalos, P. M. (2016). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, 31(2):515–532.

Li, J., Pardalos, P. M., Sun, H., Pei, J., and Zhang, Y. (2015). Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications*, 42(7):3551 – 3561.

Li, J., Wang, D., and Zhang, J. (2018). Heterogeneous fixed fleet vehicle routing problem based on fuel and carbon emissions. *Journal of Cleaner Production*, 201:896 – 908.

Lougee-Heimer, R. (2005). Coin branch and cut solver cbc. Technical report.

Mancini, S. (2016). A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*, 70:100–112. cited By 28.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115 – 129.

Naji-Azimi, Z. and Salari, M. (2013). A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem. *Applied Mathematical Modelling*, 37(6):4316 – 4324.

ORTEC (2019). Introduction training cvrs. unpublished.

Osman, I. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451. cited By 619.

Penna, P. H. V., Subramanian, A., Ochi, L. S., Vidal, T., and Prins, C. (2019). A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. *Annals of Operations Research*, 273(1):5–74.

Perboli, G., Tadei, R., and Tadei, R. (2010). New families of valid inequalities for the two-echelon vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 36:639 – 646. ISCO 2010 - International Symposium on Combinatorial Optimization.

Pichka, K., Ashjari, B., Ziaeifar, A., and Nickbeen, P. (2014). Open vehicle routing problem optimization under realistic assumptions. *International Journal of Research in Industrial Engineering*, 3:46–55.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403 – 2435.

Pisinger, D. and Ropke, S. (2010). *Large Neighborhood Search*, pages 399–419. Springer US.

Potvin, J.-Y. and Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331 – 340.

Salhi, S., Imran, A., and Wassan, N. A. (2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315 – 325. Recent advances in Variable neighborhood search.

Salhi, S., Wassan, N., and Hajarat, M. (2013). The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics. *Transportation Research Part E: Logistics and Transportation Review*, 56:22 – 35.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139 – 171.

Simons, L. (2017). Adaptive large neighborhood search for rich and real-world vehicle routing problems. Master's thesis, TU Delft, Faculty of Electrical Engineering, Mathematics and Computer Science.

Taillard, r., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186.

Tansini, L., Urquhart, M., and Viera, O. (2002). Comparing assignment algorithms for the multi-depot vrp.

Tarantilis, C., Kiranoudis, C., and Vassiliadis, V. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1):148 – 158.

Toth, P. and Vigo, D., editors (2001). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Vidal, T., Crainic, T., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624. cited By 209.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1):15 – 28.

Wang, C., Mu, D., Zhao, F., and Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, 83:111 – 122.

Xiao, Y., Zhao, Q., Kaku, I., and Mladenovic, N. (2014). Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems. *Engineering Optimization*, 46(4):562–579.

Appendices

Appendix A

ORTEC Routing and Dispatch Software


This information is marked as confidential
ORTEC
OPTIMIZE YOUR WORLD

Appendix B

Network Layout


This information is marked as confidential
ORTEC
OPTIMIZE YOUR WORLD

Appendix C

Extra Metaheuristics

C.1 Population search metaheuristics

An important class of population search algorithms is the genetic algorithms. Goldberg (1989) provides an explanation of this type of algorithms. Genetic algorithms are usually improvement algorithms which make use of mechanisms from natural genetics to improve a solution. In its most simple form, a genetic algorithm consist of 3 aspects:

1. Reproduction
2. Crossover
3. Mutation

Based on a population of solutions, one solution is selected. This process is known as reproduction. The probability of selecting a method is done based on the fitness function. Solutions with a higher fitness score have a higher probability of getting selected(survival of the fittest). From the pool of reproduction solutions, pairs are selected. These pairs are used to perform crossover. We illustrate crossover with a simple example. Consider two sequences consisting of 3 numbers:

- 1234
- 7890

Crossover is done by selecting a range within the items of the solution are swapped. If we apply crossover for only the last item in the solutions, we find the following new solutions:

- 1230
- 7894

Then with a certain probability, mutation is applied to any of the solutions in the solution pool. Although the basis of genetic algorithms is fairly straightforwards, it turned out to be effective in solving complex VRP's.

C.2 Learning Mechanisms

The learning model which is mentioned explicitly by Laporte (2009) is called Ant-Colony Optimization (ACO). Ant colony is a metaheuristic in the field of swarm optimization. The metaheuristics are based on the natural behaviour of swarms of insects Bell and McMullen (2004). Ant colony optimization is a metaheuristic which uses 'ants' to find a shortest path between locations. To communicate with each other, ants use a chemical which is called pheromone. In their quest for food, ants leave a trail of this chemical. When an Ant encounters such trail, it decides to follow the trail only if the chemical is strong enough. Short trails between a food source and the nest are travelled faster, which makes the chemical trail stronger. As a result, more ants will follow this shorter trail. The metaheuristic tries to apply this behaviour to find solutions to the VRP. Bell and McMullen (2004) explains the construction of routes with the use of ant colony optimization in the following way. A vehicle is represented by a predefined number of Ants. The first of these ants starts constructing a route until the capacity of the vehicle is reached. Next, the second ant also constructs a tour for the same vehicle. Due to the random behaviour of the ants at the start, this route is probably different than the route determined by the first ant. As more ants have constructed routes, the probability of an ant following an already travelled path increases. This probability increases based on the success of the path in question. After all ants for this vehicle have constructed a tour, the process is repeated for all other vehicles.

Appendix D

Pseudo Code For simulated Annealing

Below we present a pseudo code for a parallel application of Simulated Annealing.

Algorithm 2: Pseudocode of a simulated annealing procedure

```
Set decrease factor β < 1;
Set initial value for cooling parameter α;
Determine number of iterations i;
Determine stopping value γ;
Determine initial Solution and set it as current solution and best found solution so far;
while Cooling parameter α > γ do
    for i iterations do
        Select a neighbour solution;
        if Neighbour solution better than current solution then
            Accept neighbour solution as current solution;
        if Neighbour solution better than best found solution so far then
            Accept neighbour solution as best found solution so far;
        if neighbour solution worse than current solution then
            Accept solution with acceptance probability e^((Neighbour−Current)/T);
    end
    Recalculate cooling parameter: α = β ∗ α
end
```

Appendix E

Cost Sets Per Vehicle Type

This information is marked as confidential

Appendix F

Current Solution Method


This information is marked as confidential
ORTEC
OPTIMIZE YOUR WORLD

Appendix G

Technical details for changing depot assignment strategies


This information is marked as confidential
ORTEC
OPTIMIZE YOUR WORLD

Appendix H

Results Parameter Tuning

This information is marked as confidential

Appendix I

Figures used for component analysis of ALNS

ORTEC
OPTIMIZE YOUR WORLD