Re-allocation of commodities via line haul connections

van Beek, R.J.M.

Award date:
2018

Technische Universiteit Eindhoven
University of Technology

Department of Industrial Engineering and Innovation Sciences
Operations, Planning, Accounting & Control Group

Re-allocation of commodities via line haul connections

R.J.M. van Beek

In partial fulfillment of the requirements for the degree of
Master of Science in Operations Management and Logistics

Supervisors:

Dr. J. Kinable
Prof. Dr. T. van Woensel
Dr. L. Veelenturf

Eindhoven, September 7, 2018

MASTER THESIS

R.J.M. van Beek

September 7, 2018

Keywords: Re-allocation, Line haul, Combinatorial Benders Decomposition, Threshold-based allocation

Contents

List of Figures

List of Tables

Abstract

This thesis addresses a re-allocation problem where commodities can be re-allocated via a line haul if the customer is located in the district of another depot. Consider a situation of multiple depots with each their own non-overlapping service district and customers that are geographically dispersed. The commodities of a customer are located at a fixed depot. The depot can choose whether they want to serve the less-than-a-truckload by themselves or make the depot that is located in the same region as the customer, responsible for delivery. Re-allocation of commodities to other depots means that they have to be transported between depots via a line haul connection. Such a connection is a truck that ships the commodities between two depots. The objective is to find a re-allocation of commodities that results in the least costs for line haul and routing operations. The main goal of the thesis is to obtain insight on how to reduce costs of the decentralized non-information-sharing distribution network. The company that proposed the case do not have the possibilities to make a re-allocation decision based on a calculation of the whole order set. They calculate the routing in a decentralized way based on the orders they received. The company uses the size of an order as a threshold for the automated decision of re-allocation.

Four re-allocation methods are tested in the thesis. The first re-allocation method encloses no further action on re-allocation. Customers are served by their original depot. The second re-allocation method is based on the threshold order size, that is used by the company. Customers are served by the original depot if the order size is above the threshold and they are served by the depot in their region if the order size is beneath that threshold. The third method optimizes this threshold. The re-allocation based on the threshold remains, but the threshold can vary in value. The fourth method is without restrictions of order size. The depot can be served by the original depot or by the regional depot.

The methods are calculated using a MIP, a combinatorial benders decomposition with non-exact subproblems and a heuristic. The combinatorial benders decomposition and heuristic are used to speed up calculations since the exact problem requires lots of computation time. The benders approach splits the problem in a master problem and subproblems. The master problem takes care of the re-allocation via line-haul, strengthened with a relaxation of the routing, where subsequently the re-allocation decisions are used to calculate the routing for each depot independently. The calculations, which are executed on eight sets of 30 customers and eight sets of 50 customers and are terminated on 30 minutes, show that the non-exact approaches outperform the exact approaches on computation time and solution quality. Also is noticed that for 14 of the 34 cases, the first benders iteration equals the best-found solution.

The tested methods prove that all hypotheses were right for the tested instances, although with remarks to the number of customers used in the model. Three hypotheses are tested in the thesis.

1. Cost reduction could be obtained by more efficient re-allocation of commodities and with that, efficient use of the line haul.

The experiments show cost reduction for no re-allocation (7%), re-allocation based on decision variable order size (17%) and re-allocation without the restriction of order size (19%) in comparison with the re-allocation method based on the fixed threshold as used by the company.

2. Cost reduction could be obtained by optimizing the threshold value for the inter-hub exchange of orders.

The results clarify that the most cost-effective way of re-allocation without information sharing is to re-allocate all the nodes that are allowed to be re-allocated to another depot. This lead to an average decrease in costs of 17% in comparison with the current policy of the company. Anyhow, this is calculated for a maximum of 50 customers. The company has to serve at least 2500 customers a day. Such a network is more dense, which will probably have an effect on the cost-effectiveness. Besides that, the company has an unbalanced network which can result in an unequal load that has to be transported between both depots. A recommendation based on the conclusions is to increase the threshold parameter in practice and see whether cost reduction will be obtained.

3. Re-allocation without parameter thresholds will lead to the most cost-effective networks.

The results showed slightly better objectives for the test without restriction on order size in comparison with re-allocation based on the optimized threshold value. However, it is expected that the difference

in costs between the two re-allocation methods will raise further when the number of customers in the network raise. Re-allocation with a known network, viewed from a central perspective can decrease costs up to at least 19%. Recommended is to reconsider the problems for a higher amount of customers if resources and time are available.

1 Introduction

The allocation problem that is proposed in this paper is about re-allocating commodities to hubs via line haul connections. The problem is based on a real-world problem of a transportation company that possesses multiple hubs with a decentralized planning system, which basically means that every hub plans delivery and pick up of their own orders. All the hubs have their own service district with customers, these districts do not overlap. A customer orders transportation of less-than-a-truckload from A to B. A depot is responsible for an order if the ordering customer is in the service district of that depot. Orders are received from customers during the day. The company policy is that when an order is received, the commodities are picked up the same day by the responsible depot and shipped to that depot. These pickups are thus planned dynamically during the day, when trucks are already on the road. The subsequent transport from the depot to B will be planned for the next day. This planning is a route plan for vehicles with commodities to deliver and commodities to pick up. This planning is made every evening per depot independently. A pickup is taken into account in this evening planning if it is arranged to be picked up the next day.

The goal of the planning is to serve all the customers with the least travel costs. The problem is delineated to delivery from the depot to B or a planned pick-up from A to the depot. These orders are known at the moment of constructing the planning. Consider a situation where the commodities for a delivery are at the responsible depot and the commodities that have to be picked up have to return to the depot. Trucks will drive a route along multiple addresses to deliver the commodities, taking into account time windows and truck capacity. The independent planning of the depots is described as a capacitated vehicle routing problem with time windows and simultaneous pickup and delivery. The vehicle route problem is widely studied and can be described as: 'serving a set of customers that are geographically dispersed around a depot' (Braekers, Ramaekers, & Van Nieuwenhuyse, 2016). The customers are served using trucks, such that each customer is visited once and the trucks start and end at the depot (Koç, Bektaş, Jabali, & Laporte, 2016). One truck can serve multiple customers, depending on the sizes of the orders and the capacity of the truck. Trucks can be filled with commodities for delivering until the maximum capacity of the truck is reached. The fleet of trucks differs in capacity and cost per kilometer, described by (Koç et al., 2016) as a heterogeneous fixed fleet. Each customer is served by a delivery, a pickup or both a delivery and pickup simultaneously. First Min (1989) and later Ai and Kachitvichyanukul (2009) described this problem with the principle that all commodities that have to be delivered originate from the depot and all commodities that have to be picked up must be shipped back to the depot. Commodities are never picked up at a customer and delivered at a destination, so go from A to B, on the same day, let alone on the same route. A pickup can only be accomplished if the truck has space to load the order. If a truck starts with a full load, then it has to deliver some of the goods first before a pickup can be made. The goods are of the kind of one-commodity, but order sizes can differ. The customers expect a truck to arrive in between a given time-window. However, the truck is allowed to leave the customer outside this time window. The time windows are hard, which means that violation of the earliest and latest time is not accepted.

However, it could be possible that one of the addresses is outside the service district of the responsible depot. Those orders can be re-allocated via a line-haul connection to the depot that is responsible for the service district. A line-haul connection is defined as one or more trucks that transport commodities between two depots, which take place upfront of the day scheme. The truck transports commodities to another depot, unload them and load other commodities to be transported back to the first depot. The policy is that commodities can be re-allocated to another hub only if the address is in the service district of that hub. See figures 1 and 2. The squares indicate depots, the circles are addresses of which the left depot is responsible for and triangles are addresses of which the right depot is responsible for. The service districts are indicated by the outer lines and routing is depicted with arrows. Note that the left depot has a node in the service district of the right depot. The depot can serve it by its own, as stated in figure 1. The depot has also the option to transport it via a line haul to the other depot, where the other depot has to deliver it, as depicted in figure 2.

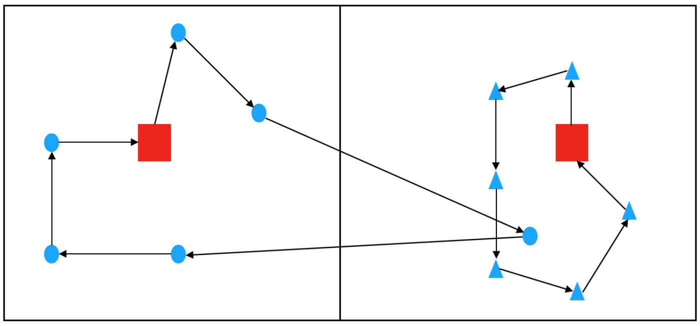

Figure 1: Routing without line haul

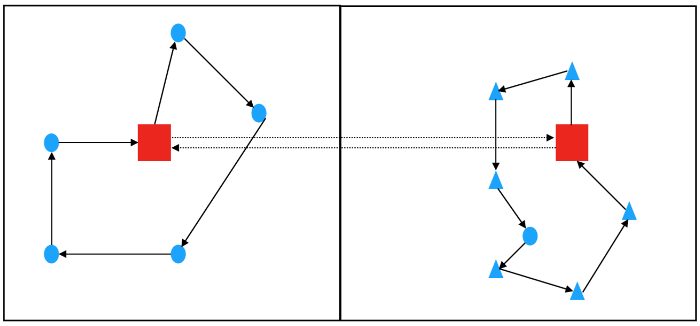

Figure 2: Routing with line haul

The main objective of the thesis is to obtain insight on how to reduce costs of the decentralized non-information-sharing distribution network. Hypothesized is that the costs can be reduced by more efficient re-allocation of commodities and with that, efficient use of the line haul. An important remark is that the re-allocation decision is made at the moment that an order is received. Only the orders which are received earlier that day are known at that moment and orders of the other hubs aren't known. The company that proposed the case is a merge of multiple companies. Due to juridical and resource restrictions, they aren't able to plan centrally and they aren't able to execute the re-allocation when all the orders are known. The company uses the size of an order as a threshold for the automated decision of re-allocation currently. This means that an order is added to the line haul and thus outsourced to another depot if the order size is beneath a certain threshold value and if the address of the order is in the district of the other depot. This value is based on experience currently, which raise questions about the accuracy of the value and whether other parameters also have an influence on the re-allocation decision. Three hypotheses are stated:

1. Cost reduction could be obtained by more efficient re-allocation of commodities and with that, efficient use of the line haul.
2. Cost reduction could be obtained by optimizing the threshold value for the inter-hub exchange of orders.
3. Re-allocation without parameter thresholds will lead to the most cost-effective networks.

A real challenge is saying something about a system with lots of unknown parameters. Historical data can simplify this challenge by using it to predict unknown parameters. In this study, the historical data of the company will be used to retrospectively optimize, which should give insight into an optimal way of re-allocating the commodities in the distribution network. The historical data makes it possible to review the network centrally. Small benchmark instances of the data will be tested, since a calculation on the network size of the company is complex and computationally heavy.

Section 2 describes what the literature determines about the exchange of commodities in a decentralized situation. The problem for optimal re-allocation of the commodities is introduced in section 3 as a Mixed Integer Problem (MIP). Further on will section 4 provide a method to calculate the optimal order size for use of a threshold for the re-allocation. The MIP is computationally heavy and needs much time to calculate an exact solution, a combinatorial benders decomposition approach is used to speed up the computations. This can be found in section 5. Further on section 6 provides a heuristic to speed up the vehicle routing problem. After that, section 7 shows results of tests for where the problem is solved without re-allocation of commodities, with re-allocation based on the company threshold and re-allocation without order size restrictions. Lastly, section 8 reflects on the hypotheses and provide recommendations.

2 Literature Review

The allocation problem is widely studied in the area of operations research, with different kind of structures. Although literature about the re-allocation of commodities without knowledge of the rest of the network, mostly defined as decentralized planning without sharing of information, is scarce. As stated in (van Beek, 2018) cost savings can be obtained through collaboration between different companies. The multiple depots can be seen as different companies due to the decentralized way of operation. In the current situation, no information between depots is shared. Interesting to know is how the exchange of orders is mainly handled in the literature if the planning is decentralized and without sharing of information. Problems in the literature propose methods to exchange goods between companies in order to reduce cost, which is mainly the same principle as in the problem of this thesis.

Most of those methods have a focus on an auction-based exchange using a central request pool (van Beek, 2018). The paper of Li et al. (2015) describes, for instance, a system with a central coordinating system, as well as Wang and Kopfer (2015) did. Even Clifton et al.(2008), who imposed that they created a non-information sharing system, need a central algorithm. They tried to secure it for privacy reasons. Although the swap algorithm needs information from both sides of carriers to detect outliers based on distance and swap them. An interesting similarity is that the algorithm takes into account the costs of swapping an individual request. The conclusion is that the best results are obtained by a central view, as stated in (van Beek, 2018).

Let's consider the network centrally, instead of the decentralized method of the company. And let's consider that re-allocation to the depot in the other service district is allowed for all orders if the address is in that district. How to determine the costs or savings of the re-allocation of commodities? The literature depicts different methods for calculating the costs of a re-allocation decision. Most of them consider the cost changes of the whole network while exchanging single or bundled requests. Li et al. (2015) determines the efficiency of a request via a minimum marginal cost strategy. The profit is determined by comparing the profit of the network before and after the exchange of the request. They can pick an incoming request based on a maximum marginal cost strategy in return. Özener et al. (2011) determines the marginal cost of an exchange by subtracting the cost of the new lane from the exchanging lane. They calculated the profit based on the marginal cost of the exchange. Clifton et al. (2008) determine the costs based on the farthest distance to the endpoint. These papers chose a system of exchanging requests one by one. The goal is to decrease costs for both the partners in the system. Although in this case, the goal is to decrease the costs for the total system and not for the individual hubs. It could even be possible that one hub has increasing costs while the cost of the total system decreases. Berger and Bierwirth (2010) investigated the cost savings of single request exchange against bundled request exchange and concluded that the bundled ones outperform the single requests. In this situation, the total costs have to be reduced and not only those of the individual hubs. However, the conclusion is that re-allocation based on a single request provides worse results than when a group of requests is offered for exchange. The company can also provide better results by sharing information about a whole truck than exchanging orders one by one. Also, the papers state that the savings or costs of a re-allocation are calculated by recalculating the whole network. For that reason, the vehicle route problem is used to consider the costs of a re-allocation.

The company exchanges orders via direct re-allocation based on order size without considering the rest of the network. However, probably more orders are available for exchange between depots. Interesting to know is whether there is a more cost-effective way of exchanging requests. None of the papers use parameter thresholds to swap the requests. This paper investigates the use of the current threshold parameter order size, as well as how re-allocation of commodities to an other depot can also be determined. Important to know is which parameters play a role in collaborative planning systems. These parameters should be taken into account in the problems. For instance, Gansterer and Hartl (2016) depicts that request exchange based on geographical information can be cost-effective for decentralized systems, where the distance to depots are important for overlapping customer regions. Cruijssen et al. (2007) conducted a research on the effect of varying market conditions in a joint route planning system. They found out that cost reduction can be created when order sizes are small, time windows are narrow and inter-customer distance is large. Cruijssen et al. (2007), and also Guajardo (2015) note that increasing demand and the number of joining companies lower the effect on cost savings. Based on this can be concluded that more parameters play a role in the exchange of orders than only order size.

The literature depicts different methods for calculating the costs of a re-allocation decision. In this calculation order size, time windows and distances are interesting parameters which have to be considered, stated by literature and professionals of the company as well. The model of the company is for that reason stated as a multi-depot VRP with time windows, a heterogeneous fixed fleet and simultaneously pickup and delivery. Dondo et al. (2003) worked on an approach to solve a multi-depot VRP with time windows and a heterogeneous fixed fleet. They proposed an optimal approach for 10 nodes, where Dondo and Cerda (2007) came up with a three-phase cluster-based optimization approach for 100 nodes for the same kind of routing problem. Bettinelli et al. (2011) used a branch-and-cut-and-price algorithm to calculate exact solutions. Ashfar-Nadjafi and Ashfar-Nadjafi (2017) proposed a heuristic that uses local searches on neighborhood solutions. These papers all point to a multi-depot VRP with time windows and a heterogeneous fixed fleet. The problems show a fast increase in computation times with an increase in the number of nodes. These problems show the most similarities with the problem in this thesis. But, the difference with the problem in this paper is the simultaneously pickup and delivery. As stated before, Min (Min, 1989) and (Ai & Kachitvichyanukul, 2009) explained this problem based on a public library distribution system. No paper is found that research the same routing problem, let alone that they also research the re-allocation part. Although, lots of papers are written about individual parts of the routing problem. The literature writes frequently about VRP with additional constraints for time windows, heterogeneous fleet and multi-depot. These papers bring up solutions to solve the VRP, but do not re-allocate commodities. The additional re-allocation problem will make the problem harder to solve. For that reason, approaches with relaxations or heuristics are desired in this problem. Papers about allocation mainly describe the allocation of commodities to depots or production facilities, sometimes combined with the determination of the depot location. The problem is mostly that there is a limited amount of commodities or production capacity. The paper of Kinable and Trick (2014), for instance, show slight similarities with this problem with their combination of allocation and routing. Although, these kinds of problems focus more on allocation than on re-allocation. In this problem, the commodities are already at a depot.

3 Problem formulation

The problem is first considered with the assumption that re-allocation of commodities is only restricted to the rule that re-allocation is allowed if the customer is in the same service district as the depot to which the customer is re-allocated. In section 4 the restriction to re-allocate based on order size is explained. The problem contains a multi-depot VRP with time windows, a heterogeneous vehicle fleet and simultaneously pickup and delivery. It is adapted with the option to re-allocate commodities via a line haul connection.

Consider the distribution network as a directed graph $G = (V, A)$, which connects n customer nodes and m depots through a set of minimum cost arcs $A = \{(i, j)|i, j \in V, i \neq j\}$. Depots are indexed by $h \in H(h = 1, \ldots, m)$. The depots are duplicated for the use of time window constraints. The duplicated depots are indexed by $h' \in H'(h' = m+1, \ldots, 2m)$, in the same order as the original depots. Customers are indexed by $i \in C(i = 2m, \ldots, 2m+n)$, the triplet composes to vertex set $V = C \cup H \cup H'$. Customers are associated with a delivery quantity d_i and a pickup quantity p_i, both non-negative. It is possible that a customer has only a delivery, only a pickup or both a delivery and pickup. As already stated, the commodities that have to be picked up must be shipped back to the depot, so the load that is picked up has to be returned to the depot after completing the route. Servicing of a customer i has to start in between time window $[e_i, l_i]$, where e_i is the earliest time and l_i is the latest acceptable time of starting servicing the customer. This starting time is indicated as t_i^k for every customer i and vehicle k. Customers have a service time s_i, which consists of a fixed time value and a variable value that is calculated per unit of demand. The fixed value consist of time for entering and leaving the customer, where the variable value is considered as a value to load or unload one unit of demand.

The commodities of a customer i, that have to be delivered, are originally located at depot h, indicated by the binary parameter β_{ih}. This is the depot that is the responsible actor of the ordering institution and thus where the commodities are located. The parameter has the value 1 if depot h is responsible for node i and a value 0 otherwise. Commodities can be shipped to an alternative depot h, indicated by the binary parameter α_{ih}. This alternative depot is determined by the region where the customer node is located. Therefore, the depot h which serves the region of customer i has a value of 1 for α_{ih}, the other depots will get the value 0. Note that α_{ih} can be equal to β_{ih} for the same depot. In that case, the customer is located in the same service district as the responsible depot and is a transfer to another depot precluded. The depot to which a customer is assigned, is determined by the binary decision variable τ_{ih}. This variable will receive value 1 if the customer i is served by depot h, and 0 otherwise. z_{hg} is an integer variable that indicates the number of trucks that travel between depots h and g. It indicates the line haul connection. Depending on the number of customer nodes, their accompanying demand and the capacity q_{line} of the line haul truck, the number of vehicles z_{hg} are raised. Meant by this is that the value of z_{hg} raises by 1 if the sum of the demand of re-allocated commodities exceeds the truck capacity.

At every depot $|K_h|$ vehicles are stationed. The vehicles are indexed as $k \in K$ and are divided over the depots, indexed as $k \in K_h$ for each depot. Vehicles can be of different types and for that reason, every k has its own capacity, noted as q_k. The load in the vehicle is denoted by continuous decision variable y_{ij}^k. This is the load that is inside truck k after node i and before node j. A truck starts at a depot with the commodities of the customers that have to be served by that truck. The truck returns at the depot with the load that is picked up on the way. The traverse of an arc in the set of A is defined by binary decision variable x_{ij}^k with cost c_{ij}^k based on the distance d_{ij} together with a cost factor a^k based on the vehicle type and the driving time t_{ij} together with an employee cost factor b. Binary decision variable r_i^k decides whether a vehicle is assigned to customer i or not. The MIP can be stated as follows. Clarification of parameters and variables can be found respectively in table 1 and 2.

$$minimize \sum_{k\in K}\sum_{h\in H} b(t^k_{h+m} - t^k_h) + \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} a^k d^k_{ij} x^k_{ij} + \sum_{h\in H}\sum_{g\in H} c^{line}_{hg} z_{hg} \tag{1}$$

s.t.

$$\sum_{j\in V} x^k_{ij} = r^k_i \qquad \forall i \in C, \forall k \in K \tag{2}$$

$$\sum_{j\in V} x^k_{ji} = \sum_{j\in V} x^k_{ij} \qquad \forall i \in V, \forall k \in K \tag{3}$$

$$\sum_{j\in C} x^k_{hj} \leq \mathbb{1}_{K_h}(k) \qquad \forall h \in H, \forall k \in K \tag{4}$$

$$\sum_{j\in C} y^k_{hj} = \sum_{j\in C} d_j r^k_j \qquad \forall h \in H, \forall k \in K_h \tag{5}$$

$$\sum_{i\in C} y^k_{i,h+m} = \sum_{i\in C} p_i r^k_i \qquad \forall h \in H, \forall k \in K_h \tag{6}$$

$$\sum_{i\in V} y^k_{ij} - \sum_{i\in V} y^k_{ji} + r^k_j(p_j - d_j) = 0 \qquad \forall j \in C, \forall k \in K \tag{7}$$

$$y^k_{ij} \leq x^k_{ij} q_k \qquad \forall i \in V, \forall j \in V, \forall k \in K \tag{8}$$

$$t^k_i + t_{ij} + s_i - M(1 - x^k_{ij}) \leq t^k_j \qquad \forall k \in K, \forall i \in V, \forall j \in V \setminus H : i \in H' \tag{9}$$

$$\sum_{h\in H} \tau_{ih} = 1 \qquad \forall i \in C \tag{10}$$

$$\tau_{ih} = \sum_{k\in K_h} r^k_i \qquad \forall h \in H, \forall i \in C \tag{11}$$

$$\tau_{ih} \leq \max(\beta_{ih}, \alpha_{ih}) \qquad \forall i \in C, \forall h \in H \tag{12}$$

$$z_{hg} \geq \frac{\sum_{i\in C} d_i \tau_{ig} \beta_{ih} \alpha_{ig}}{q_{line}} \qquad \forall h \in H, \forall g \in H, h \neq g \tag{13}$$

$$z_{hg} = z_{gh} \qquad \forall h \in H, \forall g \in H, h \neq g \tag{14}$$

$$x^k_{ij} \in \{0, 1\} \qquad \forall i, j \in A, \forall k \in K \tag{15}$$

$$0 \leq y^k_{ij} \leq q_k \qquad \forall i, j \in A, \forall k \in K \tag{16}$$

$$r^k_i \in \{0, 1\} \qquad \forall i \in C, \forall k \in K \tag{17}$$

$$e_i \leq t^k_i \leq l_i \qquad \forall i \in V, \forall k \in K \tag{18}$$

$$\tau_{ih} \in \{0, 1\} \qquad \forall i \in C, \forall h \in H \tag{19}$$

$$z_{hg} \in \mathbb{Z}_{\geq 0} \qquad \forall h \in H, \forall g \in G, h \neq g \tag{20}$$

The objective function (1) is divided into two kinds of costs that have to be minimized. The first part is about routing costs. These costs are based on the employee hours and the driven kilometers. The employee hours are calculated by variable t^k_{h+m} and t^k_h, which denote respectively the time of return at a depot and the time of start. The second part is about the line haul costs, based on the distance and driven hours from depot h to g. Constraints (2) states that if an arc that is leaving node i is enabled for vehicle k, the node is served by vehicle k. Note that together with equations (10) and (11) it guarantee that only one vehicle is servicing a node. Constraints (3) adds a flow conversation to the model, which means that a vehicle that enters a customer, must also leave a customer. Next, constraints (4) guarantees that a vehicle can only leave a depot if it is available for that depot. It is also allowing vehicles to stay in the depot if they are not used. $\mathbb{1}_{K_h}(k)$ is an indicator function that returns 1 if k is in the set of K_h and zero otherwise. Constraints (5) until (8) are about the capacitated part of the vehicle routing problem. Where, constraints (5) and (6) ensure that the total quantity that is leaving and entering the depot is equal to, respectively, the total demand and the total pick-up quantity of the nodes that are served by vehicle k. Furthermore, constraints (7) ensures that the load of a truck before visiting a node minus the load of a truck after visiting that node is equal to the demand of the customer. Followed by constraints (8), that guarantees that the vehicle capacity is not exceeded, as well that the arc does not have a load if it is not occupied. Constraints (9) ensure that the service time is calculated. Further, constraints (10) until (14) are about the line haul operations. Constraints (10) until (12) ensure that a customer is assigned to maximum one of the possible depots. Equation (13) guarantees that the line haul raises by adding the demand of transferable nodes to vehicles. Afterward, equation (14) make sure that a line haul operation is two-sided. A truck that operates in the line haul will always return to the home depot.

Table 1: Parameter definitions

Parameter	**Description**
H	Set of depots
H'	Duplicated set of depots
C	Set of customers
V	$C \cup H \cup H'$
A	Set of arcs, $\{(i,j) \mid i,j \in V, i \neq j\}$
K	Set of vehicles
K_h	Set of vehicles assigned to depot h
$[d_i, p_i]$	delivery and pick-up demand of node i
$[e_i, l_i]$	time window of node i
β_i	$\begin{cases} 1 & \text{if depot h is the responsible depot of customer i} \\ 0 & \text{otherwise} \end{cases}$
α_i	$\begin{cases} 1 & \text{if customer i is in the service district of depot h} \\ 0 & \text{otherwise} \end{cases}$
d_{ij}	distance from node i to node j
t_{ij}	driving period from node i to node j
c_{ij}	Routing cost i to node j
q_k	Capacity of vehicle k
a	cost per distance unit
b	cost per time unit
M	Large positive penalty constant
$line$	vehicle that operates in the line haul
m	number of depots
n	number of customers

Table 2: Decision variable definitions

Variable	**Description**
x_{ij}^k	$\begin{cases} 1 & \text{if node j is served after node i by vehicle k} \\ 0 & \text{otherwise} \end{cases}$
r_i^k	$\begin{cases} 1 & \text{if node i is served by node k} \\ 0 & \text{otherwise} \end{cases}$
τ_{ih}	$\begin{cases} 1 & \text{if node i is assigned to depot h} \\ 0 & \text{otherwise} \end{cases}$
y_{ij}^k	Non-negative continuous variable denoting the total load remaining in vehicle k while traveling along arc (i,j)
t_i^k	Non-negative continuous variable denoting the time vehicle k starts servicing node i
z_{hg}	Integer variable denoting the number of line hauls between depot h and depot g

4 Re-allocation based on order size

As already stated in section 1 the company currently uses order size as a threshold for re-allocation of commodities. The value of loading meters is used to indicate the order size. Commodities of different sizes and shapes flow through the network. The value loading meters indicates how much units space in a truck is needed to transport the commodity. The conversion metric relaxes the problem to one-commodity. The company re-allocates commodities to another depot if that customer is in the service district of the alternative depot and the order size is lower than 0.3 loading meters. This policy applies to deliveries and pick-ups. The rule simplifies the problem by solving the re-allocation via a fixed parameter. Basically, only the VRP for each depot remains. Those subproblems are way easier to solve. Although the fixed value 0.3 is probably not the optimal threshold value for each set of data. Let's state L as a decision variable to calculate the optimal threshold value for a dataset. As hypothesized this should result in a more cost effective re-allocation based on loading meters. Optimization with the threshold value as a decision variable could be obtained by adding the following constraints to the problem.

$$L - d_i - M(\sum_{h \in H} \tau_{ih}\alpha_{ih}) \leq 0 \qquad \forall i \in C : d_i > 0 \qquad (21)$$

$$L - p_i - M(\sum_{h \in H} \tau_{ih}\alpha_{ih}) \leq 0 \qquad \forall i \in C : p_i > 0 \qquad (22)$$

$$L - d_i + M(1 - \sum_{h \in H} \tau_{ih}\alpha_{ih}(1 - \beta_{ih})) \geq \epsilon \qquad \forall i \in C : d_i > 0 \qquad (23)$$

$$L - p_i + M(1 - \sum_{h \in H} \tau_{ih}\alpha_{ih}(1 - \beta_{ih})) \geq \epsilon \qquad \forall i \in C : p_i > 0 \qquad (24)$$

$$L \geq 0 \qquad (25)$$

Where L is a new continuous decision variable denoting the number of loading meters that ensure re-allocation to the alternative depot. Equations 21 and 22 should ensure that customers with respectively demand d_i or pick-up p_i lower than L are allocated to the alternative depot. The equations 23 and 24 ensures that customers with demand d_i or p_i higher than L aren't allocated to the alternative depot. The right-hand side of both constraints contains an ϵ, which is defined as a small value. The ϵ ensures that customers with d_i or p_i equal to L are not free of choice for re-allocation. It ensures that commodities with an order size beneath L are re-allocated, and commodities with an order size equal or bigger than L stay at the depot. Note that equation 10 ensures that a customer is allocated to one depot only.

5 Combinatorial Benders Decomposition Approach

The MIP of section 3 is computationally heavy. A combinatorial benders decomposition approach is used to speed up the computations. The combinatorial benders decomposition is an approach that splits the problem in a master and subproblem with the goal to eliminate re-allocation decisions that are not efficient. Fewer options simplify the problem and accelerate the computations. The re-allocation of customers to depots and the accompanying line haul operations are composed in the master problem. This master problem will result in a re-allocation of customers to depots. The objective of the master problem is a lower bound to the problem. Once the decision variables of the master problem that indicate the re-allocation of customers to depots are fixed, the problem decomposes into $|H|$ independent vehicle routing problems. The subproblems generate costs for the routing for each depot, together with the costs for the line haul it composes to the costs of the re-allocation decision. The lowest value generated by the iterations is defined as the upper bound. Iteratively the master- and subproblems are calculated until the lower bound and upper bound are equal. At every iteration, an optimality cut is added with the goal to generate a different solution in the next iteration. Algorithm 1 provides a more detailed outline.

Algorithm 1 Combinatorial Benders Decomposition Algorithm

procedure CBD
    Lower Bound (LB) $\leftarrow -\infty$
    Upper Bound (UB) $\leftarrow \infty$
    $t \leftarrow 0$
    while $LB \neq UB$ **do**
        $t \leftarrow t + 1$
        Solve Master Problem (MP)
        Get solution $\overline{\tau}$
        $LB \leftarrow f^T z + R$
        for $h \in H$ **do**
            procedure SUBPROBLEM
            get solution F_h
            add optimality cut (eq: 55) to MP
        $UB \leftarrow \min(UB, f^T z + \sum_{h \in H} F_h)$
    return UB and optimal solution $\overline{\tau}$

5.1 Master Problem

The constraints 26 until 33 shape the master problem. The objective function (26) minimizes the cost for the line haul operations. Note that the other constraints match with the allocation constraints of section 3. The goal of the master problem is to allocate the customers to depots. Note that re-allocation based on order size, as proposed in section 4, is also applicable to the master problem. The constraints 21 until 25 can be added to the problem beneath. These will simplify the problem and reduce the number of iterations.

$$minimize \sum_{h \in H} \sum_{g \in H} c_{hg}^{line} z_{hg} + \sum_{h \in H} R_h \tag{26}$$

$$\sum_{h \in H} \tau_{ih} = 1 \qquad \forall i \in C \tag{27}$$

$$\tau_{ih} \leq \max(\beta_{ih}, \alpha_{ih}) \qquad \forall i \in C, \forall h \in H \tag{28}$$

$$z_{hg} \geq \frac{\sum_{i \in C} d_i \tau_{ig} \beta_{ih} \alpha_{ig}}{c^{line}} \qquad \forall h \in H, \forall g \in H, h \neq g \tag{29}$$

$$z_{hg} = z_{gh} \qquad \forall h \in H, \forall g \in H, h \neq g \tag{30}$$

$$\tau_{ih} \in 0, 1 \qquad \forall i \in C, \forall h \in H \tag{31}$$

$$z_{hg} \in \mathbb{Z}_{\geq 0} \qquad \forall h \in H, \forall g \in G, h \neq g \tag{32}$$

$$R_h \geq 0 \qquad \forall h \in H \tag{33}$$

Subsequently, the master problem can be strengthened by adding a relaxation of the subproblem. The current master problem will lead to a branch and check (Thorsteinsson, 2001), with a nearly random allocation of nodes. A strong relaxation results in a smaller amount of combinations to search for. Remind that subproblems are vehicle routing problems for each depot independently. A valid relaxation for a vehicle routing problem is a minimum spanning tree. A minimum spanning tree generates a bound for a vehicle routing problem. For every depot, a minimum spanning tree is built for the nodes that are assigned to that depot via the τ_{ih} variables. The minimum spanning trees are further strengthened to the kind of 1-tree. Figures 3 and 4 consider the same situation as in figures 1 and 2 that can be found in the introduction, although these explain feasible solutions of the 1-tree relaxation. The solid black lines show the undirected arcs of a feasible spanning tree. The minimum spanning tree is built with the nodes that are assigned to the depot. In figure 3 the circles are assigned to the left depot and the rest is assigned to the right depot, where in figure 4 the nodes that are in the service district of the depot are assigned to that depot. Further, the striped line indicates the arcs between the depot and two customers to create a 1-tree.

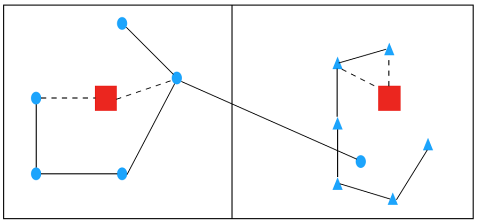

Figure 3: 1-trees without line haul

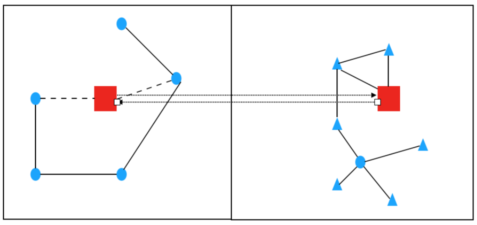

Figure 4: 1-trees with line haul

The following constraints relax the problem.

$$R_h \geq \sum_{e \in E} c_e x_e^h + b \left(\sum_{e \in E} t_e x_e^h + \sum_{i \in C} \tau_{ih} s_i \right) \qquad \forall h \in H \qquad (34)$$

$$\sum_{e \in E \setminus \delta(h)} x_e^h = \sum_{i \in C} \tau_{ih} - 1 \qquad \forall h \in H \qquad (35)$$

$$\sum_{e \in \delta(h)} x_e^h = 2 \qquad \forall h \in H \qquad (36)$$

$$\sum_{e \in E: i \in S, j \in S} x_e^h \leq |S| - 1 \qquad \forall h \in H, \forall S \subseteq C, S \neq \emptyset, C \qquad (37)$$

$$x_e^h \leq \tau_{ih} \qquad \forall h \in H, i \in C, e \in \delta(i) \qquad (38)$$

$$\tau_{ih} \in 0,1 \qquad \forall i \in C, \forall h \in H \qquad (39)$$

$$x_e^h \in 0,1 \qquad \forall e \in E, \forall h \in H \qquad (40)$$

An important difference with the MIP from section 3 is that the graph is now of an undirected form, where $G = (V, E)$ with $E = \{(i,j)|i,j \in V, i < j\}$. Alternatively defined as $e \in E$. Constraints (34) determine the cost for this minimum spanning tree, comparable to the costs of the MIP of section 3. Constraints (35) ensures that there is one edge less than the number of allocated nodes for a depot, a characteristic of the minimum spanning tree. Constraints (36) ensure that at least two edges are connecting the depot with the minimum spanning tree. Furthermore constraints (37) impede sub tours. These sub tours are found after solving the problem. An algorithm searches for groups of nodes, connected via edges, that do contain at least as many edges as nodes. This group of nodes is added as a set S. The problem is calculated iteratively until no sub tours remain in the solution. Constraints (38) ensure arcs only between two nodes assigned to the same depot.

5.2 Subproblem

The subproblems will generate a minimum routing cost F_h for every depot $h \in H$. The subproblems make use of the re-allocation decisions of the customers to depots that are determined in the master problem. The variable τ_{ih} is now used as a fixed parameter, which is indicated by $\overline{\tau}_{ih}$. The subproblem has a lot of similarities with the MIP of section 3. Although, nearly all constraints that handle the re-allocation are removed. Also, note that every single subproblem contains only one depot. These problems are much easier to solve than the original problem.

$$Subproblem_h \quad \forall h \in H$$

$$minimize\ F_h = \sum_{k \in K} b(t^k_{h+m} - t^k_h) + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} a^k d^k_{ij} x^k_{ij} \tag{41}$$

s.t.

$$\sum_{j \in V} x^k_{ij} = r^k_i \qquad \forall i \in C, \forall k \in K_h \tag{42}$$

$$\sum_{j \in V} x^k_{ji} = \sum_{j \in V} x^k_{ij} \qquad \forall i \in V, \forall k \in K_h \tag{43}$$

$$\sum_{j \in C} x^k_{hj} \leq 1 \qquad \forall k \in K_h \tag{44}$$

$$\sum_{j \in C} y^k_{hj} = \sum_{j \in C} d_j r^k_j \qquad \forall k \in K_h \tag{45}$$

$$\sum_{i \in C} y^k_{i,h+m} = \sum_{i \in C} p_i r^k_i \qquad \forall k \in K_h \tag{46}$$

$$\sum_{i \in V} y^k_{ij} - \sum_{i \in V} y^k_{ji} + r^k_j (p_j - d_j) = 0 \qquad \forall j \in C, \forall k \in K_h \tag{47}$$

$$y^k_{ij} \leq x^k_{ij} q_k \qquad \forall i \in V, \forall j \in V, \forall k \in K \tag{48}$$

$$t^k_i + t_{ij} + s_i - M(1 - x^k_{ij}) \leq t^k_j \qquad \forall k \in K, \forall i \in V, \forall j \in V \setminus h : i \in h' \tag{49}$$

$$\overline{\tau}_{ih} = \sum_{k \in K_h} r^k_i \qquad \forall i \in C \tag{50}$$

$$x^k_{ij} \in 0,1 \qquad \forall i,j \in A, \forall k \in K_h \tag{51}$$

$$0 \leq y^k_{ij} \leq q_k \qquad \forall i,j \in A, \forall k \in K_h \tag{52}$$

$$r^k_i \in 0,1 \qquad \forall i \in C, \forall k \in K_h \tag{53}$$

$$e_i \leq t^k_i \leq l_i \qquad \forall i \in V, \forall k \in K_h \tag{54}$$

The solutions generated by the subproblems still need lots of computation times. The number of customers and the complexity of real-live data often do not allow a problem to be solved exactly. In these situations, heuristics can be applied (El-Sherbeny, 2010). The advantage of the decomposition is that it is now able to be solved by a heuristic that can handle the routing part. Rahmaniani et al. (2017) stated that research to the use of heuristics in the benders decomposition is limited, although they expect that heuristics can help to speed up for large problems. These heuristics are able to find solutions faster than the exact method. However, the solutions of the heuristic are not exact. The use of a heuristic changes the problem to one of a non-exact kind.

5.3 Cuts

The algorithm 1 already showed that an optimality cut is added tot the master problem. The cuts are used to avoid current and similar solutions in succeeding iterations (Rasmussen & Trick, 2007). This cut is added after master and subproblems both generated solutions.

Optimality cut:

$$R_h \geq F_h^t - M(\sum_{\overline{\tau}_{ih}^t=0} \tau_{ih} + \sum_{\overline{\tau}_{ih}^t=1} 1 - \tau_{ih}) \qquad \forall h \in H, \forall t \in T \qquad (55)$$

The variable R_h in equation (55) can be traced back to the second part of the objective function (26) of the master problem. Furthermore, the cut contains the variable F_h^t which are the optimal solutions that are generated in the subproblems for every $h \in H$. Set T is introduced here as the set of iterations. Every iteration of the combinatorial benders decomposition a succeeding iteration t is added to the set T. The equation contains the summations with $\overline{\tau}_{ih}^t$ at the right-hand side. The left summation will add a τ_{ih} variable if customers where not allocated to depot h in iteration t, while the right summation add a τ_{ih} if customers where allocated to depot h. This part avoids current and similar solutions in succeeding iterations. The cut ensures that R_h will be bounded by the value of F_h^t if the allocation remains the same for that depot and iteration. This will happen if the routing cost of the minimum spanning tree will exceed the upper bound or no other solutions are available.

6 Heuristic approach

A heuristic is used to calculate the actual effect of the re-allocation decisions. Since exact methods are very time-consuming for the large instances, a heuristic approach is chosen to handle the problem and calculate an upper bound for it. The objective gained from the master problem is based on a minimum spanning tree that relaxes on direction, capacity and time windows. The master problem works as lower bound to the problem. Key features of the problem are time windows and capacity. The problem can be simplified by dropping the rule of the number of vehicles per depot. The amount of owned vehicles is in the real situation fixed, but additional vehicles can be hired and loads can be outsourced. The customer characteristics of the problem, known as time windows and the demand for delivery and pick-up, aren't changeable and for that reason considered as important. VRP's with time window constraints are very time consuming, even for heuristics. Another key feature of the heuristic should be that it can handle thousands of nodes in acceptable computation times. Gendreau and Tarantilis (2010) wrote a literature review comparing 17 kinds of heuristics that were able to handle the instances of Gehring and Hornberger (2001), which contain up to 1000 nodes. They found out that the heuristics of Hashimoto and Yagiura (2008) and Nagata et al. (2010) deliver the best combination of solution quality and computation times. The results show that the Edge Assembly Memetic Algorithm (EAMA)(Nagata et al., 2010) outperforms over the path relinking approach when instances increasing. Other papers in the literature also refer to the heuristic or methods that are used in the heuristic of Nagata et al. The heuristic is used as a comparison for results, for instance in the paper of Kalina Vokrinek (2012). But more often parts of the method are used as a basis for other heuristics, see (Blocho & Czech, 2013), (Vidal, Crainic, Gendreau, & Prins, 2015), (Nalepa & Blocho, 2016) (Miranda & Conceição, 2016), (Schneider, 2016) or (Laporte, 2007). Study on the algorithm concluded that it is adaptable for this vehicle routing problem since it contains a method with a feasibility check were additional constraints can be added.

6.1 EAMA phase 1

The EAMA algorithm contains two phases. The first phase is about minimization of routes, called the $RMheuristic()$. This phase is extensively described in (Nagata & Bräysy, 2009). The algorithm starts initially with creating separated routes, let denote this set of routes as R and an individual route as r. These initial routes are very basic, namely from the depot to the customer and back. Using the $DeleteRoute(R)$ procedure the routes will be merged iteratively using three different methods. See Algorithm 2 for an outline of this procedure. First, a route r will be chosen randomly, deleted from R and the customers of this route are put aside in an ejection pool (EP). A customer, stated as i_{in}, is taken from this pool and inserted on all possible positions in the remaining routes. If the insertion is confirmed feasible, it is added to the set of possible insertions N_{in}^{fe}. Feasibility is defined by the capacity and time window constraints. Note that the capacity constraint is different than in (Nagata & Bräysy, 2009).

Capacity constraints:

$$y_i \leq q \qquad \forall i \in r \qquad (56)$$

where

$$y_1 = \sum_{i \in r} d_i \qquad (57)$$

$$y_i = y_{i-1} - d_i + p_i \qquad \forall i \in r \setminus \{1\} \qquad (58)$$

Time window constraints:

$$e_i \leq t_i \leq l_i \qquad \forall i \in r \qquad (59)$$

where

$$t_1 = e_1 \qquad (60)$$

$$t_i = \max(t_{i-1} + s_{i-1} + t_{i-1,i}, e_i) \qquad \forall i \in r \setminus \{1\} \qquad (61)$$

A feasible insertion position is randomly chosen from the set and the routes are updated with the inserted customer. If there are no feasible insertion positions, the squeeze method will be used. This method searches for the insertion position with the least exceeding of constraints. Let denote the amount of exceeding as $F_p = F_q + F_{tw}$. F_q is the total sum of capacity that is exceeding vehicle capacity q. F_{tw} is the total sum of exceeding time for each customer calculated by $max(t_i - l_i, 0)$. If a time window is exceeded, the t_{j-1} is set back to l_{j-1} for the next customer j in the route. The customer will be inserted at that position with the least F_p, where after the infeasible route is repaired using the local search techniques 2-opt* (Potvin & Rousseau, 1995) , in- and out-relocate and intra- and interexchange (Kindervater & Savelsbergh, 1997). The infeasible routes will be iteratively repaired by searching for the local search move that will lower F_p the most, until all routes are feasible. If repair didn't succeed R will be restored to its original form. In the last case, a third method searches for the best insertion-ejection combination. In a lexicographic manner, customers in the feasible routes are ejected from the routes until i_{in} finds a feasible insertion position. The ejected customers will be added to the ejection pool. A penalty system helps customers which are difficult to insert to give them privileges in the methods. After the insertion-ejection method, a perturbing method mixes R by executing p random feasible local search moves. The $DeleteRoute(R)$ procedure ends when the ejection pool is empty or a given time limit T_{delR} is reached.

Algorithm 2 Delete route

```
procedure DeleteRoute(R, T_delR, p)
    Select and remove a random r from R
    Initialize EP with customers from removed route
    Initialize penalty counter pen_i := 1, ∀i ∈ C
    while EP ≠ ∅ and time < maxTime do
        Select and remove customer i_in from EP using the LIFO strategy
        if N_in^fe ≠ ∅ then
            Select randomly insert position from N_in^fe
            Update R
        else
            Select infeasible insertion such that min(F_p)
            Update R
            while F_p > 0 do
                randomly select infeasible r from R
                Find local search move that min(F_p)
                if local search move is found then
                    update R
                else
                    break
            if F_p > 0 then
                restore R
        if no success to insert i_in then
            set pen_i = pen_i + 1
            find eject-insert combination such that min(∑_{i_eject ∈ C} pen_i)
            add all i_eject to EP
            Perturb p times with random feasible local search moves
    if EP = ∅ then
        Return R
    else
        Return original R
```

6.2 EAMA phase 2

The second phase, stated in (Nagata et al., 2010), starts after a population of initial solutions that is created by the $RMheuristic()$. The amount of initial solutions, saved in set N, is defined by parameter *pop*. The parameters have an effect on the quality en computation time of the solution. The initial solutions are used as input for the objective minimization phase. In this phase first the set of initial solutions is permuted. The algorithm works with an EAX crossover that needs two parents, P_a and P_b, to produce *ch* offspring solutions. The parents are feasible routes, taken from the set N_{pop}. Every set R in the set N_{pop} is selected once as P_a and once as P_b. The EAX crossover mechanism, introduced by Nagata (2007) for the capacitated problem, consist of five steps. In step one a directed graph is created as $G_{AB} = (V, E_A \cup E_B \setminus E_A \cap E_B)$. In step two AB-cycles are created by linking edges of P_a and P_b one by one in the opposite direction, where P_a have to been seen as the forwarding edge and P_b is followed in the reversed direction. In step three a set E is created with the single strategy or the block strategy. In the single strategy, a random AB-cycle is chosen as set E. The block-strategy adapt on the single strategy with adding AB-cycles that have at least one node in common but have fewer nodes than the first cycle. In step four P_a is used as a base where the edges $E \cap E_A$ are removed and the edges $E - set \cap E_B$ are added. In step five the sub tours are eliminated. If the routes that are generated by the crossover are infeasible a repair procedure will be started. The infeasible routes will be iteratively repaired by searching for the local search move that will decrease F_p, and has the least value for F. Iterations continue until all routes are feasible or no feasible local searches are found. A local search procedure will optimize the objective when feasible routes are obtained. Search moves are only executed on routes r that contain differences in comparison with P_a. The R of P_a will be updated by the optimized route set R_{new} if the objective of the optimized route is lower than the objective of P_a. The objective is calculated slightly different than the one in the paper to create similarity with the MIP, see equations 62 until 64. Note that the values of t_i have to be calculated in such a way that it minimizes F_{time}. The time variable t_i is calculated by $t_{i-1} + s_{i-1} + t_{i-1,i} + w_{i-1}$, where w_i denotes a certain waiting time at node i in route r.

$$F = F_{dist} + F_{time} + P_{tw} + P_c \tag{62}$$

where

$$F_{dist} = \sum_{r \in R} \sum_{i=1}^{|r|-1} d_{i,i+1} a \tag{63}$$

$$F_{time} = \sum_{r \in R} b(t_{|r|} - t_1) \tag{64}$$

The algorithm will continue until no improvements are found for $2 * g_{max}$ times. After g_{max} non-improved iterations the EAX procedure will use the block-strategy instead of the single strategy.

Algorithm 3 EAMA

```
procedure EAMA(pop, ch, g_max, T_delR, p)
    m = ∑_{i∈C} d_i/q
    R_0 = RMheuristic(m, T_delR, p)
    update m = |R|
    add R to set N
    for i=1 to pop do
        R_i = RMheuristic(m, T_delR)
        add R_i to N
    gen = 0
    while gen < 2 * g_max do
        perturb N
        for i = 0 to i < |N| do
            P_a = R_i, P_b = R_{i+1} (NOTE: if (R_i = |N|) : P_b = R_0)
            R_best = P_a
            for i = 0 to i < ch do
                generate graph G_AB = (V, E_A ∪ E_B \ E_A ∩ E_B)
                create AB-cycles
                create set E (if gen ≥ g_max use block strategy, else single strategy)
                create graph with P_a−E-set∩E_A+E-set∩E_B
                eliminate sub tours
                if R_new contains infeasible r then
                    while F_p > 0 do
                        select randomly infeasible r
                        Find local search move that min(F) and decreases F_p
                        if local search move is found then
                            update R_new
                        else
                            break
                if R_new is feasible then
                    while F is decreasing do Find local search move that min(F)
                        if local search move is found then
                            update R_new
                if F then(R_new) < F(R_best
                    R_best = R_new
            R_i = R_best
        if F(N) decreases then
            gen = 0
        else
            gen = gen + 1
```

7 Experimental results

The hypotheses are tested via the MIP, Benders Decomposition and heuristic approaches on instances of real datasets. The MIP and master problem of the benders decomposition are programmed using Gurobi in Java and the heuristic is programmed in Java. The calculations are performed on a 2.2 GHz Intel Core i7 PC.

7.1 Dataset

The approaches are tested on a real historical data set of a company that contains multiple depots. The historical data that is obtained contains around 400,000 orders that are placed from the 1st of January 2018 until the 20th of June 2018. The orders contain information of locations, demand volumes, dates, time windows, service districts and the depot that is responsible for the order. The data set is reduced to around 250,000 records after a data cleaning procedure. Most of the orders are removed due to missing data. Analysis of the dataset after the cleaning procedure shows a mean of around 2,600 orders a day. Note that the mean total number of orders a day is probably higher. February had the highest mean of the number of orders with 2,695 per day, where June had only 2,500 a day. National free days and weekends are filtered out. Mondays represent generally the lowest number of orders with 1,875 as a minimum. The highest values are found around the 3,000 orders a day. The locations are spread out over the whole Benelux, known as Belgium, the Netherlands and Luxembourg. Figure 5 shows a mix of random and clustered customer locations through the whole Benelux. The clustered nodes suggest cities. The squares indicate the locations of the depots.

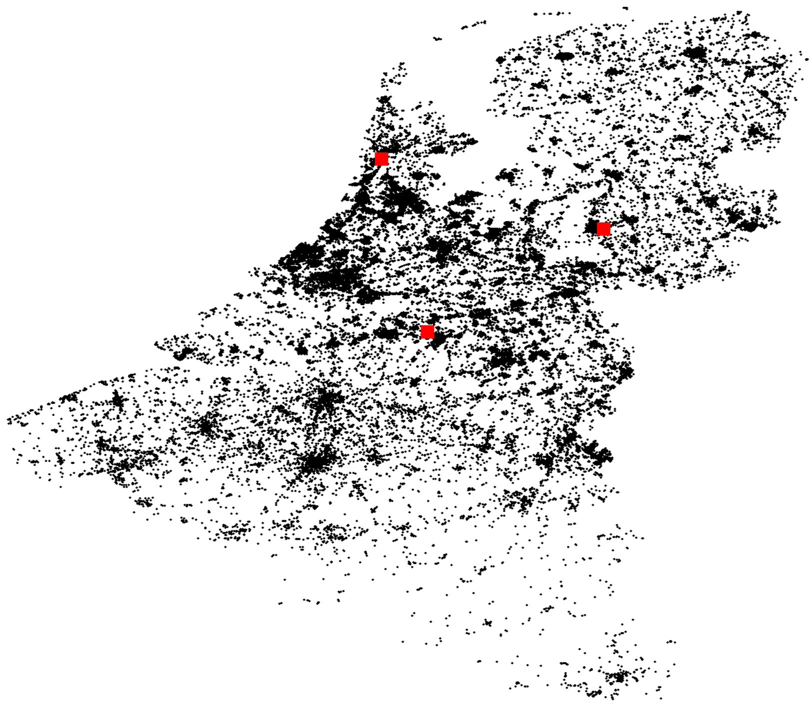

Figure 5: Overview of all customer locations

As stated above the daily data contain more than 2,500 orders a day. Calculating the optimal network of that size is complex and computationally heavy. The resources are limited, for that reason the sets are created in such a way that reasonable results can be obtained in about 30 minutes. The data in the sets have to contain a decent number of orders that are located in a different service district different than the service district of the responsible depot. The reason for this is that it improves the chance of assigning line haul connections. A data set with customers exclusively in the same service district as the responsible depot will only test the routing part and not the re-allocation. Also, the number of customers and total order sizes per depot are controlled for the creation of instances. Eight sets of 30 customers and eight sets of 50 customers are created. All sets contain two depots. Order size and time windows are, as mentioned before, influencing characteristics of the cost of the network. Those parameters are verified on similarities to the network. The average time window for customers is 505 minutes, by this is meant the difference in time between earliest arrival time and latest arrival time. Remarkable for this average is that 54% of the time windows have a gap between 400 and 600 minutes and 18% have the full time window of 960 minutes (from 4:00 until 20:00). The average order size of an order is 1.1 loading meters. Although remarkable is that 74% of the orders have an order size of less than 1 loading meter, where 54% of the orders are less than 0.5 loading meters and 23% of the orders satisfy the threshold of less than 0.3 loading meters. Further, the average ratio of demand orders and pick up orders are 89 against 11, verified is whether the instances have a ratio similar to this. The created set is representative when all those figures are met.

Table 3 depict a summary of the instances with, in order of column, the number of depots, the total number of customers, for each depot the number of customer originally located, for each depot the maximum number of customers that is allowed to be served by that depot, sum of all load that is allowed to be shipped from depot 1 to 2 or backwards, the average order size in loading meters and the average time window in minutes.

Table 3: Summary of instances

Instance	$\|H\|$	$\|C\|$	$\|C\| : \beta_{ih} = 1$		$max(\|C\|)$		Line-haul load		Demand size	Time window
Depot			1	2	1	2	1	2		
Set30-1	2	30	21	9	25	17	4.0	3.5	1.58	393
Set30-2	2	30	16	14	21	17	2.3	5.0	0.98	434
Set30-3	2	30	19	11	26	17	10.5	1.8	1.27	461
Set30-4	2	30	19	11	26	20	6.2	8.4	1.16	418
Set30-5	2	30	19	11	23	15	2.3	2.8	1.34	409
Set30-6	2	30	18	12	24	16	3.7	2.6	0.94	510
Set30-7	2	30	18	12	23	16	1.3	4.8	1.45	430
Set30-8	2	30	21	9	25	17	3.9	1.9	1.76	465
Set50-1	2	50	33	17	44	24	7.6	12.2	1.1	438
Set50-2	2	50	35	15	44	26	6.1	7.3	1.6	442
Set50-3	2	50	35	15	45	21	6.6	6.0	1.2	432
Set50-4	2	50	36	14	45	24	6	5.28	0.98	487
Set50-5	2	50	35	15	42	23	12.6	6.1	0.84	385
Set50-6	2	50	35	15	44	28	10.7	6.2	0.93	413
Set50-7	2	50	39	11	45	22	10.6	3.9	1.3	395
Set50-8	2	50	29	21	41	32	7.8	8.7	0.95	466

The time windows of the depots are the widest. The range is from 240 minutes (4:00 A.M.) until 1200 minutes (8:00 P.M.) and the depots are not taken into account for the calculation of the average time window. The capacity of all vehicles, q_k, are defined as 13.6 loading meters. Distances and driving periods between two nodes, indicated as d_{ij} and t_{ij}, are calculated via the Open Source Routing Machine http://project-osrm.org. The geographical locations, which are received from the company, are used as input values. The site returns a distance and duration. These values are saved to an asymmetric distance matrix for the distances and to an asymmetric duration matrix. The cost per distance a and the cost per time unit b are both defined on 1.

7.2 Setup

Different experiments of re-allocation are executed on the datasets. The goal of the experiments is to be able to conduct a conclusion whether the hypotheses are right or wrong. Firstly, a baseline assessment will be done to be able to report a reduction in costs. The customer nodes will be assigned to the depot where they are already staying, this means no line haul operations. The MIP will be calculated with τ_{ih} as fixed parameters based on that allocation, this simplifies the problem to only the routing part. The allocation is also calculated using the heuristic for each depot independent. Secondly, an assessment of the re-allocation based on the fixed threshold order size, noted as L, will be done. This re-allocation method is the current method of the company. The current fixed value of $L = 0.3$. This re-allocation method is again solved using the MIP and the heuristic. In the third experiment optimization on loading meters is applied. This experiment will be conducted using the MIP and the combinatorial benders decomposition. As hypothesized this should result in a more cost-effective re-allocation, anyhow at least as good as the previous methods. Fourth, re-allocation without restriction on the loading meters is tested. This approach should result in the best re-allocation decisions. The test is conducted using the MIP and the combinatorial benders decomposition. Additionally, a fifth experiment is tested which should give insight into the solution quality of the first iteration of the Benders approach. The objective of this is to test the use of an allocate first, route second principle. Such an approach is much faster since it only needs one iteration of the benders decomposition. So, the following five tests are executed on the instances.

Test 1 No re-allocation;

Test 2 Re-allocation based on fixed parameter order size.

Test 3 Re-allocation based on decision variable order size.

Test 4 Re-allocation without restriction of order size

Test 5 Performing a single Benders Decomposition

For calculations of the MIP is chosen to use enough, although not too many vehicles in such a way that the fleet size isn't restricting. The parameters settings for the EAMA are fixed with: $pop = 200$ parents, $ch = 20$ offspring solutions, $g_{max} = 10$ succeeding iterations, $T_{delR} = 10$ seconds, $p = 100$ local search moves. Parameters settings on the EAMA as a subproblem in the benders decomposition approach are downsized for the matter of computational speed increase. The parameters are then: $pop = 50$ parents, $ch = 20$ offspring solutions, $g_{max} = 8$ succeeding iterations, $T_{delR} = 10$ seconds, $p = 100$ local search moves. The maximum running time for all problems is fixed to 1800 seconds, after this time the MIP or algorithm is terminated. For the MIP problems with 50 customers, a warm-start is added to the MIP. A warm-start is a feasible solution which should assist the problem with solving. The MIP will use this solution as a start. The feasible solution is obtained by executing the $RMheuristic()$ of the EAMA minimization phase.

7.3 Results

The results of tests 1 until 5 can be found in tables 4, 5, 6, 7 and 8. Table 4 until 7 present the results of the four re-allocation methods, indicated as tests 1 until 4. As already stated is every method tested on two approaches, the best objective of both approaches can be found in bold text format.

Test 1: No re-allocation

Table 4 shows computations for a situation where no re-allocation is performed. All nodes of which the commodities are at the responsible depot are served by this responsible depot. The table displays the results for both the MIP and heuristic approach. The objectives and lower bounds of the MIP are the best objectives and bounds that were found in 30 minutes, further is the gap and computation time in seconds presented. For the heuristic approach the objective, the difference in percentages between the heuristic objective and the MIP bound and the computation times are presented. The results show that the MIP outperforms the heuristic on solution quality for most of the instances with 30 customers. The reason for this can be found in the route solutions provided by the methods. The heuristic makes use of a route minimization phase. Comparison of the routes shows that in these cases more routes result in a cheaper solution. However, for the sets with 50 customers, the heuristic outperforms the MIP for nearly all sets. An important remark is that the computations are terminated at 30 minutes. The optimal value of the MIP should always contain the best result when solving to optimum. It is still interesting that the heuristic gives results with good quality in way less time than the MIP, certainly when the instance size raises. The conclusion is that the heuristic should be preferred over the MIP. Even more, when instance size raises

Table 4: Test 1 - No re-allocation

	MIP				Heuristic		
Instances	Objective	Bound	Gap	CPU (s)	Objective	Gap to MIP bound	CPU (s)
Set30-1	**4,689**	3,030	35%	1,800	4,862	38%	172
Set30-2	**3,876**	1,951	50%	1,800	4,204	54%	87
Set30-3	4,306	1,976	54%	1,800	**4,278**	54%	143
Set30-4	5,126	2,898	43%	1,800	**5,117**	43%	90
Set30-5	**4,831**	2,313	52%	1,800	4,869	52%	84
Set30-6	**4,285**	2,376	45%	1,800	4,286	45%	129
Set30-7	4,513	2,495	45%	1,800	**4,493**	44%	85
Set30-8	**3,626**	1,736	52%	1,800	3,630	52%	107
Set50-1	7,635	2,287	70%	1,800	**6,023**	62%	582
Set50-2	**6,609**	2,528	62%	1,800	6,708	62%	577
Set50-3	10,317	2,748	73%	1,800	**6,663**	59%	363
Set50-4	9,108	2,530	72%	1,800	**6,107**	59%	558
Set50-5	8,543	2,171	75%	1,800	**5,813**	63%	605
Set50-6	8,883	2,710	69%	1,800	**6,144**	56%	519
Set50-7	9,488	2,471	74%	1,800	**6,653**	63%	1,067
Set50-8	6,267	2,340	63%	1,800	**6,243**	63%	361

Test 2: Re-allocation based on fixed parameter order size

Table 5 shows computations for a model where re-allocation is performed based on order size 0.3. The table shows the same kind of results as in table 4 Deliveries or pick-ups with size less than 0.3 are outsourced to the alternative depot, others are restricted to be executed by the responsible depot. The results show again that the MIP is able to find better objectives in 30 minutes for the sets with 30 customers, but is outperformed by the heuristic when the instances increases. Again the heuristic is also way faster.

Table 5: Test 2 - Re-allocation based on fixed demand size ($L = 0.3$)

	MIP				Heuristic		
Instances	Objective	Bound	Gap	CPU (s)	Objective	Gap to MIP bound	CPU (s)
Set30-1	5,280	3,204	39%	1,800	**5,226**	39%	92
Set30-2	**4,358**	2,547	42%	1,800	4,438	43%	82
Set30-3	4,912	2,630	46%	1,800	**4,838**	46%	138
Set30-4	5,416	3,281	39%	1,800	**5,370**	39%	64
Set30-5	**4,929**	2,684	46%	1,800	5,067	47%	110
Set30-6	**4,541**	2,851	37%	1,800	**4,541**	37%	105
Set30-7	4,513	2,438	46%	1,800	**4,504**	46%	80
Set30-8	3,648	1,717	53%	1,800	**3,644**	53%	118
Set50-1	7,552	2,285	70%	1,800	**6,434**	64%	714
Set50-2	10,250	2,994	71%	1,800	**7,429**	60%	686
Set50-3	10,742	3,202	70%	1,800	**7,160**	55%	400
Set50-4	9,522	2,928	69%	1,800	**6,571**	55%	863
Set50-5	8,924	2,901	67%	1,800	**6,267**	54%	626
Set50-6	9,484	2,955	69%	1,800	**6,472**	54%	446
Set50-7	9,325	2,313	75%	1,800	**6,653**	65%	911
Set50-8	9,098	3,053	66%	1,800	**6,721**	55%	315

Test 3: Re-allocation based on decision variable order size

Table 6 also shows computations for the re-allocation based on the size of the order. Nevertheless, the threshold is not fixed for this test but used as a decision variable. The table presents the results of test 3 calculated by the MIP and combinatorial benders decomposition for every instance. The MIP displays the objective, the lower bound, the gap between objective and lower bound, the calculated value L and the running time in seconds. The combinatorial benders decomposition displays additionally to the same kinds of columns the number of iterations that are performed. The problem is getting harder to solve with this extra allocation constraints, the MIP as well the Benders decomposition needs the full 30 minutes in case of 50 customers. The results show the same threshold value for MIP and Benders decomposition for 5 of 8 sets with 30 customers. A nice result is that the benders decomposition approach is able to solve the best objective in 14 of the 16 cases. The purpose of the benders decomposition was to obtain better results in less time. An exploration of the datasets shows that the value of the decision variable L, that is calculated by the benders decomposition, contains the same value as the biggest order that is allowed to be exchanged to the other depot for all instances. This suggests that the best results can be obtained when all allowed orders are re-allocated, which is a rule that is realizable in a decentralized non-information sharing environment. The number of trucks that are needed for the line haul to make this decision is unfortunately not balanced, where the instances have a balanced line haul connection. By this is meant that the summed order size of the re-allocatable customers probably will result in a different number of trucks back and forth. For instance, the service district of the company is not equally divided. One depot has 50% of the workload, where others have 20% and 30%. A different amount of truck back and forth will increase the costs, the reduction that can be obtained through collaboration will be decreased. Important to mention is that the lower bound of the MIP isn't comparable with the lower bound of the decomposition approach since the decomposition isn't exact.

Table 6: Test 3 - Re-allocation based on variable demand size

	MIP					Combinatorial Benders Decomposition					
Instances	Objective	Bound	Gap	L	CPU (s)	Objective	Bound	Gap	L	Iter.	CPU (s)
Set30-1	**3,994**	1,997	50%	2.40	1,800	4,127	4,127	0%	2.40	21	614
Set30-2	3,876	1,740	55%	0.12	1,800	**3,665**	3,665	0%	1.84	12	278
Set30-3	3,687	1,772	52%	7.77	1,800	**3,649**	3,649	0%	7.77	18	575
Set30-4	4,420	2,139	52%	4.00	1,800	**4,171**	3,879	7%	4.00	39	1800
Set30-5	**4,497**	2,146	52%	2.00	1,800	4,553	4,553	0%	2.00	12	265
Set30-6	3,883	1,890	51%	2.00	1,800	**3,789**	3,789	0%	2.00	20	694
Set30-7	4,513	2,016	55%	0.40	1,800	**4,145**	4,145	0%	2.80	16	453
Set30-8	3,784	1,550	59%	0.40	1,800	**3,464**	3,464	0%	2.00	28	888
Set50-1	7,505	2,071	72%	0.20	1,800	**4,882**	3,939	19%	7.20	7	1800
Set50-2	8,676	2,347	73%	0.10	1,800	**6,398**	4,345	32%	4.40	13	1800
Set50-3	9,178	2,482	73%	0.11	1,800	**5,838**	4,380	25%	2.90	10	1800
Set50-4	10,148	2,171	79%	0.15	1,800	**5,234**	3,976	24%	2.40	4	1800
Set50-5	7,889	2,167	73%	0.10	1,800	**4,696**	3,883	17%	4.40	15	1800
Set50-6	9,691	2,373	76%	0.20	1,800	**5,953**	4,037	32%	4.20	3	1800
Set50-7	6,478	2,218	66%	0.30	1,800	**5,971**	4,249	29%	2.80	12	1800
Set50-8	7,970	2,195	72%	0.10	1,800	**5,251**	4,372	17%	4.00	14	1800

Test 4: Re-allocation without restriction of order size

Table 7 shows results of the computations for test 4. In this test, the restriction of re-allocation via the threshold order size is released. This makes the problem again harder to solve. The decomposition approach, for instance, did for the first time not solve any of the cases before termination. Again 14 of the 16 cases show better objectives for the benders decomposition than for the MIP. Important to mention is that the objectives of the MIP for the sets with 50 customers after 30 minutes are equal to the value that is generated by the heuristic for the warm-start. In the other three tests, this is only for 2 out of 8 the case.

Table 7: Test 4 - Re-allocation without restriction of order size

	MIP				Combinatorial Benders Decomposition				
Instances	Objective	Bound	Gap	CPU (s)	Objective	Bound	Gap	Iter.	CPU (s)
Set30-1	**3,917**	1,971	50%	1,800	4,002	3,218	20%	29	1,800
Set30-2	3,867	1,828	53%	1,800	**3,278**	3007	8%	30	1,800
Set30-3	3,857	1,641	57%	1,800	**3,539**	2,879	19%	27	1,800
Set30-4	5,460	2,103	61%	1,800	**4,201**	3,149	25%	14	1,800
Set30-5	4,497	2,162	52%	1,800	**3,793**	3,169	16%	41	1,800
Set30-6	**3,874**	1,934	50%	1,800	4,344	3,464	20%	53	1,800
Set30-7	4,146	2,000	52%	1,800	**4,029**	3,207	20%	59	1,800
Set30-8	3,535	1,488	58%	1,800	**3,389**	2,650	22%	27	1,800
Set50-1	8,258	2,061	75%	1,800	**4,890**	3,659	25%	6	1,800
Set50-2	9,944	2,397	76%	1,800	**5,948**	3,852	35%	2	1,800
Set50-2	9,892	2,445	75%	1,800	**5,701**	3,821	33%	3	1,800
Set50-4	9,634	2,190	77%	1,800	**5,065**	3,792	25%	7	1,800
Set50-5	8,347	2,167	74%	1,800	**4,689**	3,331	29%	11	1,800
Set50-6	7,646	2,399	69%	1,800	**5,953**	3,876	35%	1	1,800
Set50-7	9,418	2,255	76%	1,800	**5,706**	3,824	33%	9	1,800
Set50-8	8,104	2,186	73%	1,800	**5,057**	3,785	25%	7	1,800

Test 5: Performing a single benders decomposition

Additionally to the tests on the re-allocation methods, is analyzed whether the first iterations of the benders decomposition provide qualitative good results. The first iteration results of the benders decomposition of test 3 and 4 are presented in table 8. The objectives that are presented in bold indicate instances where the first iterations showed immediately the best-found solution in 30 minutes. The first iteration of test 4 produced solutions that are on average 9% higher than the lowest solution for instances with 30 customers. The first iteration was here zero times the best solution. Although for instances with 50 customers 5 out of 8 times the first iteration was the best, with an average difference of 2%. For test 3 yields an average increase of 4 % for 30 customer sets and 2% for 50 customer sets. Here respectively 4 and 5 times the first solution was the best solution. The differences between the sets with 30 customers in test 4 and the other results can be a consequence of the number of iterations found by the solver. For the 30 customer instances in test 4 significant more iterations are found than in the other tests. Apparently, when more iterations are calculated, the chance of finding better solutions raises. Interesting is to have a look at the computation times of one iteration. The times are reduced to less than a minute for sets with 30 customers. However, also here yields that an increase of customers leads to an increase of computation times.

Table 8: Test 5 - Performing a single Benders Decomposition

	Single Benders iteration test 3					Single Benders iteration test 4			
Instances	Objective	Bound	Gap	L	CPU (s)	Objective	Bound	Gap	CPU (s)
Set30-1	**4,127**	3,097	25%	2.40	16	4,432	3,068	31%	35
Set30-2	4,211	2,752	35%	0.12	28	4,205	2,752	35%	31
Set30-3	**3,649**	2,719	25%	7.77	22	3,827	2,713	29%	29
Set30-4	**4,171**	3,104	26%	4.00	24	4,226	3,097	27%	58
Set30-5	4,908	3,195	35%	0.20	31	4,951	3,195	35%	29
Set30-6	**3,789**	2,961	22%	2.00	40	3,811	2,961	22%	40
Set30-7	4,497	3,053	32%	0.40	28	4,550	2,964	35%	27
Set30-8	3,743	2,567	31%	0.40	27	3,656	2,494	32%	31
Set50-1	**4,882**	3,614	26%	7.20	169	4,936	3,607	27%	263
Set50-2	**6,398**	3,852	40%	4.40	92	6,153	3,821	38%	967
Set50-2	6,365	3,875	39%	2.90	167	**5,701**	3,793	34%	556
Set50-4	5,506	3,737	32%	2.40	243	5,493	3,737	32%	244
Set50-5	**4,696**	3,260	31%	4.40	103	**4,689**	3,260	30%	124
Set50-6	**5,953**	3,876	35%	4.20	375	**5,953**	3,876	35%	674
Set50-7	6,026	3,827	36%	6.07	103	**5,706**	3,801	33%	179
Set50-8	**5,251**	3,725	29%	4.00	104	**5,057**	3,717	26%	128

7.4 Analysis of results

Interesting results can be detected, when observing the differences between the first four tests. The minimum objective of the two approaches, that are used for calculation of the first four tests, is used to compare the differences between the re-allocation methods. See table 9, every column present the difference in minimum objective between test x and y. The percentage indicates the increase or decrease of costs between test x and test y, calculated with the formula $\frac{x-y}{x}$. Figures 6 until 11 present scatter plots where the objectives of test x are plotted against the objectives of test y. Every data point is an instance. If the data point is above the diagonal line it means that the method on the x-axes has a better result and vice versa for the y-axes.

Table 9: Comparison of costs between tests

	Test comparison: Test x $\leftrightarrow$ Test y					
Instances	1 $\leftrightarrow$ 2	1 $\leftrightarrow$ 3	1 $\leftrightarrow$ 4	2 $\leftrightarrow$ 3	2 $\leftrightarrow$ 4	3 $\leftrightarrow$ 4
Set30-1	-11%	15%	16%	24%	25%	2%
Set30-2	-12%	5%	15%	16%	25%	11%
Set30-3	-13%	15%	17%	25%	27%	3%
Set30-4	-5%	18%	18%	22%	22%	0%
Set30-5	-2%	7%	10%	9%	12%	3%
Set30-6	-6%	12%	11%	17%	16%	0%
Set30-7	0%	8%	10%	8%	11%	3%
Set30-8	0%	4%	7%	5%	7%	2%
Set50-1	-7%	19%	19%	24%	24%	0%
Set50-2	-12%	3%	10%	14%	20%	7%
Set50-2	-7%	12%	14%	18%	20%	2%
Set50-4	-8%	14%	17%	20%	23%	3%
Set50-5	-8%	19%	19%	25%	25%	0%
Set50-6	-5%	3%	3%	8%	8%	0%
Set50-7	0%	10%	14%	10%	14%	4%
Set50-8	-8%	16%	19%	22%	25%	4%
Average	-7%	12%	15%	17%	20%	3%

Comparing test 1 and 2, it can be concluded that no re-allocation (test 1) is better than re-allocation based on a fixed order size of 0.3 (test 2). Figure 6 depict better objectives for 14 of the 16 instances. An interesting remark is that the other 2 did not contain orders with an order size beneath 0.3. The average increase in costs is 7%. The explanation for this behavior can be found in the number of nodes with an order size of 0.3. The number of nodes that are tested do not fill a full truck for the line haul connection, which results in a relatively high line haul cost per order. Besides that, the truck still has to drive to the other service district to serve the remaining customers with order sizes bigger or equal to 0.3. For that reason is expected that the difference will decrease when the number of nodes raises.

Analysis of the difference between test 3 and the first two tests shows that re-allocation based on variable order size (test 3) provide better results than both test 1 (no re-allocation) or 2 (re-allocation based on fixed order size). Figures 7 and 9 depict both all instances above the diagonal, which means that the objectives of test 3 are better for all instances. The average decrease in costs between test 1 and 3 is 11%, where the average decrease in costs between test 2 and 3 even 17% is. These results confirm the hypothesis that cost reduction could be obtained by optimizing the threshold for the exchange of orders. The average computed threshold value is 3.57. That value is way higher than the value of 0.3 that is used now. For all the instances in the benders decomposition, the value L was equal to the highest order size that was able to be outsourced. All the nodes that had to be delivered in another service district were re-allocated. Anyhow, the number of customers in the network have to be taken into account before drawing a conclusion.

Finally test 4 shows, as hypothesized, the best improvements in comparison to the other methods. Test 4 was a re-allocation method without restriction on order size. An average improvement in costs of 14% is found between test 4 and test 1. The difference between results for test 2 and 4 is even higher, an average of 19% is obtained. The improvements of test 4 in comparison with test 3 are 3% of cost reduction. This last improvement seems to be low since the solutions of test 3 re-allocated all the possible customers, where test 4 also chose to re-allocate a high amount of nodes. Research on the parameters of the customers that are re-allocatable did not result in useful insights. The research was obtained with the goal to find parameters that have an effect on the re-allocation decision of customers. The results of all instances of test 4 show a high percentage of re-allocation. Only a few customers, which were allowed to be re-allocated, were not re-allocated. Those numbers are too small to adopt useful insights on why they weren't re-allocated.

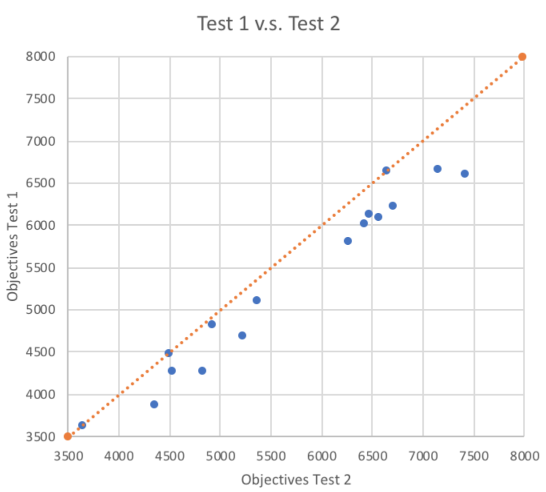

Figure 6: Test 1 v.s. Test 2

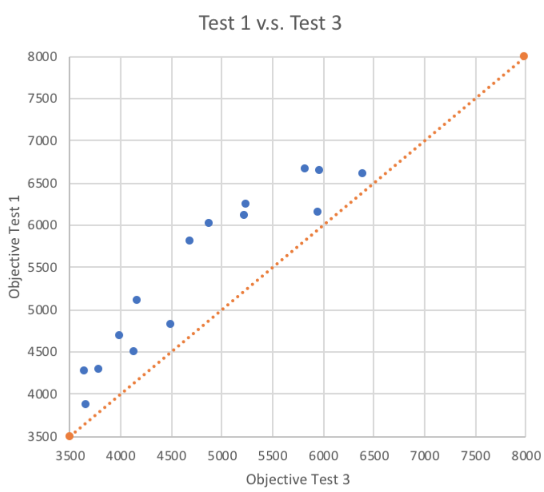

Figure 7: Test 1 v.s. Test 3

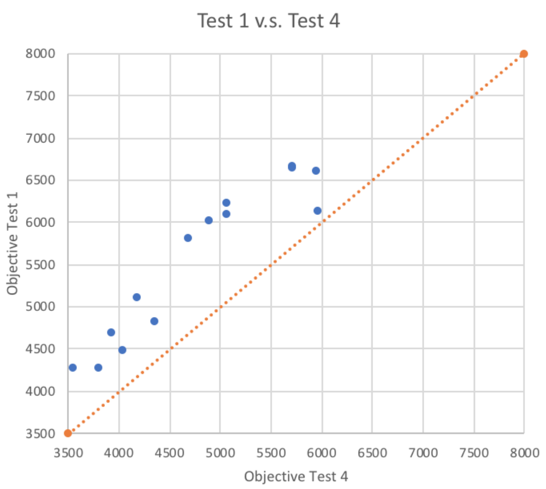

Figure 8: Test 1 v.s. Test 4

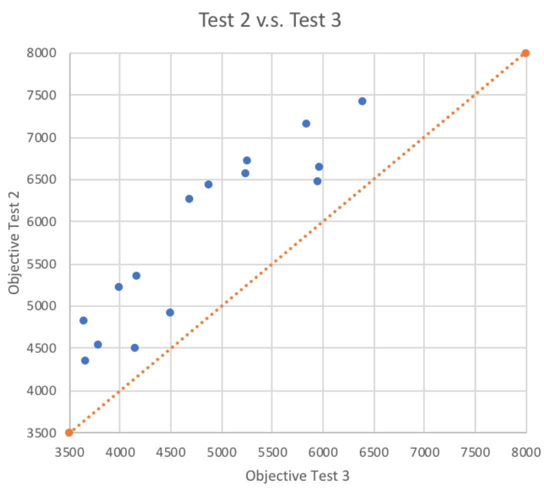

Figure 9: Test 2 v.s. Test 3

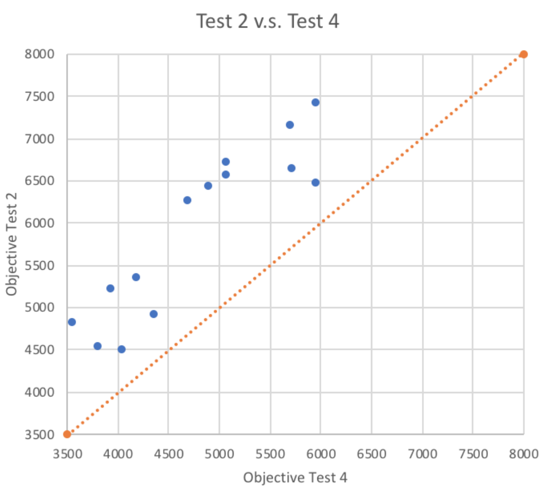

Figure 10: Test 2 v.s. Test 4

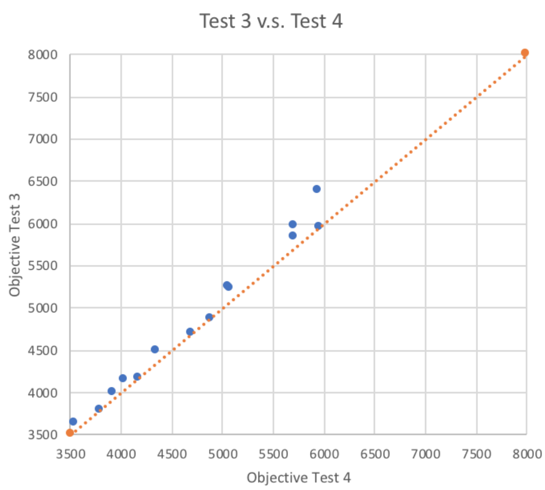

Figure 11: Test 3 v.s. Test 4

8 Conclusion

The main goal of the thesis was to propose insight in how the costs of the decentralized non-information-sharing distribution network can be reduced. The company does not have the possibilities to control the planning from a central perspective. The company re-allocates their commodities based on order size as a threshold value. Researched is whether their fixed value of 0.3 is accurate. Beside that is research what parameters do have an effect on the decision of re-allocation of commodities.

Results are obtained using an exact MIP approach, a combinatorial benders decomposition with a non-exact subproblem and a heuristic approach. The non-exact approaches were used to obtain qualitative good results in less time than the MIP. A conclusion is that the non-exact approaches outperform the exact approaches on computation time and solution quality, certainly when the number of nodes raises and the re-allocation method gets harder to solve. Further on is tested how the first benders iteration performs on quality and calculation time. The results show that for 14 of the 32 cases, the first benders iteration equals the best-found solution. The method generates 9% higher objectives, but in just a fraction of the time that the whole benders algorithm needs. When faster computations are requested, the single benders decomposition would provide qualitative good re-allocations.

The results were obtained with the goal to prove whether the hypotheses are right and to provide insight into the costs of four re-allocation methods. The first tested re-allocation method was no re-allocation at all. Secondly, re-allocation was based on the fixed parameter order size of the company. Third, re-allocation based on decision variable order size was tested. Lastly, re-allocation without restriction on order size had to show the possibilities of cost reduction.

The results of these methods proved all hypotheses to be right for the tested instances, nevertheless with remarks. The conclusions of the hypotheses are explained one by one beneath.

1. Cost reduction could be obtained by more efficient re-allocation of commodities and with that, efficient use of the line haul.

The experiments show cost reduction for no re-allocation (7%), re-allocation based on decision variable order size (17%) and re-allocation without the restriction of order size (19%) in comparison with re-allocation based on the fixed threshold 0.3 loading meter.

2. Cost reduction could be obtained by optimizing the threshold value for the inter-hub exchange of orders.

Based on the results should be concluded that, for the maximum amount of 50 nodes, the most cost-effective way of re-allocation without information sharing is to re-allocate all nodes that are in another service district. All the instances show that when re-allocation based on order size was applied, all orders were re-allocated. This lead to an average decrease in costs of 17% in comparison with the current policy of the company. However, keep in mind that the company deals with a higher amount of nodes, and thus more dense network. It is plausible that this density will have an effect on the costs. A recommendation is to try full re-allocation in practice. Recalculate a planning as it would be a real planning, but now with full re-allocation of commodities. Afterwards, compare the cost of the new method with the current cost of the planning.

3. Re-allocation without parameter thresholds will lead to the most cost-effective networks.

The results showed slightly better objectives (3%) than with re-allocation based on the optimized threshold value. However, it is expected that the difference in costs between the two re-allocation methods will raise further when the number of customers in the network raise. Re-allocation with a known network, viewed from a central perspective can decrease costs up to at least 19%. Recommended is to reconsider the problems for a higher amount of customers if resources and time are available.

References

Afshar-Nadjafi, B., & Afshar-Nadjafi, A. (2017). A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of king saud university-Engineering sciences*, *29*(1), 29–34.

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, *36*(5), 1693–1702.

Berger, S., & Bierwirth, C. (2010). Solutions to the request reassignment problem in collaborative carrier networks. *Transportation Research Part E: Logistics and Transportation Review*, *46*(5), 627–638.

Bettinelli, A., Ceselli, A., & Righini, G. (2011). A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, *19*(5), 723–740.

Blocho, M., & Czech, Z. J. (2013). A parallel memetic algorithm for the vehicle routing problem with time windows. In *P2p, parallel, grid, cloud and internet computing (3pgcic), 2013 eighth international conference on* (pp. 144–151).

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

Clifton, C., Iyer, A., Cho, R., Jiang, W., Kantarcıoğlu, M., & Vaidya, J. (2008). An approach to securely identifying beneficial collaboration in decentralized logistics systems. *Manufacturing & Service Operations Management*, *10*(1), 108–125.

Cruijssen, F., Bräysy, O., Dullaert, W., Fleuren, H., & Salomon, M. (2007). Joint route planning under varying market conditions. *International Journal of Physical Distribution & Logistics Management*, *37*(4), 287–304.

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, *176*(3), 1478–1507.

Dondo, R., Mendez, C. A., & Cerdá, J. (2003). An optimal approach to the multiple-depot heterogeneous vehicle routing problem with time window and capacity constraints. *Latin American applied research*, *33*(2), 129–134.

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.

Gansterer, M., & Hartl, R. F. (2016). Request evaluation strategies for carriers in auction-based collaborations. *OR spectrum*, *38*(1), 3–23.

Gehring, H., & Homberger, J. (2001). A parallel two-phase metaheuristic for routing problems with time-windows. *ASIA PACIFIC JOURNAL OF OPERATIONAL RESEARCH*, *18*(1), 35–48.

Gendreau, M., & Tarantilis, C. D. (2010). *Solving large-scale vehicle routing problems with time windows: The state-of-the-art.* Cirrelt Montreal.

Guajardo, M., & Rönnqvist, M. (2015). Operations research models for coalition structure in collaborative logistics. *European Journal of Operational Research*, *240*(1), 147–159.

Hashimoto, H., & Yagiura, M. (2008). A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. In *European conference on evolutionary computation in combinatorial optimization* (pp. 254–265).

Kalina, P., & Vokrınek, J. (2012). Algorithm for vehicle routing problem with time windows based on agent negotiation. In *Proceedings of the 7th workshop on agents in traffic and transportation, aamas (to appear).*

Kinable, J., & Trick, M. (2014). A logic based benders' approach to the concrete delivery problem. In *International conference on ai and or techniques in constriant programming for combinatorial optimization problems* (pp. 176–192).

Kindervater, G., & Savelsbergh, M. (1997). *Local search in combinatorial optimization.* John Wiley & Sons, Chichester, USA.

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

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

Li, J., Rong, G., & Feng, Y. (2015). Request selection and exchange approach for carrier collaboration based on auction of a single request. *Transportation Research Part E: Logistics and Transportation Review*, *84*, 23–39.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, *23*(5), 377–386.

Miranda, D. M., & Conceição, S. V. (2016). The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, *64*, 104–116.

Nagata, Y. (2007). Edge assembly crossover for the capacitated vehicle routing problem. In *European conference on evolutionary computation in combinatorial optimization* (pp. 142–153).

Nagata, Y., & Bräysy, O. (2009). A powerful route minimization heuristic for the vehicle routing problem with time windows. *Operations Research Letters*, *37*(5), 333–338.

Nagata, Y., Bräysy, O., & Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & operations research*, *37*(4), 724–737.

Nalepa, J., & Blocho, M. (2016). Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. *Soft Computing*, *20*(6), 2309–2327.

Özener, O. Ö., Ergun, Ö., & Savelsbergh, M. (2011). Lane-exchange mechanisms for truckload carrier collaboration. *Transportation Science*, *45*(1), 1–17.

Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, *46*(12), 1433–1446.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817.

Rasmussen, R. V., & Trick, M. A. (2007). A benders approach for the constrained minimum break problem. *European Journal of Operational Research*, *177*(1), 198–213.

Schneider, M. (2016). The vehicle-routing problem with time windows and driver-specific times. *European Journal of Operational Research*, *250*(1), 101–119.

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In *International conference on principles and practice of constraint programming* (pp. 16–30).

van Beek, R. (2018). *Research proposal.*

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2015). Time-window relaxations in vehicle routing heuristics. *Journal of Heuristics*, *21*(3), 329–358.

Wang, X., & Kopfer, H. (2015). Rolling horizon planning for a dynamic collaborative routing problem with full-truckload pickup and delivery requests. *Flexible Services and Manufacturing Journal*, *27*(4), 509–533.