



ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Defining a wood strategy using exact approaches for the Resource Constrained Elementary Shortest Path Problem

MASTER THESIS OPERATIONS RESEARCH & QUANTITATIVE LOGISTICS

20 April, 2021

Author:
RYAN DE REUS (457818)

Supervisor:
SHADI SHARIF AZADEH

Company Supervisor:
ROEL KOOLE

Second Assessor:
WILCO VAN DEN HEUVEL

Abstract

The pick replenishment process is one of the many processes in the distribution centres of Albert Heijn. Optimisation of this process can be achieved by increasing the efficiency of reach truck drivers. When giving advice which location needs to be replenished, the disposal of empty pallets must be taken into account. This thesis will find the optimal wood strategy, which defines when reach truck drivers should drop and sort the stack of empty pallets at a wood location. The sorting time at the wood location will be obtained by performing an Ordinary Least Squares (OLS) regression on the number of pallet types in the stack. Two exact solution methods will be proposed for solving the Resource Constrained Elementary Shortest Path Problem (RCESPP). First, using the Mixed Integer Programming (MIP) formulation and then with a Labelling Algorithm. The proposed methods produce decent results, especially for the labelling algorithm, where the running time remains low. Optimal solutions for the RCESPP, which is a strongly NP-hard problem, are found within less than a second, even for larger instances. After analysing the results, a sensitivity analysis is performed and advice will be given for the wood strategy.

Contents

1	Introduction	1
1.1	Problem Description	3
2	Literature	5
3	Data	8
4	Methodology	10
4.1	Transforming Data	10
4.2	Sorting at the Pick Location	11
4.3	Sorting at the Wood Location	12
4.3.1	Ordinary Least Squares (OLS)	12
4.4	Resource Constrained Elementary Shortest Path Problem (RCESPP)	14
4.5	Labelling Algorithm	16
5	Results	19
5.1	Sorting Model	19
5.2	Resource Constrained Elementary Shortest Path Problem (RCESPP)	24
5.2.1	Mixed Integer Programming (MIP) formulation	27
5.2.2	Labelling Algorithm	28
6	Sensitivity Analysis	30
6.1	Merging two areas	30
6.2	Wood locations at the end of every aisle	30
6.3	(Temporary) wood locations in the middle of every aisle	32
6.4	Strategically adding extra wood locations	33
7	Conclusion	34
8	Discussion	36
	References	37
A	Ordinary Least Squares (OLS)	39
B	Results RCESPP	41
C	Results Additional Wood Locations	45

1 Introduction

Supermarkets are a concept everyone knows nowadays. Grocery shopping has become one of our daily tasks. On average, we make 221 trips to the supermarket every year ([Kantar Worldpanel, 2015](#)). When the Covid-19 pandemic started in March last year, customers started unnecessary panic buying ([Savage and Wall, 2020](#)). Even though there was no need to buy more than we usually would, we started stockpiling due to lockdown fears. We have all seen the anxious customers tearing through the grocery aisles loading toilet paper, hand sanitiser, pasta and long-lasting canned food into piled-high shopping carts. When the supermarket shelves became empty, it was the first time people started to think about the logistic processes behind these supermarket chains. Products always need to be in stock and have to be supplied as fast as possible. This is necessary as there is a lot of competition in the world of retail. Customer satisfaction is a must for these companies in order to become the market leader ([Gomez et al., 2004](#); [Noyan and Simsek, 2011](#)). You don't want your customers to do their groceries somewhere else, do you?

Within the last decades, the role of logistics in a supply chain has arisen ([Neeraja et al., 2014](#)). Logistics are defined as all processes between the production and the consumption/sale of a product. For the transportation part of logistics, lots of research has been done ([Coelho et al., 2016](#)). By using the principles of the Vehicle Routing Problem (VRP) and Travelling Salesman Problem (TSP), progress has been made on cost reduction. However, there are also a lot of logistic processes where costs can still be reduced. For supermarkets, these processes are merely in the distribution centres.

Albert Heijn is a supermarket chain in The Netherlands and Belgium with more than a thousand stores ([Albert Heijn, 2021](#)) and is part of the mother company Ahold Delhaize. For the supply of the supermarkets, Albert Heijn has five distribution centres (DC) in Geldermalsen, Pijnacker, Tilburg, Zaandam and Zwolle. These DCs together supply all stores in The Netherlands and Belgium. For delivery at home, Albert Heijn also has eight Home Shop Centres (HSC) in Amsterdam, Almere, Bleiswijk, De Meern, Eindhoven, Oosterhout, Roosendaal and Rotterdam. To complete the chain, Albert Heijn also uses a Shared Fresh Centre (SFC) in Nieuwegein and some external logistic parties (for the supply of all frozen products or flowers, for example).

Over the years, warehouses have developed from a small role (mostly storage or cross-docking) to a major part of the supply chain ([Lambert et al., 2006](#)). Figure 1 shows the layout of the area for ambient products in DC Pijnacker. The largest part in the centre (marked by 1) is the storage/production area, where all products are stored and eventually picked for orders. At the top (marked by 2), we have the inbound docks and the receiving area, where suppliers deliver their products. At the bottom (marked by 3), we find the shipping area and the outbound docks. Other areas and distribution centres of Albert Heijn have a similar layout.

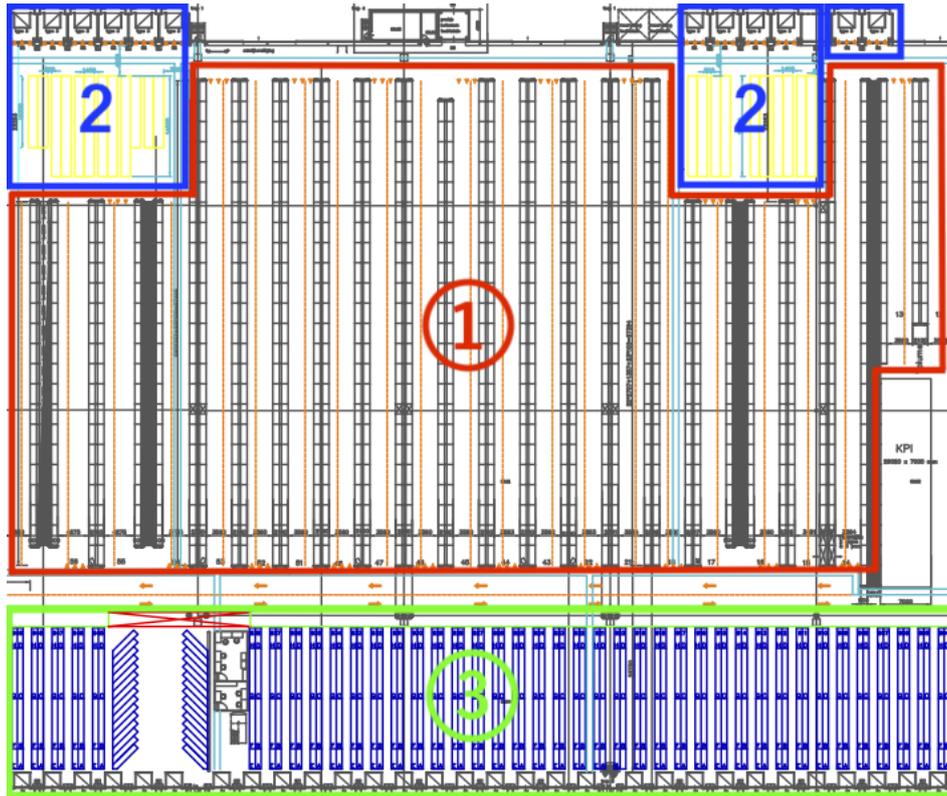


Figure 1: Layout of the ambient area in DC Pijnacker, consisting of a storage/production area (1), receiving area (2) and shipping area (3).

The most important processes in a distribution centre are receiving, put-away, storage, pick replenishment, order picking and shipping (Kusrini et al., 2018). In the distribution centres of Albert Heijn, the storage area consists of two parts. First, we have the products on the ground, the pick locations, where store orders are collected. Next to that, we have the products in the racks above, these are called buffer locations and are extra stock for the pick locations. Figure 2 shows an aisle in DC Zaandam and its pick and buffer locations.



Figure 2: Aisle in the storage area of DC Zaandam.

When an inbound truck arrives at the distribution centre, products will be unloaded, checked and temporarily stored in the receiving area (Figure 3a) close to the dock. This is the process we call receiving. Products arrive on pallets for handling and storage efficiencies. In the put-away process, reach trucks will move these pallets from the receiving area to a storage location in the racks, where this buffer location should always be as close as possible to the pick location. Order picking is the process where order pickers collect the products at the pick location and put them on a roller container. Each roller container is part of a store order. When the roller containers are full, they will be moved to the shipping area (Figure 3b), where the containers are loaded into the outbound trucks.



(a) Receiving area in DC Tilburg

(b) Shipping area in DC Geldermalsen

Figure 3: Receiving and shipping areas in a warehouse of Albert Heijn.

Over time, the pick locations are becoming emptier because of the order picking, so products have to be replenished. This is the pick replenishment process, where reach truck drivers are responsible for replenishing the (almost) empty pick locations. After removing the empty pallet from the pick location, a full pallet must be lowered from the buffer location (storage location in the racks above) to the ground and placed into the pick location. If there are any products left on the almost empty pallet, these products need to be stacked onto the new full pallet. After that, the old pallet needs to be dropped at a wood location, where all empty pallets are sorted by type.

1.1 Problem Description

In the current situation, reach trucks are driving around in the pick zones to check for (almost) empty pick locations. Driving around and searching for (almost) empty pallets takes a lot of time. Furthermore, choosing the wrong pick replenishment order can lead to a downtime of the order pickers arriving at another empty pick location. Hence, we can conclude that the current process, replenishment on sight, is very inefficient. A new algorithm is being developed, which will give the reach truck drivers advice about the next location that needs to be replenished. This advice will be based on the locations of all reach trucks, information about orders that are being picked, knowledge about picking times and some other data sources in the warehouse management system.

One unknown variable in this process is the disposal of empty pallets. After every replenishment, the reach truck driver is left with an empty pallet. These pallets must be collected and dropped at a specific location, where the pallets need to be sorted by type. These locations are called wood locations and are usually in a cross-aisle or at the rear end of the storage/production area.

Decisions have to be made, whether the empty pallet will be dropped at the wood location directly after the pick replenishment order or later after some other replenishments, where the reach truck carries the empty pallet(s) to the new replenishment location. Our main research question will be about this wood strategy, where we have to determine the optimal moment for a reach truck driver to visit the wood location and drop and sort the stack of empty pallets. This optimal moment will not be a specific time during the day, but rather something like a thumb rule or strategy. For example, when the next replenishment is further than a certain threshold, after a fixed amount of replenishments, or when the stack has a certain amount of pallet types.

In order to give the best advice for the next replenishment order, the algorithm must take into account the disposal of these empty pallets. Therefore, the wood strategy is very important in this process. First, the current method of disposal (wood strategy) must be determined. Then, an optimal wood strategy must be developed for including the disposal of empty pallets in this new algorithm. This must be done in such a way that the reach truck driver can do his/her work as efficiently as possible, including the disposal of empty pallets, where the advice fits seamlessly into practice and safety is guaranteed. This optimal wood strategy can then be compared to the current strategy. Finally, the wood strategy needs to be implemented in the new algorithm that will increase the overall efficiency of reach truck drivers.

Even though there is a rich body of literature on the optimisation of warehouse processes, there is not much written about the pick replenishment process. For the wood strategy specifically, or any other processes concerning the disposal of empty pallets in a warehouse in general, we see that this kind of problems has never been addressed in literature before. Together with the fact that we use the available data at Albert Heijn for a wood strategy that results in an increase in the overall efficiency for reach trucks, we think this paper might be of great scientific relevance. We will split the pick replenishment process and only consider the parts that involve the choice of dropping empty pallets at the wood location. Subsequently, we will transform the wood strategy problem into a Resource Constrained Elementary Shortest Path Problem, to find the optimal moment when a reach truck driver should visit a wood location. Since this will result in the optimal solution for a given day and a given reach truck driver, we will do some further analysis to come up with advice for a wood strategy that can be used in real practice.

The remainder of this paper is structured as follows. Section 2 provides an overview of the most important literature relevant for this thesis. Section 3 describes what data we use and how we will gather it. In Section 4, we will present the appropriate methods to solve the Resource Constrained Elementary Shortest Path Problem (RCESPP). We will solve the RCESPP using two exact approaches, first with the Mixed Integer Programming formulation and then with a labelling algorithm. Section 5 describes the results. Thereafter, we will perform a sensitivity analysis in Section 6. In Section 7, we will draw a conclusion. Finally, we will reflect upon our research, outline some limitations and discuss recommendations for future research in Section 8.

2 Literature

Although there is written a lot of literature about logistics in general, only a small part concerns the logistic processes in a distribution centre, which is the scope of this thesis. Optimisation of these warehouses can be achieved in many ways.

One of the processes where optimisation can result in a large cost reduction is order picking. The order picking costs are around 55% of all warehouse operation costs ([Tompkins et al., 2010](#)). Before an order can be picked, it has to be decided which orders are put together on a roller container (load carrier) and which load carriers are picked by which order picker. Albert Heijn has introduced the Load Carrier Algorithm (LCA) and the Pick Order Build (POB) algorithm to optimise this process. LCA takes a store order and decides which products need to be placed on which load carrier, where certain restrictions apply such as the balance between heavy/light and small/large products and the affinity with other products (for efficiency in the stores).

Once the order per load carrier is generated, the POB creates combinations of load carriers, which are translated to pick orders. Each pick order consists of 5 load carriers. Order pickers then start with these orders at one point in the picking circuit, making S-shapes through the aisles, as all aisles can be traversed in only one direction because of the small aisle width and high pick density. [Hall \(1993\)](#) also shows that traversal routing (traversing the entire aisle when we choose to enter the aisle) is widely used in warehouses with high pick density (such as those of Albert Heijn), as this policy is very simple and gives good results.

As mentioned by [De Koster and Van Der Poort \(1998\)](#), most articles about operations within a distribution centre (DC) are based on an automatic storage/retrieval system (AS/RS). Even though Albert Heijn is currently working on a project to automate the majority of the processes for the ambient products in DC Zaandam, manual processes are still used for other distribution centres and the area for fresh products in DC Zaandam. Hence, the focus of this paper is on manual storage, which means there is only a small number of papers that might be relevant.

Pick locations become empty because of order picking. The pick replenishment process is there to avoid empty pick locations by replenishing the stock from the buffer locations (storage locations in the racks above the pick location). Unfortunately, there is only few relevant literature about pick replenishment. One of these few relevant papers, [Gagliardi et al. \(2008\)](#), analyses the order picking and stock replenishment operations in a high-throughput distribution centre. Their goal is to optimise the replenishment system, in order to minimise possible stock-outs. The methods and operations they are using are very similar to those of Albert Heijn. For example, the storage and pick locations are separated, order pickers have to wait when reach truck drivers did not replenish the pick location in time, batch picking is used for the order pickers, etc.

Furthermore, the warehouse management system currently in use in their research does not synchronise the operations of the reach trucks with those of the order pickers. Instead, the reach truck driver inspects the slots visually and brings a new pallet of the product that seems to need replenishing. This is exactly the same as the ‘replenishment on sight’ system that is currently used at Albert Heijn. They come up with some replenishment heuristics. A combination of their third and fourth heuristic will be used in the new algorithm that is being developed that will increase the overall efficiency of the reach trucks. Information about the inventory level of products that will be picked within a certain amount of time from now is used to make a choice which pick locations needs to be replenished first. As [Gagliardi et al. \(2008\)](#) also mentions, this kind of replenishment problems has never been addressed in literature before.

Within the pick replenishment process, we have to come up with a solution to deal with the empty pallets resulting from a pick replenishment, a so-called wood strategy. In this wood strategy, we have to decide when empty pallets should be dropped at the wood location. To the best of our knowledge, this issue has never been addressed in literature before. This is also one of the reasons why this paper is relevant and contributes to the existing literature.

Finding the optimal wood strategy will be achieved by formulating our problem as a Resource Constrained Elementary Shortest Path Problem (RCESPP) in Section 4.4. The RCESPP is a strongly NP-hard problem ([Dror, 1994](#)), mostly found in the context of column generation for Vehicle Routing Problems (VRP) and often solved with a dynamic programming algorithms ([Lozano et al., 2015](#)). Given our resources and their constraints, our mathematical formulation is very similar to a Shortest Path Problem with Time Windows (SPPTW), except for the fact that we do not allow waiting time (as our ‘time resource’ will actually be the number of pallets). The SPPTW was first studied in [Desrosiers et al. \(1984\)](#) and is later generalised to the Resource Constrained Shortest Path Problem (RCSPP) with two resources. Therefore, we will use the literature of the RCESPP, the RCSPP and the SPPTW. [Pugliese and Guerriero \(2013\)](#) highlights the similarity and the differences between these problems and presents a survey of exact solution approaches, which is used for inspiration of our methods.

We will first consider the Resource Constrained Shortest Path Problem (RCSPP), which is a special case of the Shortest Path Problem where some additional constraints apply. The first introduction of the RCSPP is in [Desrochers \(1987\)](#), where the goal is to find the shortest path among all paths from the source to the sink, satisfying a set of constraints on the resource variables. Resources can be time, quantities, weight, etc. These resources all have a resource window at each node, which leads to the constraints. The upper and lower bound of this resource window can be different for each resource and each node. [Irnich and Desaulniers \(2005\)](#) discusses the different variants of the RCSPP, some modelling issues and the most common solution methods. One of these solution methods, dynamic programming with labelling algorithms will be used to solve our problem, which we will describe in Section 4.5.

Labelling algorithms can be seen as an improvement of other methods (such as dynamic programming), since they follow the line of Pareto-optimal approaches and only consider non-dominated paths. Dominated paths are by definition less efficient in the costs and resource consumption and therefore not considered (Ziegelmann, 2001). For the SPPTW, Desrochers and Soumis (1988) presents an efficient labelling algorithm that runs in pseudo-polynomial time. In Joachim et al. (1998) a new dynamic programming algorithm will be developed for the SPPTW with linear node costs. Since our node costs depend on the resource consumption and are therefore not constant either, this paper might be of interest for the implementation of the node costs.

As elaborated by Ahuja et al. (1993) and Ziegelmann (2001), there are two different types of labelling algorithms, the label setting and the label correcting algorithm. In the label setting algorithm, paths are extended in topological order and after each iteration one label is marked as permanent (optimal). A well-known label setting algorithm is Dijkstra's algorithm, introduced in Dijkstra et al. (1959). Most papers only consider graphs with nonnegative resource consumption on the arcs (Feillet et al., 2004). However, after visiting a wood location, the number of pallets equals zero. Hence, we have to 'reset' the resource variable for the number of pallets and we consequently have a negative resource consumption. Unfortunately, label setting algorithms require at least one resource to be nonnegative on all arcs and are therefore not useful in our problem. Nevertheless, we might use the idea and algorithms to improve our labelling algorithm.

In the label correcting algorithm, we use a list of nodes, take the first node and explore all other nodes by their outgoing arcs. For every non-dominated path, we create a label and extend all these possible labels until we reach the final node or the path becomes infeasible by one of the resource constraints (Ziegelmann, 2001). The existence of a negative resource consumption on arcs means we cannot treat paths in a strictly increasing order of their resource vectors (Irnich and Desaulniers, 2005). Therefore, all labels in the label correcting algorithm are temporary until the final node. The Bellman-Ford algorithm, introduced in Bellman (1957), is a well-known label correcting algorithm, which extends the first label in the queue and adds the newly generated labels back at the end of this queue.

Feillet et al. (2004) presents an algorithm based on the Bellman-Ford algorithm and Beasley and Christofides (1989), that can solve the RCSPP with negative resource consumption. Since our cost function partly depends on resource consumption, negative cost cycles may appear in our graph. We will avoid these negative cost cycles by adding the restriction that a path can traverse all arcs and visit all nodes at most once. This means our optimisation problem becomes a Resource Constrained Elementary Shortest Path Problem (RCESPP). The label correcting algorithm from Feillet et al. (2004) will therefore be adapted, so we can solve the RCESPP and find the optimal wood strategy.

3 Data

Optimising the wood strategy requires a lot of data. First of all, we need to know all the ins and outs of the pick replenishment process and knowledge about all other actions reach truck drivers perform throughout the day. Next to that, we need to know how other processes, such as the put-away and order picking process, are related to the pick replenishment process. Due to the fact that our research has many practical applications and will be used in practice, it is of utmost importance to fully comprehend the processes it may have an impact on. This kind of data/knowledge will be gathered by some explanatory sessions given by employees of the logistics support team, team leaders of the distribution centres and the reach truck drivers themselves.

Once we have a clear idea of all the processes related to this study, we proceed with gathering data for the pick replenishment process itself. For almost every action in the distribution centre, from receiving goods and store orders to coordinates of locations and shipping goods, data will be registered in the Warehouse Management System (WMS), which will be available for this research. However, after receiving the goods, the focus is on the product itself and not the pallet underneath it. Therefore, after replenishing the product at the pick location, the empty pallet we have left is nowhere to be found in the data. Also, while dropping empty pallets at the so-called wood locations, no label will be scanned (like for the other actions in the pick replenishment process). Hence, there is no data available for the wood strategy, so we need to collect this data ourselves.

We need to know information about the current strategy or procedures reach truck drivers follow when they are dropping their empty pallets at the wood location. A strategy here might be something of a thumb rule, for example:

- Drop the empty pallets once we have a stack of a certain amount of pallets.
- Drop the empty pallets once we are close to a wood location.
- Drop the empty pallets when the distance to the next replenishment location is more than a certain threshold.
- Drop the empty pallet after each replenishment.

For gathering data about the current strategy and procedures, we will ask some reach truck drivers in the distribution centres of Albert Heijn.

Calculating distances between the locations of pick replenishment orders can be done easily by taking the coordinates of all locations and aisle entry/exit points in the Warehouse Management System (WMS). However, the coordinates of the wood locations are not registered in WMS and we need to visit all distribution centres to see where these wood locations are located and to register their coordinates.

WMS can also provide us with some constants such as the driving speed and fixed driving drive of a reach truck, so we can calculate the exact driving time between locations. Determining when the reach truck driver should drop the empty pallets depends on all pick replenishment moves. Therefore, we will need to know all pick replenishment orders a reach truck driver has done during his shift. For all these pick replenishment orders, we also need to know the date and time, the DC, the area, the pick and buffer locations and their coordinates, the reach truck id, the terminal id, the article id and the type of pallet. This information will all be extracted from multiple databases in WMS.

Since the suppliers of Albert Heijn use different types of pallets, empty pallets need to be sorted at the wood locations. Therefore, we cannot assume that every stack of empty pallets takes the same amount of time to drop at the wood location. Hence, we need to come up with a model for sorting a stack of pallets. Section 4 will give more details and explains the models we will use to solve this subproblem. For these models, we need some additional data, namely the time it takes for certain moves with a reach truck and the time it takes to sort a stack of empty pallets. For every stack, we also need to know how many pallets it consists of and how many different pallet types the stack has. This data will be conducted in the distribution centres by timing with a stopwatch, as these sorting actions are also not visible in the WMS databases.

As mentioned earlier, Albert Heijn has distribution centres in Zaandam, Tilburg, Pijnacker, Zwolle and Geldermalsen. Since the layouts of these distribution centres are all different, we need to visit all distribution centres, to gather the coordinates of the wood locations. These visits will then be combined with the conversations with reach truck drivers as mentioned before. For the timings of the sorting tasks, we will plan another visit to the distribution centres in Pijnacker, Geldermalsen and Tilburg.

4 Methodology

This section describes the different methodologies we develop to solve our problem: finding the optimal wood strategy. We will first transform the data from Section 3 so we can use it as input in our models and algorithms. This will be discussed in Section 4.1. For all methods, the costs are defined as the time it takes to perform all processes related to empty pallets. Therefore, it is important to know how much time it takes for a given stack of empty pallets to sort this stack at the pick location (in the aisle). We will discuss this further in Section 4.2. Furthermore, we also need to know how much time it takes to drop these empty pallets in the right place at the wood location. Therefore, Section 4.3 will show how we can determine this time component by creating a sorting model for the wood location. In Section 4.4, we will formulate a Resource Constrained Elementary Shortest Path Problem (RCESPP) that can find the optimal wood strategy. The RCESPP will then be solved using both the Mixed Integer Programming (MIP) formulation and a labelling algorithm, as explained by Section 4.5.

4.1 Transforming Data

As shown in Section 3, we have collected lots of data. Unfortunately, this data cannot be used directly as input for our models and algorithms. We have to link and combine the raw data and filter out unnecessary columns. Once this is done, we have three types of datasets.

1. Layout: For each DC we have a file containing the layout (locations, aisles, areas). Each location consists of a unique name, the aisle number and the X, Y and Z coordinates. Each aisle consists of a unique aisle number, the length, the coordinates of the entry and exit points and the area where the aisle is located. Each area consists of a unique identifier, its specific features (e.g. temperature, type of products) and some base coordinates (that need to be added to the locations/aisles).
2. Wood Locations: For each DC and each area we have a file containing the coordinates of the wood locations.
3. Pick Replenishment Orders: For each reach truck, we have a file containing all pick replenishment orders fulfilled by that specific reach truck (driver) on a given day. This will also be referred to as an individual dataset or instance.

An individual dataset contains the following information.

- Identity of the reach truck.
- Area(s) where the pick replenishment orders are located.
- Date and time of all pick replenishment orders.
- Buffer (storage) and pick locations of all pick replenishment orders.
- Pallet types of all replenished products.

Next, only some of the individual datasets will be selected, namely those with a minimum of 20 and a maximum of 40 pick replenishment orders. Datasets with fewer pick replenishment orders do not give enough useful information in the analysis of the results. Datasets with more than 40 pick replenishment orders will (generally) result in a running time that exceeds the time limit of 1 hour when we use our first solving method (MIP formulation). The second solving method (labelling algorithm) can solve these instances within a second. However, we want to compare the results and are therefore restricted by the solving time of the MIP formulation.

4.2 Sorting at the Pick Location

Before starting a pick replenishment order, the reach truck driver may already carry a stack of empty pallets. This stack will be placed in the aisle, close to the pick location that needs to be replenished. Then, the (almost) empty pallet in the pick location will be removed and also placed in the aisle somewhere close to the pick location. After this, the new full pallet will be collected at the buffer (storage) location and put into the pick location. The products left on the old pallet (if any) then need to be stacked on top of the new pallet. Finally, the empty pallet needs to be added to the stack.

Depending on the composition of the stack, there are different scenarios for correctly putting the new empty pallet into the stack. We will split this into 4 different cases.

1. New pallet type (Figure 4a)
2. Pallet type is already in the stack
 - a) Stack has at most 2 pallet types (Figure 4b)
 - b) Stack has more than 2 pallet types, new pallet type at top/bottom (Figure 4c)
 - c) Stack has more than 2 pallet types, new pallet type not at top/bottom (Figure 4d)

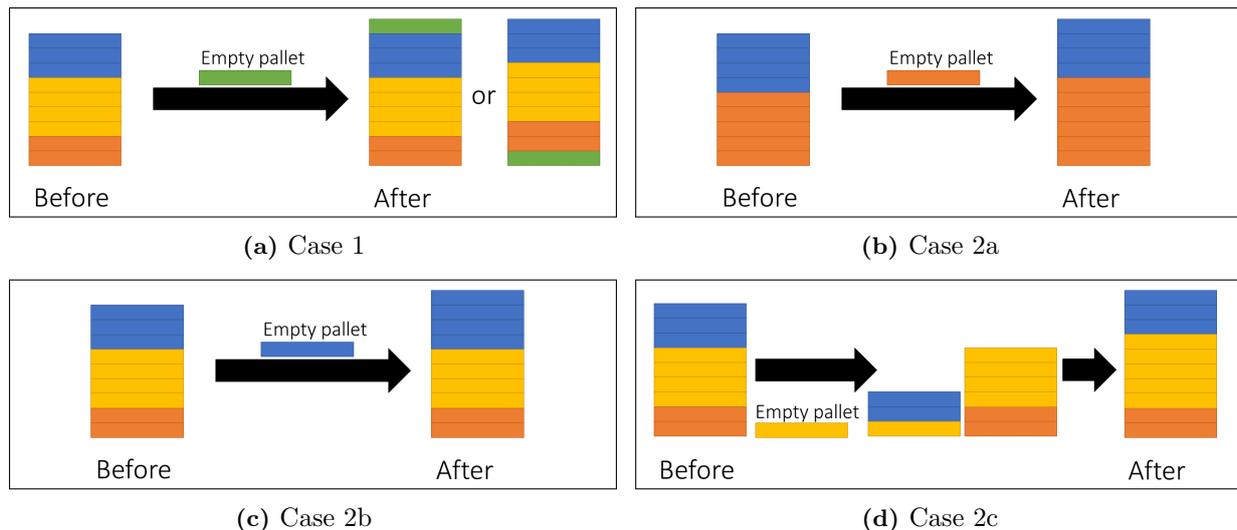


Figure 4: Different cases of putting an empty pallet in the stack.

For cases 1, 2a and 2b, a reach truck driver has to make 1 move. For case 2c, it takes 2 moves. A move is defined as the process of putting the forks underneath a (stack of) pallet(s), moving it somewhere else and pulling the forks backs. Figure 4 shows a visualisation of all cases.

4.3 Sorting at the Wood Location

Whenever a pick replenishment order is finished, a reach truck driver has a choice whether he/she wants to drop the pallets at the wood location or continues with the next replenishment (and thus carrying the stack of empty pallets to the next location). When a reach truck driver arrives at the wood location, we assume that the stack of empty pallets is sorted by type, as explained by Section 4.2. At the wood location, every pallet type has its own stack. Figure 5 shows some wood locations in DC Tilburg and DC Pijnacker. The reach truck driver has to move all pallets from the (mixed) stack on the reach truck to the right stack at the wood location. The time it takes to complete this sorting process depends on the number of pallet types in the mixed stack and probably some other factors.



(a) DC Tilburg (DCT)



(b) DC Pijnacker (DCP)

Figure 5: Wood locations and pallet aligners.

4.3.1 Ordinary Least Squares (OLS)

Since the time for sorting the (mixed) stack at the wood location may depend on multiple factors, we will create a regression model to investigate which factors/variables might be explanatory and which formula would most closely fit the data. We have looked at the differences between the wood locations across all distribution centres and then decided which relevant factors (differences between observations, areas or distribution centres) will be included in our regression model. Therefore, we will first perform a regression analysis to determine the linear relation between the time and the number of pallet types, as we know this is the most important factor. Next, we will also check whether the total number of pallets, the area, the distribution centre or a pallet aligner (as shown in Figure 5b for the red LPR pallets and the grey plastic DPB pallets) have a significant effect on the time as well. Sometimes, the geometry of the area of distribution centre is also considered as a factor, however, the areas and the layout of the distribution are all similar, so we will not test for this factor. We will perform this regression by using Ordinary Least Squares (OLS).

Suppose we want to describe our model by $y = \alpha + \beta x + \epsilon$, where α is the intercept, β the coefficient vector of our variables and ϵ the residual error. Let (1) be the system of linear equations, where n is the number of observations and p the number of dependent variables + 1 (as our α is β_1 and therefore the first column of X are all ones).

$$y_i = \sum_{j=1}^p \beta_j X_{ij} \quad i = 1, \dots, n \quad (1)$$

We will continue with the matrix form $y = \beta X$, where

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

We will then estimate parameter β by minimising Equation (2).

$$S(\beta) = \sum_{i=1}^n \left| y_i - \sum_{j=1}^p X_{ij} \beta_j \right|^2 = \|y - X\beta\|^2 \quad (2)$$

As $S(\beta)$ is quadratic in β , with a positive-definite Hessian, we will get a unique global minimum for $\hat{\beta} = \arg \min_{\beta} \|y - X\beta\| = (X^T X)^{-1} X^T y$ (Hutcheson and Sofroniou, 1999).

In our regression analysis, we will look at the R square (R^2), Adjusted R square (\bar{R}^2), standard errors ($\hat{\sigma}_j$) for all variables j , p -value and t -statistic to see which model has the best fit and which variables have a significant effect at the sorting time. These outputs will be calculated as follows.

$$\hat{\sigma}_j = \left(\hat{\sigma}^2 [X^T X]_{jj} \right)^{1/2} \quad (3)$$

$$t = \hat{\beta}_j / \hat{\sigma}_j \quad (4)$$

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad (5)$$

$$\bar{R}^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} \quad (6)$$

Additionally, we will perform an one-way analysis of variance (ANOVA) on the differences between the regression and residuals, so we can compare the F-statistic of the different OLS regression models.

4.4 Resource Constrained Elementary Shortest Path Problem (RCESPP)

Solving our wood strategy, i.e. determining at what moment a reach truck driver should drop the stack of empty pallets at the wood location, requires a mathematical model. This model could then be solved using a commercial solver. The mathematical model we use is a Resource Constrained Elementary Shortest Path Problem (RCESPP). Our RCESPP is based on a graph, where the nodes represent the pick replenishment orders and the wood locations and the arcs represent the driving time (costs) and information flow between the nodes. Figure 6 shows an example of such a graph with 8 pick replenishment orders and 2 wood locations. Additionally, we define some other costs at the nodes. For the pick replenishment nodes, we use the sorting time at the pick location. For the wood location nodes, we use the sorting time at the wood location. All other costs such as the stacking time for the remaining products and the driving time to the buffer location, will not be used, as these costs do not affect the choices of dropping wood.

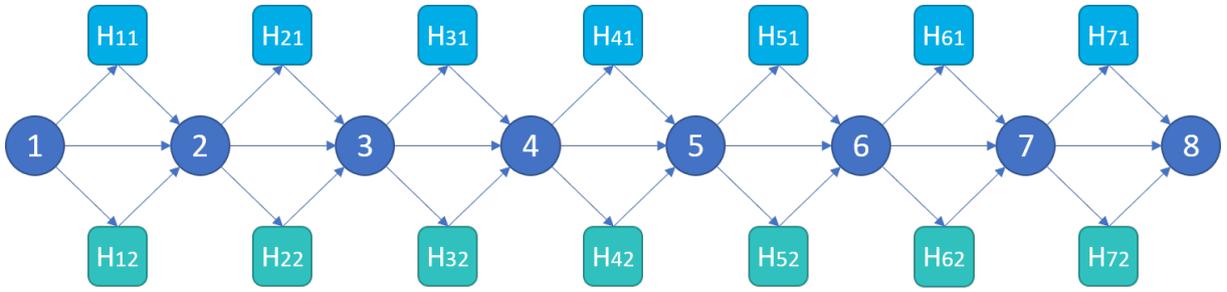


Figure 6: Example of the RCESPP graph with 8 pick replenishment orders and 2 wood locations.

For correctly calculating the sorting times, we keep track of the number of pallets for each pallet type, which are the resources of our RCESPP. These resources are constrained by a lower and upper bound, which may vary for each node. The information flow on the arcs is defined as the number of pallets (for each pallet type specifically) that need to be added in order to correctly calculate the resources. When visiting a wood location, the resource vectors need to be reset, as all pallets are dropped at the wood location and the stack is empty. Hence, we have a negative flow on the arcs and we may encounter cycles in our graph. Since our goal is to find the shortest path that minimises the total costs, satisfies all resource constraints and visits nodes and arcs at most once, we will use as a Resource Constrained Elementary Shortest Path Problem (RCESPP) to model our optimisation problem. We will first introduce our sets, decision variables and parameters.

Sets

N	Set of nodes (replenishments and wood locations, index i, j and k).
WL	Set of wood locations, $WL \subset N$.
A	Set of arcs (index (i, j)).
PT	Set of pallet types (index t).

Variables

x_{ij}	Decision variable, is 1 if we use the arc from node i to j , 0 otherwise.
p_{ti}	Number of pallets of type t at node i .
q_i	Total number of pallets in the stack at node i .
c_i	Number of different pallet types at node i .

Parameters

w_{ij}	Driving time between node i and j .
p_{tij}	Resource consumption variable, is 1 if the replenishment at node j has an empty pallet of type t , is -20 if node j is a wood location and 0 otherwise.
a_i and a_{ti}	Lower bound of the resource vector at node i .
b_i and b_{ti}	Upper bound of the resource vector at node i .

The MIP formulation of our Resource Constrained Elementary Shortest Path Problem (RCESPP) is as follows.

$$\min \sum_{(i,j) \in A} w_{ij} x_{ij} + \sum_{i \in N} f_i(c_i) \quad (7)$$

$$\sum_{k \in N} x_{ik} - \sum_{k \in N} x_{ki} = \begin{cases} 1 & \text{for } i = S \\ -1 & \text{for } i = T \\ 0 & \text{otherwise} \end{cases} \quad i \in N \quad (8)$$

$$p_{ti} + \sum_{(i,k) \in \delta^+(i)} p_{tik} x_{ik} \leq p_{tj} \quad t \in PT, (i, j) \in A \quad (9)$$

$$q_i = \sum_{t \in PT} p_{ti} \quad i \in N \quad (10)$$

$$c_i = \sum_{t \in PT} \mathbb{1}_{\{p_{ti} \geq 1\}} \quad t \in PT, i \in N \quad (11)$$

$$a_{ti} \leq p_{ti} \leq b_{ti} \quad t \in PT, i \in N \quad (12)$$

$$a_i \leq q_i \leq b_i \quad i \in N \quad (13)$$

$$x_{ij} \in \mathbb{B} \quad i, j \in N \quad (14)$$

$$p_{ti} \in \mathbb{N} \quad t \in PT, i \in N \quad (15)$$

$$c_i, q_i \in \mathbb{N} \quad i \in N \quad (16)$$

Equation (7) is the objective function that minimises the total costs. The first part minimises the arc costs, which is defined as the driving time. The second part minimises the node costs, which is the sorting time. As we have defined different sorting times for the pick replenishment nodes and the wood location nodes, we will implement the node costs using function f , which is described in Algorithm 1. Constraint (8) is our flow conservation constraint. Except for the source and the sink, the net flow at each node must equal 0, which means that the number of incoming arcs is equal to the number of outgoing arcs. Constraint (9) defines the resource consumption flow to ensure the correct number of pallets at each node. We sum over $\delta^+(i)$, the set of outgoing arcs of node i . When we visit wood location $k \in WL$, p_{tik} takes value -20 in order to reset the number of pallets. In Equation (10) we calculate the total number of pallets at each node. Equation (11) calculates the total number of pallet types. This is done by summing over an indicator function that takes value 1 if pallet that pallet type is present in the stack and 0 otherwise. Constraints (12) and (13) define the upper and lower bounds of the number of pallets at each node. Finally, Constraints (14) - (16) define the scope of our variables.

Algorithm 1 Function f in the RCESPP

```

1: function  $f_i(c_i)$ 
2:   if  $i \in WL$  then                                     ▷ Node  $i$  is a wood location
3:      $f_i(c_i) \leftarrow 22.79c_i - 8.63$ 
4:   else                                                   ▷ Node  $i$  is a pick replenishment
5:     Introduce variable  $m$  for the number of required moves.
6:     Get the type of the empty pallet at this location.
7:     if pallet type is already in the stack then
8:       if  $c_i \leq 2$  then
9:          $m \leftarrow 1$ 
10:      else
11:         $m \leftarrow 2 - \frac{2}{c_i}$                        ▷ Expected value of the number of moves
12:      end if
13:    else                                                 ▷ New pallet type
14:       $m \leftarrow 1$ 
15:    end if
16:     $f_i(c_i) \leftarrow 12.76 \cdot m$                    ▷ Each move takes 12.76 seconds
17:  end if
18:  Return  $f_i(c_i)$ 
19: end function

```

4.5 Labelling Algorithm

As shown in Section 2, there is a lot of literature written about algorithms that can solve the Shortest Path Problem, either with or without resource constraints. However, there is only one paper that allows for negative costs/resources on arcs. Therefore, we will use the label correcting algorithm from Feillet et al. (2004) and adapt this to our optimisation problem.

The graph, cost function, notation and formulation of Section 4.4 will be used here as well. Additionally, we need some extra notation for the labels. Every node in our graph gets a label in the algorithm, which indicates the cost and resource consumption of the partial path from the source to that node. This will be done iteratively until we reach the sink and no more labels can be created. For each node, we will extend its label(s) to every other node that is connected via an outgoing arc and is feasible in terms of resource constraints.

Suppose we are at node $i \in N$. Then we have path X_{si} from the source (s) to node i . At this node, we create a label $(p_i^t, q_i, c_i, s_i, v_i^k, C_i)$, where p_i^t is the resource consumption vector for all pallet types $t \in PT$, q_i the total number of pallets, c_i the number of pallet types, s_i the number of unreachable nodes, v_i^k the vector of unreachable nodes (where v_i^k equals 1 if node $k \in N$ is unreachable and 0 otherwise) and finally C_i the cost of path X_{si} . A node is called unreachable if it cannot be attained directly via an outgoing arc, if it is already in our path X_{si} or if there exists a resource that exceeds the resource bounds b_{ti} or b_i . Now we only need to consider non-dominated paths, as proven by [Feillet et al. \(2004\)](#).

The domination rules used in the labelling algorithm are as follows. Let X_{si}' and X_{si}^* be two distinct paths from the source s to node $i \in N$. Their associated labels are $(p_i^{t'}, q_i', c_i', s_i', v_i^{k'}, C_i')$ and $(p_i^{t*}, q_i^*, c_i^*, s_i^*, v_i^{k*}, C_i^*)$, respectively. Then X_{pi}' dominates X_{pi}^* if and only if $C_i' \leq C_i^*$, $s_i' \leq s_i^*$, $p_i^{t'} \leq p_i^{t*} \forall t \in PT$, $q_i' \leq q_i^*$, $t_i' \leq t_i^*$, $v_i^{k'} \leq v_i^{k*} \forall k \in N$ and the labels are not equal.

Finally, the following notation is required to explain the algorithm.

Λ_i	List of labels at node i .
S_i	Set of successors of node i .
U	List of unprocessed nodes.
$\text{Extend}(\lambda_i, j)$	Function that returns the label following the extension of label $\lambda_i \in \Lambda_i$ towards node j when feasible, nothing otherwise. If node j is a wood location, the pallet resources will be reset to 0. The resources are updated first and if the resource constraints are satisfied, it looks into the set of outgoing arcs to update the number and vector of unreachable nodes.
E_{ij}	List of labels extended from node i to j .
$\text{EFF}(\Lambda)$	Procedure that keeps only non-dominated labels in Λ .

Now that we have explained the dominance rules and the notation for both the algorithm and the labels, we can present the labelling algorithm in Algorithm 2.

Algorithm 2 Labelling Algorithm

```
1: Initialisation
2:  $\Lambda_s \leftarrow \{(0, \dots, 0)\}$ 
3: for all  $i \in N \setminus \{s\}$  do
4:    $\Lambda_i \leftarrow \emptyset$ 
5: end for
6:  $U = \{s\}$ 
7:
8: while  $U \neq \emptyset$  do
9:   Choose  $i \in U$ 
10:  for all  $v_j \in S_i$  do
11:     $E_{ij} \leftarrow \emptyset$ 
12:    for all  $\lambda_i \in \Lambda_i$  do
13:      if  $v_i^j = 0$  then
14:         $E_{ij} \leftarrow E_{ij} \cup \{\text{Extend}(\lambda_i, v_j)\}$ 
15:      end if
16:    end for
17:     $\Lambda_j \leftarrow \text{EFF}(E_{ij} \cup \Lambda_j)$ 
18:    if  $\Lambda_j$  has changed then
19:       $U \leftarrow U \cup \{v_j\}$ 
20:    end if
21:  end for
22:   $U \leftarrow U \setminus \{v_i\}$ 
23: end while
24: Return  $\Lambda = (\Lambda_s, \Lambda_1, \dots, \Lambda_t)$ .
```

The labelling algorithm in Algorithm 2 starts at the source node (s). For initialisation, we create a label where all costs and resources equal zero and only the first node is reachable. The label will be added to the list of labels on the source node and the source node will be put in the list of unprocessed nodes, so we can start the recursion. When there are no unprocessed nodes left, the algorithm returns all non-dominated paths from the source to every other node. For solving the Resource Constrained Elementary Shortest Path Problem (RCESPP), we only need to look at Λ_t (the list of all non-dominated labels at the sink) and take the path/label with the lowest total costs.

The time complexity of Algorithm 2 strongly depends on the structure of the graph, the numbering of the nodes and the tightness of the resource constraints (Feillet et al., 2004). For the first two factors, we can use topological sorting to speed up the algorithm (Kahn, 1962). When the nodes in a directed acyclic graph are in topological order, it means that for every arc $(i, j) \in A$, node i will always come before node j . For the last factor, we can make the resource constraints as tight as possible, since the number of pallets is always non-negative and we know the maximum number of pallets that a reach truck can carry.

5 Results

In this section, we will report the results of the methods described in the previous section. Section 5.1 will show the results of the sorting model that is created to correctly define the costs of our main optimisation problem. Next, Section 5.2 will focus on the results of the Resource Constrained Elementary Shortest Path Problem (RCESPP), first with the Mixed Integer Programming (MIP) formulation in Section 5.2.1 and then with our labelling algorithm in Section 5.2.2.

5.1 Sorting Model

The objective of the RCESPP is defined as the total costs for the driving time between pick replenishment orders and sorting time at the pick and wood locations. From our observations and the conversations with the reach truck drivers, we learned there are multiple strategies for the sorting the pallet stack at the pick location. Some of these strategies included a smarter and faster way for stacking the remaining products of the old pallet onto the new pallet, but required more moves for adding the old pallet to the stack of empty pallets. Since the total time of both actions (stacking the products and adding the pallet to the stack) was approximately equal, other methods would not lead to different results. Therefore, we decided to choose the method in Section 4.2, since this method is used significantly more than the others, is used in all distribution centres and is the only method learned by new reach truck drivers.

Table 3: Time (in seconds) per move.

DCP	LDC	DCT
11.75	8.83	12.16
12.99	13.01	13.27
13.96	10.03	15.23
11.38	14.55	14.34
11.67	14.07	
12.39	10.88	
14.69	14.36	
15.36	13.55	
13.39	7.96	
12.23	9.67	
14.05		
14.53		
14.19		

In Table 3 we see the results of the time (in seconds) required to perform one move with pallets, where each column contains observations of one distribution centre. On average, it will take 12.76 seconds to put the forks underneath a (stack of) pallet(s), move it somewhere else and pull the forks back. The results might suggest there are some difference between the distribution centres. Therefore we looked at the number of observations (#), average, variance and sum of ranks (R) and performed a Kruskal-Wallis test and ANOVA F-test. Table 4 shows these results and since we cannot reject the null hypothesis (p-values are larger than 0.05), we can conclude the differences between the distribution are not significant.

Table 4: Statistics and two tests for the different distribution centres.

DC	#	Average	Variance	R
<i>DCP</i>	13	13.28	1.69	165
<i>LDC</i>	10	11.69	6.20	171
<i>DCT</i>	4	13.75	1.77	42
Kruskal-Wallis Test			H	p-value
			2.656	0.2650
ANOVA Single Factor			F	p-value
			2.7733	0.0825

Even though the differences are not significant, there are still some differences between the distribution centres in the number of wood locations per area and the space around them. For DC Geldermalsen (LDC), there is a wood location for almost every aisle and it is very spacious around these wood location. Hence, it is less busy around a wood location and the results reflect this. In DC Pijnacker (DCP) and DC Tilburg (DCT) the wood locations are mostly located in smaller aisles where it is busy. Reach truck drivers have therefore less moving space and take more time to perform their actions.

Another important cost component of the RCESPP is sorting at the wood location. In Section 4.3, we introduced a sorting model where we determined a linear relation between the time and the number of pallet types by performing a regression analysis. We used Ordinary Least Squares (OLS) for this regression model to investigate whether the number of pallet types, the total number of pallets, the area, a factor for the distribution centre or a pallet aligner have a significant effect on the sorting time. Figure 7 shows the representation of the different areas in the observations.

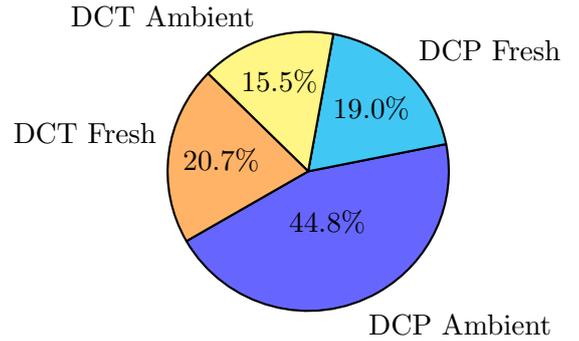


Figure 7: Representation of the areas in the regression model.

Table 5 shows the characteristics of the variables used in the OLS regression. The *time* column shows the total sorting time (in seconds). *Pallets* is the total number of pallets and *pallet types* is the number of different pallet types in the stack. The *pallet aligner* column shows the characteristics of the variable `aligner`, which was set to 1 if there was a pallet aligner present at the wood location. The last columns shows the statistics for the variable DC, which was set to 1 for all observations in DC Pijnacker (DCP) and remained 0 for DC Tilburg.

Table 5: Descriptive statistics of the observations.

	Time (s)	Pallets	Pallet Types	Pallet Aligner	DC
<i>Mean</i>	51.8793	7.7414	2.6552	0.4483	0.6379
<i>Standard Error</i>	4.4054	0.6416	0.1850	0.0659	0.0637
<i>Median</i>	45	7	2.5	0	1
<i>Mode</i>	12	10	1	0	1
<i>Standard Deviation</i>	33.5502	4.8866	1.4086	0.5017	0.4848
<i>Sample Variance</i>	1125.6168	23.8793	1.9843	0.2517	0.2350
<i>Kurtosis</i>	-0.8704	-0.0722	-0.4764	-2.0255	-1.7130
<i>Skewness</i>	0.4567	0.5593	0.5680	0.2136	-0.5893
<i>Range</i>	119	20	5	1	1
<i>Minimum</i>	7	1	1	0	0
<i>Maximum</i>	126	21	6	1	1
<i>Sum</i>	3009	449	154	26	37
<i>Count</i>	58	58	58	58	58

Figure 8 shows our observations for the sorting time and the number of pallet types. For some observations we see a higher variance than for others, this might be caused by other effects such as the busyness of (the space around) the wood location and thus the moving space of the reach truck, the experience of the reach truck driver, etc. Based on the observations alone, we could also suggest a nonlinear regression. However, the regression will be based on the number of pallet types (and probably some other factors), which is one of the decision variables in our Resource Constrained Elementary Shortest Path Problem. In order to keep the MIP formulation, as proposed by Section 4.4, we can only use a linear regression.

The equation resulting from nonlinear regression would be $-0.64x^2 + 26.85x - 13.21$, where x is the number of pallet types. This changes the MIP formulation into a Mixed Integer Quadratic Program (MIQP), which is more complex and even harder to solve (Kenney et al., 2021). Another solution is to forget about regressions and look at each pallet type myopically, by only using the averages. However, we would require more observations and a lower variance to ensure the validity of these averages. Besides, having two types of pallets, for example, requires the same procedure as for only one type, but then performed twice in a row. Hence, using a nonlinear regression or averages does not seem like a logical choice. All things considered, it would be best to use a linear regression model.

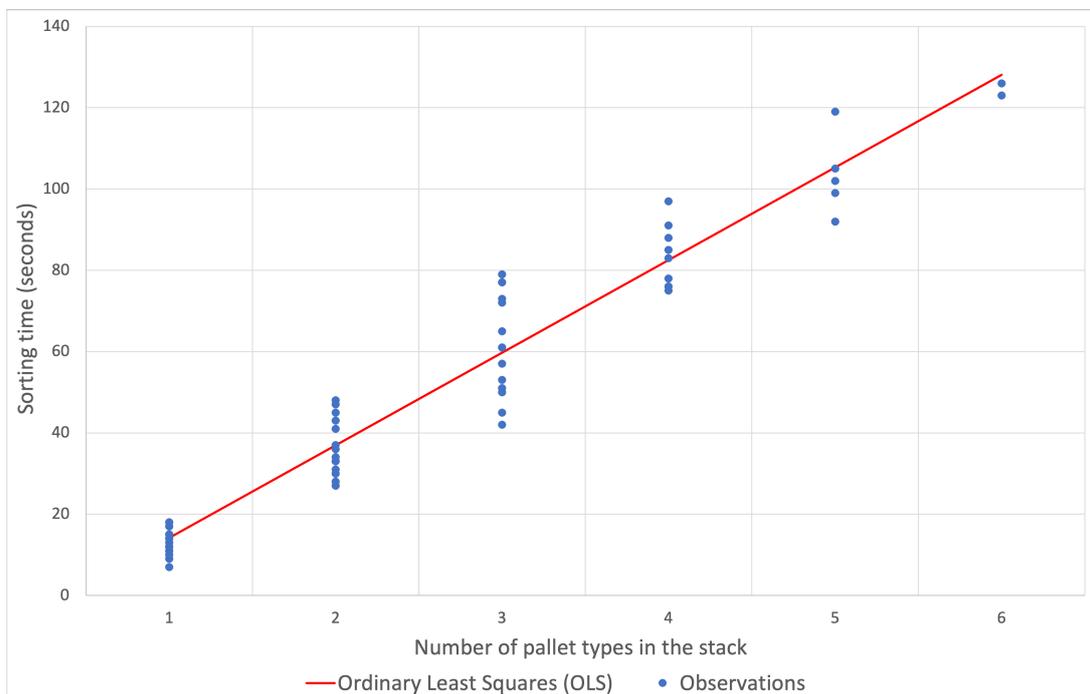


Figure 8: Sorting time at the wood location: observations in the distribution centres and an Ordinary Least Squares (OLS) regression model (OLS-type) on the number of pallet types.

In Table 6, we summarise the results of the different OLS regressions. The complete regression analysis for each OLS regression can be found in Appendix A. The significance level used in these regressions was set to 1%.

Table 6: Summary of the performed OLS regressions.

	R^2	Adjusted R^2	Standard Error	ANOVA F-stat	Significant (Total) Variables	Average P-value
<i>OLS-type</i>	0.9156	0.9140	9.836	607.17	2 (2)	0.0015
<i>OLS-type-DC</i>	0.9293	0.9267	9.084	361.26	2 (3)	0.0744
<i>OLS-type-DC*</i>	0.9788	0.9606	9.127	1294.09	2 (2)	0.0001
<i>OLS-type-aligner</i>	0.9266	0.9240	9.252	347.28	3 (3)	0.0019
<i>OLS-type-pallets</i>	0.9157	0.9126	9.918	298.66	2 (3)	0.2591
<i>OLS-type-SML</i>	0.9172	0.9126	9.918	199.40	1 (4)	0.3031
<i>OLS-type-area</i>	0.9328	0.9277	9.020	183.89	2 (5)	0.1193

In the regression **OLS-type**, the required time to sort a stack only depends on the number of pallet types. According to the results, this model fits the data quite good, all explanatory variables are significant and have a low p-value and high F-statistic, one of the best compared to the other regressions. The advantage of using this regression is the simplicity, only one variable we need to keep track of in the RCESPP and easy implementation in practice, since all distribution centres can use the same model.

Next, we have added the DC variable in **OLS-type-DC**, where DC was set to 1 for all observations in DC Pijnacker (and thus 0 for DC Tilburg). When there are differences between the two distribution centres, this model could use the extra variable DC to adjust these differences. In the results, we can see that the F-statistic and all p-values are worse than the previous regression. Besides, the intercept is not significant anymore. Hence, we decided to perform another regression on the DC variable where the intercept was removed (or set to 0). This regression, **OLS-type-DC***, has the same advantages as the **OLS-type-DC** regression and all variables are significant. Moreover, it has the best results of all regressions. However, the results are based on only two distribution centres (Tilburg and Pijnacker) and we would therefore require observations from the other locations (Geldermalsen, Zaandam and Zwolle) to do a complete regression analysis and give a correct conclusion.

During our visits to the distribution centres, we have seen that some wood locations have a pallet aligner present. Therefore, we decided it would be interesting to investigate whether this pallet aligner affects the time it takes to sort a stack of empty pallets. Regression **OLS-type-aligner** shows that a pallet aligner indeed has a significant effect on the time. Although we expected the pallet aligner to have a negative coefficient (as it is an auxiliary tool), it turns out the pallet aligner has a positive effect on the time. We also see that the average p-value is a bit higher and the F-statistic quite lower than the regression models without the aligner variable.

Until now, we only investigated the effect of the number of pallet types, since the stack of empty pallets is sorted by type. In the regression `OLS-type-pallets`, we also check whether the total number of pallets, the height of the stack, might influence the sorting time. In `OLS-type-SML`, we additionally classify the height of the stack (number of pallets in total) into 3 groups; small, medium and large. Small means less than 5 pallets in the stack, medium 5 to 9 pallets and large 10 or more pallets. Since a stack of pallets can only be in one group, we have set the medium group as default in the regression model. The differences between the two regressions is that `OLS-type-pallets` has a different value for each number of pallets, which is harder to implement in real practice and might be overfitting the data, while the `OLS-type-SML` has only three different values. For both regression models, `OLS-type-pallets` and `OLS-type-SML`, we have a low R^2 , high standard error, low F-statistic, insignificant variables and high p-values, which means these regressions fit our data very bad.

Lastly, regression `OLS-type-area` investigates whether a different area in the distribution centre might lead to different results. The advantage of this regression might be to filter out some effects caused by the differences of every wood location. Although the R^2 and standard error seems to be quite good, we also see the lowest F-stat of all regression models. Unfortunately, all variables related to the area are insignificant, which means we can conclude that the area does not affect the duration for sorting a stack of empty pallets.

Only regression models `OLS-type`, `OLS-type-DC*` and `OLS-type-aligner` perform good and have a good fit for our data. However, we see that the pallet aligner has the opposite effect of what we expected. In our observations, a pallet aligner was only present at the wood location in the ambient area of DC Pijnacker (DCP). Therefore, we think that the small moving space and the busy aisle (where the wood location is situated) causes the effect we see in the regression rather than the pallet aligner itself. We have to plan another visit to DC Zaandam (DCZ), where they also have a pallet aligner, in order to ensure this.

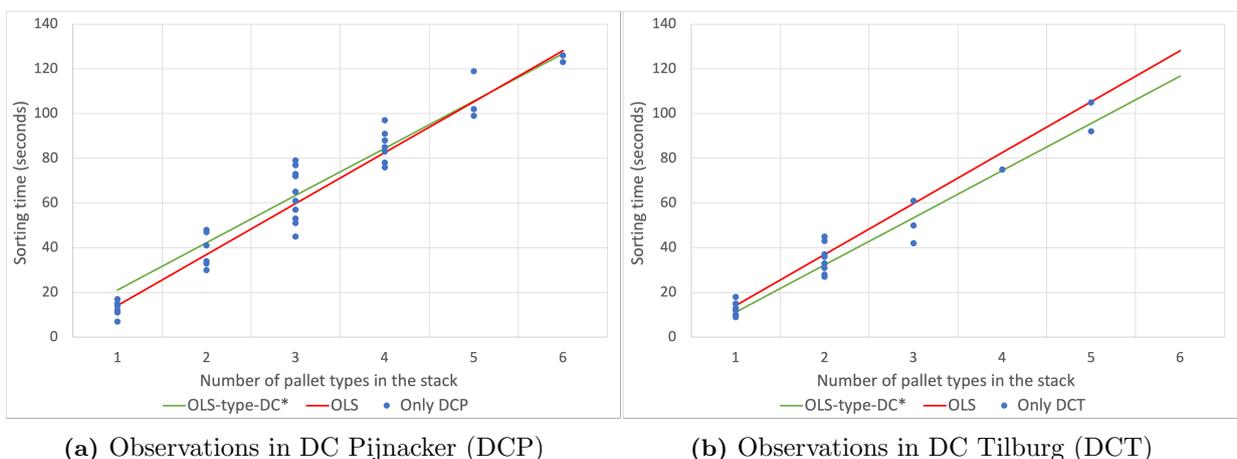


Figure 9: Results of the `OLS-type-DC*` regression.

If we look at the results in Table 6, we see that regression **OLS-type-DC*** should have the best fit. However, when plotting the data and adding the line that results from this regression in Figure 9, we see that for DC Pijnacker (DCP) the line does not fit the observations with only one pallet type (Figure 9a). For DC Tilburg (DCT) we see that the line might be a little too low for the observations with one or two pallet types (Figure 9b). Besides, we would require observations from all other distribution centres before we could give a complete regression analysis with effects for each DC. The line resulting from the regression **OLS-type**, where the time only depends on the number of pallets, might therefore be the best choice for the sorting time, which will be one of the cost components of our main optimisation problem. Figure 8 shows all observations, together with the results of the **OLS-type** regression.

5.2 Resource Constrained Elementary Shortest Path Problem (RCESPP)

The optimisation problem of finding an optimal wood strategy, i.e. determining when a reach truck driver should drop the stack of empty pallets at the wood location, can be solved using a Resource Constrained Elementary Shortest Path Problem, as described by Section 4. We have used two different approaches. First, with a Mixed Integer Programming (MIP) formulation and a commercial solver and then with a labelling algorithm. Results of this first method will be discussed in Section 5.2.1. Section 5.2.2 will focus on the results of our second approach, the labelling algorithm.

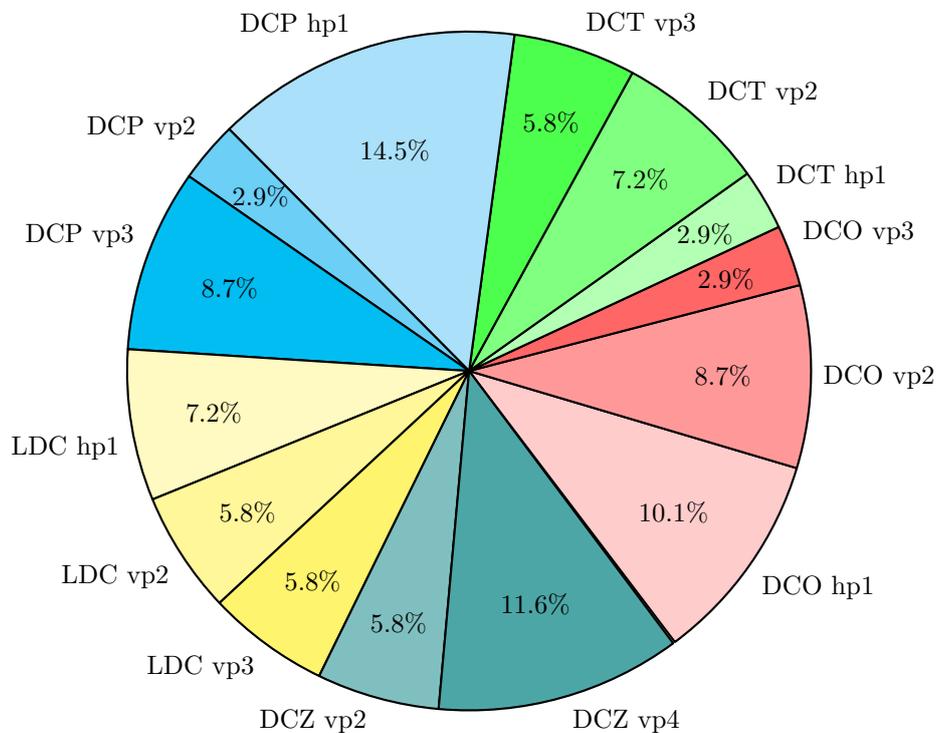


Figure 10: Pie chart illustrating the representation of the areas in the results of the Resource Constrained Elementary Shortest Path Problem (RCESPP).

Before moving on to the results of our two solving methods, we will present some information about the datasets and the analyses that will be performed on the results. As described by Section 4.1, we transformed the data into individual instances with 20 to 40 pick replenishment orders. These pick replenishment orders were all performed on 1 December 2020 across all distribution centres of Albert Heijn. Figure 10 shows the representation of each area.

Table 7 shows the characteristics for each area. Each area has its own wood location(s). The ambient and the fresh 2 °C areas are larger than the fresh 14 °C areas, so we see more choices for the wood location there. In the ambient and crate picking areas in DC Geldermalsen (LDC), there is a cross-aisle that crosses all picking aisles in the middle. At every intersection, there is a wood location in the cross-aisle. During the selection of datasets, we ignored all fresh 0 °C areas (vp1) and the medicine area (hp5), since these areas do not allow reach trucks and are therefore not interesting for our research.

Table 7: Characteristics per area.

DC/Area	Name	Observations	Average Pick Replenishments	Wood Locations in Area
<i>DC Zwolle</i>	DCO	15	30.67	7
- Ambient	hp1	7	31.57	2
- Fresh 2 °C	vp2	6	30.17	3
- Fresh 14 °C	vp3	2	29	2
<i>DC Pijnacker</i>	DCP	11	31	7
- Ambient	hp1	2	27	3
- Fresh 14 °C	vp2	5	33	1
- Fresh 2 °C	vp3	4	30.5	3
<i>DC Tilburg</i>	DCT	18	25.67	7
- Ambient	hp1	10	25	4
- Fresh 14 °C	vp2	2	31.5	1
- Fresh 2 °C	vp3	6	24.83	1
<i>DC Zaandam</i>	DCZ	12	27.83	2
- Fresh 14 °C	vp2	4	27.25	1
- Fresh 2 °C	vp4	8	28.13	1
<i>DC Geldermalsen</i>	LDC	13	39.38	53
- Ambient	hp1	4	28.75	47
- Flammable	hp2	4	38.5	1
- Crate Picking	hp3	3	32.67	5
- Ambient Merged	hp1+hp3	2	72.5	52
Grand Total		69	30.57	76

We have 69 instances in total, which are all solved as a Resource Constrained Elementary Shortest Path Problem (RCESPP) using both methods, the MIP formulation and the labelling algorithm. The solution of the RCESPP is a path from the source node to the sink node, passing through all pick replenishment nodes. After the last pick replenishment order in the dataset, a

reach truck driver is finished with his shift and always has to empty the stack of empty pallets. Therefore we have no direct arc connecting the last pick replenishment node and the source node, the path has to go via a wood location node. Since the number of pallets on a reach truck may not exceed 12 pallets and each dataset has 20 to 40 pick replenishment orders, the path has to visit some other wood location nodes as well. This path contains the optimal moments when the reach truck drivers should drop the stack of empty pallets at the wood location. Furthermore, it also shows which wood location (in case we have multiple) has the least extra distance.

Even though the solutions are based on datasets of pick replenishment orders in the past (1 December 2020), we have still got some useful information out of the results. All this information is then combined to advise on what moments (in general) might be the best to visit a wood location, so this advice/strategy can be implemented in algorithms Albert Heijn is currently developing.

For each instance, we have looked at the path of the solution. For a good advice for the wood strategy, we have analysed the following features.

- **Pick Replenishments per Wood Location:** The number of pick replenishment orders before visiting a wood location.
- **Aisle Switch:** When the aisle of the pick replenishment order before and after visiting a wood location are not the same. For example, if a reach truck driver finished a pick replenishment order in aisle 270, drops the stack of empty pallets at the wood location and then starts a pick replenishment order in aisle 272, we have an aisle switch. We can check whether we have an aisle switch by combining the optimal path from the results and the aisle information of the pick replenishment orders in the database.
- **Type Switch:** When visiting a wood location avoids adding a new pallet type to the stack of empty pallets and therefore extra sorting time at the wood location. The optimal path from the results has to be combined with the information of the pallet types corresponding to the pick replenishment orders on the nodes before and after the wood location.
- **Type3 Switch:** When visiting a wood location avoids adding a pallet that is already in the stack of empty pallets and has a high chance of not being at the top or bottom.
- **Low Ratio Switch:** Suppose Figure 11 is part of a graph and a reach truck driver is at pick replenishment node $R6$ and the optimal solution is to visit wood location $H6,2$ before continuing with pick replenishment $R7$. When the ratio $LR = (w_{R6-H6,2} + w_{H6,2-R7}) / w_{R6-R7}$ is smaller than 1.5, we have a low ratio switch.

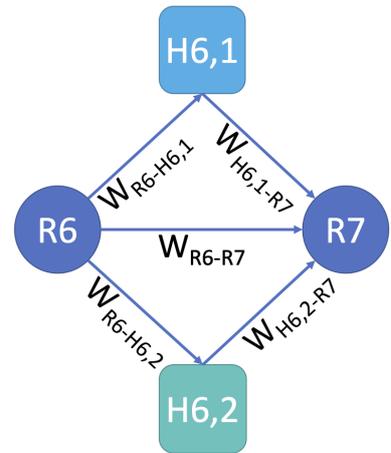


Figure 11: Part of a RCESPP graph between the 6th and the 7th pick replenishment order.

Additionally, if an extraordinary choice was made for a wood location, we did some extra analysis on why other possible choices (which seemed better or more logical at first sight) perform worse, to gain a better understanding of which factors are most important for a wood strategy. Since the analyses of the switches requires combining multiple data sources, it is very time consuming. Therefore, some instances do not include this detailed analysis, but only show the main results of the RECESPP. In order to maintain a good representation of all areas, we required that for each area, at least 50 % of the instances includes an analysis for the switches. Overall, we performed this switch analysis on 43 instances, which is 62.3 % of all instances.

5.2.1 Mixed Integer Programming (MIP) formulation

In this section, we discuss the results of the Resource Constrained Elementary Shortest Path Problem (RCESPP) solved using the MIP formulation from Section 4.4 and a commercial solver. The solver we have used is CPLEX 12.9 in combination with Java. All instances were run on a MacBook Pro with an 8-core Apple M1 CPU and 8GB of RAM. We set a time limit of 3600 seconds (1 hour) per instance.

Table 8 describes the results of the RCESPP, summarised by area. In the first column we see the number of instances that were run for each area, then the running time of the two solution approaches. Next, we have the number of pick replenishment order between two visits at the wood location (*PR per WL*) and the averages of the switches. All detailed results (per instance instead of summarised by area) are presented by Table 19 in Appendix B. Even though some areas have the same name in other distribution centres, it is important to know we cannot compare these areas, as they may have a different number of wood locations and pick replenishment orders per instance.

For the average running time for the MIP formulation, we see that those for DCP vp2, DCT vp2/vp3 and DCZ are quite low compared to the running times for other areas. This is caused by the number of wood locations in those areas (shown in Table 7). All aforementioned areas have only 1 wood location. When we have more wood locations, the graph becomes larger and therefore more complex and this causes an increase in the running time. This is definitely reflected by DC Geldermalsen (LDC), where there are a lot of wood locations (in hp1 and hp3 in every aisle), as we see that the running is extremely high for the MIP formulation and sometimes even reached the time limit of 1 hour. It is very clear that a large number of wood locations is the main reason for the high running time. Nevertheless, we see that increasing the number of pick replenishment orders increases the running time as well.

Table 8: Most important results of the Resource Constrained Shortest Path Problem (RCESPP). Summarised per area.

DC/Area	#	Time MIP (s)	Time LA (s)	PR per WL	Aisle (%)	Type (%)	Type3 (%)	Low Ratio (%)
DCO	15	343.90	0.0044	8.72	75.9	41.4	17.2	37.9
- <i>hp1</i>	7	615.13	0.0057	9.30	81.8	27.3	27.3	0.0
- <i>vp2</i>	6	63.63	0.0037	7.09	73.3	60.0	13.3	73.3
- <i>vp3</i>	2	235.40	0.0018	11.58	66.7	0.0	0.0	0.0
DCP	11	54.40	0.0074	9.06	88.9	55.6	5.6	44.4
- <i>hp1</i>	2	177.93	0.0114	7.83	80.0	60.0	0.0	60.0
- <i>vp2</i>	5	3.63	0.0076	10.04	100.0	42.9	0.0	14.3
- <i>vp3</i>	4	56.12	0.0051	8.45	83.3	66.7	16.7	66.7
DCT	18	346.76	0.0065	7.97	67.7	54.8	19.4	48.4
- <i>hp1</i>	10	617.97	0.0091	6.86	63.2	63.2	31.6	63.2
- <i>vp2</i>	2	26.93	0.0048	8.96	60.0	60.0	0.0	20.0
- <i>vp3</i>	6	1.35	0.0022	9.49	85.7	28.6	0.0	28.6
DCZ	12	9.57	0.0044	9.73	71.4	28.6	21.4	14.3
- <i>vp2</i>	4	1.33	0.0047	8.54	75.0	50.0	0.0	25.0
- <i>vp4</i>	8	13.70	0.0043	10.32	70.0	20.0	30.0	10.0
LDC	13	1610.69	0.1886	8.54	92.1	47.4	39.5	84.2
- <i>hp1</i>	4	1823.92	0.1077	8.48	100.0	0.0	100.0	100.0
- <i>hp2</i>	4	256.13	0.0758	9.20	85.7	28.6	14.3	57.1
- <i>hp3</i>	3	1806.42	0.1172	8.83	88.9	44.4	44.4	88.9
- <i>hp1+hp3</i>	2	3600.00	0.6628	6.94	94.7	63.2	36.8	89.5
Average	69	479.02	0.0397	8.72	80.0	46.9	23.1	52.3

5.2.2 Labelling Algorithm

Besides the MIP formulation, we have used another method to solve the Resource Constrained Elementary Shortest Path Problem (RCESPP). As described by Section 4.5, this second method is a labelling algorithm. In this section, we will focus on the results of this labelling algorithm and compare these results with the MIP formulation.

Since our labelling algorithm is also an exact solving method for the RCESPP, and thus guarantees optimality for its solutions, the paths resulting from this algorithm equals those presented in Section 5.2.1. Therefore, we have no gap between the objective values. However, the method itself is completely different. The MIP formulation in CPLEX uses a branch and cut algorithm, and has to make lots of decisions when exploring the solution space. The labelling algorithm simply ‘walks’ through the entire graph stores a label at each node when the extension to this node is feasible. While walking through the graph, the labelling algorithm only looks at non-dominated labels to reduce the total number of labels at each node. At the end of the labelling algorithm, we only have to choose the label with the lowest cost, as this label contains the optimal path.

The labelling algorithm is therefore much faster than solving the MIP formulation with a commercial solver like CPLEX. This is also reflected by the running time in Table 8, which is less than a second for all instances. Again, we see that the running time is strongly related to the number of wood locations in the area. Due to the completely different solving method, we see that the running time of the labelling algorithm is now related to the number of pick replenishment orders as well. Obviously, if we have more pick replenishment orders (or nodes), the labelling algorithm has to ‘walk’ further in a graph, creating and storing more labels and we have therefore a higher running time.

Even though a reach truck can carry up to 12 pallets, we see that the average number of pallets in the stack (equal to the number of pick replenishment orders until the wood location is visited, i.e. PR per WL) is significantly lower. Obviously, carrying more pallet can result in fewer visits to a wood location. However, more pallets in the stack also means a higher chance of more pallet types and therefore a higher sorting time at both the pick and wood location. In Table 19 in Appendix B, the total costs (objective value) of an instance is divided into three groups: distance costs, node costs and wood costs. On average, the distance costs are 44.0% of the total costs, which include the travel time between pick replenishment locations and the travel time from and to the wood locations. Around 41.6% of the total costs are related to the node costs, i.e. the sorting costs at the pick location when a reach truck driver has to put the empty pallet in the right position in the stack. The costs related to dropping and sorting the stack of empty pallets at the wood location (wood costs) are around 14.4% on average.

Finally, Table 8 shows that an aisle switch is the most important factor for deciding when a reach trucker driver should drop the stack of empty pallets at the wood location. In 52.3% of the cases, we see that a low ratio switch is also an important factor. For a lot of instances, we had a low ratio switch and an aisle switch simultaneously. Switching aisles automatically means a larger distance between two pick replenishment locations and visiting a wood location is therefore less harmful than the case where two pick replenishment locations close to each other. This is also part of the advice that will be discussed later on in Section 7.

6 Sensitivity Analysis

In this section, we will discuss some further analysis for the wood strategy. After gathering the results in Section 5, where we found that our labelling algorithm runs very fast, we decided to do some sensitivity analysis on the wood locations and areas. In this analysis, we will test some extra scenarios, which will be insightful for a business case of Albert Heijn and might reduce the costs as well. Section 6.1 discusses the merging of two areas. In Section 6.2 we will test a scenario where we add a wood location at the end of every aisle. Section 6.3 presents the scenario where we test for (temporary) wood locations in the middle of every aisle. Finally, Section 6.4 focuses on the addition of extra wood locations at strategic locations.

6.1 Merging two areas

Using our methods described in Section 4, we gathered results for lots of instances for each area of the distribution centre. However, after analysing the results we learned that some areas share the same reach trucks and it is therefore possible to perform pick replenishment orders in multiple areas. At the moment, there is only one combination of areas where it is possible to switch areas and continue with the pick replenishment orders of the other area. The process to switch between the ambient and crate picking area in DC Geldermalsen (LDC) is as follows. There are two stickers on the reach truck, one for the ambient area (hp1) and one for the crate picking area (hp3). When we want to switch areas, we have to log out of the terminal first, then log back in and scan the sticker of the other area. Then, we can continue with our pick replenishment orders, but now for the other area.

Even though the terminal and the reach truck remains the same, scanning the other sticker changes the reach truck id in the Warehouse Management System (WMS). Hence, we had to make some queries and combine multiple databases to create datasets with reach trucks performing pick replenishment orders in both hp1 and hp3. We have created two large instances and solved these instances with both the MIP formulation and the labelling algorithm. Unfortunately, the MIP formulation could not solve these instances to an optimum within the time limit of 1 hour (3600 seconds), so we have only used the results of the labelling algorithm. These results are also included in Table 8 in the previous section, or Table 19 in Appendix B for the full results. From the results, we can conclude that merging two areas leads to the same advice for the wood strategy. Even though the instances consist of around 70 pick replenishment orders and 52 possible wood locations, the running time of the labelling algorithm is still less than a second.

6.2 Wood locations at the end of every aisle

In the results in Table 8, we see that the percentage of low ratio switches is significantly higher in the ambient and crate picking areas in DC Geldermalsen (hp1 and hp3 in LDC, respectively).

In these areas, we have a wood location in every aisle. In the middle of every aisle, there is an intersection with the cross-aisle, and the wood locations are situated at these intersections. Since every aisle has a wood location, we have a shorter extra distance when dropping the empty pallets at the wood location and therefore more low ratio switches. Hence, we did some further analysis on whether adding similar wood locations could reduce the total costs. Unfortunately, the other distribution centres do not have a cross-aisle with intersections at the middle of every aisle, so we decided to create the new wood locations at the end of every aisle.

Table 9: Results before and after creating the wood locations at the end of every aisle.

RT	DC	Area		Wood Locations					Obj	Dist	Node	Wood
vr21	dcp	vp3	Before	12,1	23,1	28,2			698.14	291.66	364.00	42.48
			After	12,3	23,5	28,5			691.86	285.38	364.00	42.48
vr05	dcp	vp3	Before	1,2	13,3	25,3	28,3	36,3	1185.85	533.10	468.00	184.75
			After	1,16	13,16	25,16	28,16	36,15	1183.31	530.56	468.00	184.75
vr17	dcp	vp3	Before	11,3	23,3	33,3			862.09	206.29	476.58	179.22
			After	11,17	12,16	24,15	33,14		855.99	228.45	456.95	170.59
vr20	dcp	vp3	Before	9,3	15,3	16,2	25,3		854.22	427.00	325.00	102.22
			After	11,17	15,9	16,7	25,16		855.07	427.85	325.00	102.22
vr51	dcp	vp2	Before	3,1	14,1	22,1	30,1	39,1	1053.75	407.58	507.00	139.17
			After	3,1	14,1	22,1	30,1	39,4	990.36	344.19	507.00	139.17
vr52	dcp	vp2	Before	7,1	19,1	28,1	39,1		998.75	389.53	507.00	102.22
			After	7,1	19,1	28,1	39,1		943.72	507.00	507.00	102.22
vr55	dcp	vp2	Before	11,1	22,1	32,1			698.87	240.39	416.00	42.48
			After	11,6	22,5	32,6			702.18	243.70	416.00	42.48
vr57	dcp	vp2	Before	10,1	22,1			464.13	149.81	286.00	28.32	
			After	10,4	22,6			456.66	142.34	286.00	28.32	
vr60	dcp	vp2	Before	12,1	22,1	33,1			805.94	288.88	429.00	88.06
			After	12,2	22,1	33,1			774.66	257.60	429.00	88.06

This scenario is tested on all instances of the fresh 14 °C (vp2) and fresh 2 °C (vp3) areas in DC Pijnacker (DCP). Table 9 shows the reach truck id, DC and area of these instances. For each instance, we see the visited wood locations, where a, b indicates dropping wood after pick replenishment order a at wood location b . In the right columns we see the objective function (total costs) and the costs related to the driving time and sorting time at the pick and wood locations. The results show that having a wood location at the end of every aisle (instead of the current wood locations) leads to a small decrease in the total costs, 2.03 % on average. There is even one case where the node and wood costs changed as well, even though the wood location is visited once more. However, there are also some cases where the costs increase, probably because the current wood location was already nearby and reach truck driver might have to go to the other side of the aisle for the new wood location. The timing for dropping the pallets (e.g. after the 11th pick replenishment order) did remarkably not change for most instances.

6.3 (Temporary) wood locations in the middle of every aisle

The next scenario we have tested is adding (temporary) wood locations in the middle of every aisle. During our visits to all distribution centres, we have seen that some reach truck drivers use empty pick locations, where no products are slotted for that day, as a temporary wood location. The reach truck drivers use these empty pick locations when the cross-aisles or wood locations are too busy. When it is less busy, or at the end of the work shift, the stacks of empty pallets will be moved to the correct wood location. Once we implement the wood strategy, which tells us exactly when a reach truck driver should visit a wood location, questions may be asked why they cannot use their own strategy anymore. Hence, we decided to explore this scenario of temporary wood locations in the middle of an aisle, in order to justify the business case for the wood strategy. It might also help to explain to the reach truck drivers why following the wood strategy is so important and why their own strategy cannot be used anymore.

Dropping the stack of empty pallets when it is less busy requires some changes in the process for the wood strategy. For every aisle, we have assigned a wood location to the pick location closest to the middle of the aisle. Thereafter, we have solved the RCESPP with our labelling algorithm and the coordinates of the new wood locations. However, the stack of empty pallets is only stored at the temporary wood locations and not sorted. Hence, we have set the wood cost to zero for now. Then, we had to calculate the costs for the distances between the new (temporary) and the correct wood locations, which will be referred to as $dist^*$ in the results. Lastly, we had to calculate the costs for moving the groups of pallets in the (mixed) stack of empty pallets to the right stack at the correct wood location.

Table 10 shows the results with the distance, node, $dist^*$ and wood costs, the total costs and absolute en relative difference with the original results from Section 5. We see that the difference in costs is very high, both absolute and relative. Besides, the number of empty pick locations in an aisle is very low. In some distribution centres, there are no empty pick locations for a great majority of the aisles. Even if a reach truck driver thinks the pick location is empty, it happens sometimes that throughout the day a new product will be slotted at that specific location, and the stack of empty pallets thus needs to be moved somewhere else. This takes even more time and causes delays in both pick replenishment and other processes. Even though the reach truck drivers avoid the busyness around the wood location when using this strategy, we strongly advise against this strategy as it has more costs than benefits.

Table 10: Results for mid-aisle wood locations and the differences in costs.

RT	DC	Area	Wood Locations	Dist	Node	Dist*	Wood	Total	Difference	%
vr32	dct	vp2	7,5 17,6 29,5 37,6	298.46	485.29	88.82	444.07	1316.64	326.88	33.0
vr42	dct	vp2	4,4 14,3 26,2	197.16	338.00	65.81	65.27	666.24	83.58	14.3
hr07	dcp	hp1	12,14 21,14 28,11	343.74	381.16	85.36	361.54	1171.80	316.71	37.0
hr25	dcp	hp1	11,7 20,6 24,6 35,6 40,13	629.45	520.00	131.47	344.28	1625.20	263.76	19.4
vr66	dco	vp2	4,9 16,9 28,9 37,10	578.15	481.00	119.00	330.12	1508.27	234.50	18.4
vr89	dco	vp2	5,23 16,23 26,21	158.06	338.00	195.89	88.06	780.01	133.66	20.7

6.4 Strategically adding extra wood locations

According to the results and analyses of Section 5, adding wood locations might reduce the total costs in our wood strategy problem. Important here is that these new wood locations cover the parts of the area where we have a longer distance to the current wood locations. Additionally, these new wood locations should not decrease the safety in general or the capacity of the area. Hence, we decided to add wood locations that can be easily implemented in reality and have a good chance of decreasing the total costs. Strategic locations were selected using the maps, layouts and our observations of the distribution centres. Simultaneously, we discussed these strategic locations with the team leaders in each distribution centre to decide whether adding a certain wood location does not affect the safety and can be implemented quite easily.

In DC Geldermalsen (LDC), we already have a wood location at every intersection with the cross-aisle, which is in the middle of every aisle. Therefore, we do not need to add more wood locations in this distribution centre. In DC Zwolle (DCO), the results showed that in the fresh 14 °C area (vp3), we already have two wood locations, while the same area in other distribution centres only have one. For the fresh 2 °C area in DC Zwolle and DC Pijnacker (DCO vp2 and DCP vp3, respectively), we already a high percentage of low ratio switches, which means that an additional wood location here would probably not lead to a decrease in the total costs. For all other areas, we have added some additional wood locations.

In Table 11, we present a summary of the results of adding the additional wood locations. Detailed results for each instance are presented by Table 20 in Appendix C. WL stands for the average number of visited wood locations and Obj for the objective value (total costs), where * stands for the results before adding the extra wood locations and \pm and % stand for respectively the absolute and relative differences. The addition of wood locations in DCO hp1, DCP hp1 and vp2 and DCZ vp4 is clearly resulting in a cost reduction. This is remarkable since especially these areas have an increase in the number of visited wood locations, which also shows the high potential of these new wood location. In DCT vp3, all instances consisted of pick replenishment orders that were not close to the new pick location. Even though it is very likely that this new wood location should be added, more research is required to ensure this.

Table 11: Results of the RCESPP when adding additional wood locations, summarised by area.

DC/Area	Time (s)	WL	WL*	WL \pm	Obj	Obj*	Obj \pm	Obj%
<i>DCO</i> <i>hp1</i>	0.0153	3.86	3.43	0.43	1126.69	1193.76	-67.07	-5.2
<i>DCP</i> <i>hp1</i>	0.0028	3.50	3.50	0	704.70	750.72	-46.02	-7.1
<i>DCP</i> <i>vp2</i>	0.0085	3.60	3.40	0.20	745.09	804.29	-59.20	-6.4
<i>DCT</i> <i>hp1</i>	0.0104	4.18	3.82	0.36	932.57	945.90	-13.33	-1.5
<i>DCT</i> <i>vp2</i>	0.0053	3.50	3.50	0	767.11	786.21	-19.10	-2.4
<i>DCT</i> <i>vp3</i>	0.0025	2.67	2.67	0	631.01	649.55	-18.54	-2.0
<i>DCZ</i> <i>vp2</i>	0.0026	3.25	3.25	0	638.06	655.93	-17.87	-2.5
<i>DCZ</i> <i>vp4</i>	0.0061	2.88	2.75	0.13	805.23	847.20	-41.97	-5.7
Average	0.0079	3.49	3.29	0.20	835.43	870.11	-34.68	-3.8

7 Conclusion

The pick replenishment process is one of the many processes in the distribution centres of Albert Heijn, the largest supermarket chain in the Netherlands, with over a thousand stores across the Netherlands and Belgium. Optimisation of this process can be achieved by increasing the efficiency of reach truck drivers. A new algorithm will be developed that gives advice on the next pick location that needs to be replenished. As mentioned by [Gagliardi et al. \(2008\)](#), this kind of replenishment problems has never been addressed in literature before. In order to give the best advice, the algorithm must take into account the disposal of empty pallets.

The main scope of this thesis is to find the optimal wood strategy, which defines when reach truck drivers should drop the stack of empty pallets at the wood location. This problem can be solved using a Resource Constrained Elementary Shortest Path Problem (RCESPP), where the nodes represent the pick replenishment orders and the wood locations and the arcs represent the costs and information flow between the nodes. Some of these costs are based on the sorting time of a stack of empty pallets. The number of moves requires to sort the stack at the pick location is based on a chance model, where each move takes approximately 12.76 seconds. By performing an Ordinary Least Squares (OLS) regression, we found that the sorting time at the wood location only depends on the number of pallet types in the stack.

In this paper, we used two different approaches to solve the RCESPP. The first approach is based on a Mixed Integer Programming (MIP) formulation of the RCESPP, which is then solved using a branch and cut algorithm. For instances up to 40 pick replenishment orders and 10 wood locations, the RCESPP can be solved within the time limit of 1 hour. Unfortunately, larger instances or more wood locations will not guarantee the optimal solution within a reasonable time.

That is where the second solving method comes in, a label correcting algorithm based on [Feillet et al. \(2004\)](#), adapted for finding the optimal wood strategy. This labelling algorithm ‘walks’ through the entire graph and stores labels at each node. These labels are only extended to other nodes when the resource consumption constraints are satisfied. While walking through the graph, the algorithm only considers non-dominated labels to reduce the total number of labels at each node. When the labelling algorithm arrives at the sink node, the label with the lowest costs is selected, as this label contains the optimal path. The running time of the labelling algorithm is extremely fast, all instances, even those with 76 pick replenishment orders and 52 wood locations, are solved in less than a second, with an average of 0.0397 seconds per instance.

Using the results and analyses, we give advice for the wood strategy. First of all, we have concluded that an aisle switch is the most important factor for dropping wood. The low ratio switch is also quite important for keeping the costs low. The type switch learned that when the next pick replenishment order comes with a new pallet type and the distance to the wood location is less than 53 meters, empty pallets should always be dropped at the wood location before continuing with the next pick replenishment order. When there is a type switch and aisle switch simultaneously, the stack of empty pallets should be dropped at the wood location. If the stack contains lots of pallet types, the type switch becomes more important than the aisle switch. For stacks with at most two pallet types, a higher stack is preferred and the stack should only be dropped at the wood location in case of an aisle switch.

Besides the switches, we see that the number of pallet types in the stack is also important. In the fresh 14 °C areas, we find the least number of pallet types. When there are two or more reach truck drivers available, we strongly advise assigning the pick replenishment orders in such a way that each reach truck driver has at most two pallet types in the stack. The more pallet types we have in the stack, the higher the chance of a pallet not being at the top or bottom of the stack. For more than three pallet types, the costs will increase faster due to the extra moves required for correctly sorting the stack. When a new pallet type is added to the stack, we would suggest to first assign all available pick replenishment orders of that same pallet type. If there are no other pick replenishment orders of the same pallet type left, it would be best to look for a new pallet type rather than a type that is already in the stack to minimise the sorting costs at the pick location.

Finally, we have some wood strategy advice left that is related to the layout of the distribution centre. If the distribution has a cross-aisle that crosses all aisles in the middle (instead of the end or beginning), the intersections of these aisles with the cross-aisle should be used as a wood location. Even though more wood locations means less distance-related costs, creating wood locations in every aisle is not advised, as the benefits would not weigh out the costs. More importantly, the number of wood locations should be proportional to the size of the area. Take the fresh 2 °C area for example, 3 wood locations in DC Zwolle and DC Pijnacker seems a good choice, while only 1 wood location in DC Tilburg and DC Zaandam seems too little. Lastly, wood locations should not be located around busy spots, as this leads to delays in the processes.

Overall, we can conclude that the proposed methods in this thesis provide decent results, especially for the labelling algorithm, where the running time remains low. Optimal solutions for the RCESPP, which is a strongly NP-hard problem, are found within less than a second, even for larger instances. The sensitivity analysis revealed that merging areas and changing wood locations does not lead to a decrease in the quality of the proposed methods. The combination of the results, analyses and advice for a wood strategy showed some interesting insights. Hence, this paper might have a valuable contribution to the existing literature.

8 Discussion

Even though the proposed solution methods produce optimal solutions of decent quality, there is always room for improvement. The sorting time at the pick and wood locations could be improved since the OLS regression in Section 5.1 is based on a small set of observations. However, once the new algorithm (that advises on the next pick replenishment location) is in use at Albert Heijn, the time required for sorting at the wood location can be calculated and we have more observations that could easily be re-calibrated.

Furthermore, we made some assumptions needed for our solving methods. Even though these assumptions are realistic, there may be some situation where an assumption may not hold. For example, the assumption where the stack of empty pallets is always sorted after replenishing the pick location. Unfortunately, during our visits to the distribution centres, we have seen some situations where the stack was not sorted. The foremost reason was busyness in the aisles which leads to less moving space for a reach truck. Even though most stacks are correctly sorted, we might investigate what effect an unsorted stack may have on the results. Another assumption is the pallet type of each product. On very rare occasions, a product may arrive on another pallet type than is defined in the database. This can result in wrong advice for the next pick replenishment location.

Finally, we discuss some ideas for future research. After implementing the wood strategy, Albert Heijn will optimise the put-away process (which is shortly described in Section 1). It might be interesting to investigate the possibility to combine the put-away and pick replenishment process in order to increase the efficiency of reach truck drivers even further. For example, when a product is almost out of stock, arriving pallets could be moved to the pick location directly, skipping the buffer location. One of the problems in the distribution centres of Albert Heijn is the delay caused by the busyness in small aisles or for other reasons. Therefore, robustness could be addressed for the Resource Constrained Elementary Shortest Path (RCESPP). Lastly, future research could be done on other solving methods for the RCESPP, such as column generation, benders decomposition or Lagrangian relaxation.

References

- Ahuja, R. K., Orlin, J. B., and Magnanti, T. L. (1993). *Network flows: theory, algorithms, and applications*. Prentice-Hall.
- Albert Heijn (2021). Albert Heijn stores in the Netherlands and Belgium.
<http://www.ah.nl/winkels> and <http://www.ah.be/winkels>.
- Beasley, J. E. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394.
- Bellman, R. (1957). Dynamic programming. *Princeton University Press*, 1(2):3.
- Coelho, L. C., Renaud, J., and Laporte, G. (2016). Road-based goods transportation: a survey of real-world logistics applications from 2000 to 2015. *INFOR: Information Systems and Operational Research*, 54(2):79–96.
- De Koster, R. and Van Der Poort, E. (1998). Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE transactions*, 30(5):469–480.
- Desrochers, M. (1987). *La fabrication d’horaires de travail pour les conducteurs d’autobus par une méthode de génération de colonnes*. Université de Montréal, Centre de recherche sur les transports.
- Desrochers, M. and Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR: Information Systems and Operational Research*, 26(3):191–212.
- Desrosiers, J., Soumis, F., and Desrochers, M. (1984). Routing with time windows by column generation. *Networks*, 14(4):545–565.
- Dijkstra, E. W. et al. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978.
- Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*, 44(3):216–229.
- Gagliardi, J.-P., Ruiz, A., and Renaud, J. (2008). Space allocation and stock replenishment synchronization in a distribution center. *International Journal of Production Economics*, 115(1):19–27.
- Gomez, M. I., McLaughlin, E. W., and Wittink, D. R. (2004). Customer satisfaction and retail sales performance: an empirical investigation. *Journal of retailing*, 80(4):265–278.

- Hall, R. W. (1993). Distance approximations for routing manual pickers in a warehouse. *IIE transactions*, 25(4):76–87.
- Hutcheson, G. D. and Sofroniou, N. (1999). *The multivariate social scientist: Introductory statistics using generalized linear models*. Sage.
- Ioachim, I., Gelinias, S., Soumis, F., and Desrosiers, J. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks: An International Journal*, 31(3):193–204.
- Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In *Column generation*, pages 33–65. Springer.
- Kahn, A. B. (1962). Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562.
- Kantar Worldpanel (2015). Thoughts On...: the British grocery market. https://www.kantarworldpanel.com/dwl.php?sn=news_downloads&id=73.
- Kenney, A., Chiaromonte, F., and Felici, G. (2021). Mip-boost: Efficient and effective l0 feature selection for linear regression. *Journal of Computational and Graphical Statistics*, pages 1–12.
- Kusrini, E., Novendri, F., and Helia, V. N. (2018). Determining key performance indicators for warehouse performance measurement—a case study in construction materials warehouse. In *MATEC Web of Conferences*, volume 154, page 01058. EDP Sciences.
- Lambert, D., Stock, J. R., Ellram, L. M., and Grant, D. (2006). *Fundamentals of Logistics Management: First European Edition*. McGraw-Hill.
- Lozano, L., Duque, D., and Medaglia, A. (2015). An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, In Press.
- Neeraja, B., Mehta, M., and Chandani, A. (2014). Supply chain and logistics for the present day business. *Procedia economics and finance*, 11:665–675.
- Noyan, F. and Simsek, G. G. (2011). Structural determinants of customer satisfaction in loyalty models: Turkish retail supermarkets. *Procedia-Social and Behavioral Sciences*.
- Pugliese, L. D. P. and Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200.
- Savage, M. and Wall, T. (2020). UK supermarkets appeal for calm as shelves empty. *The Guardian*.
- Tompkins, J. A., White, J. A., Bozer, Y. A., and Tanchoco, J. M. A. (2010). *Facilities planning*. John Wiley & Sons.
- Ziegelmann, M. (2001). *Constrained shortest paths and related problems*. PhD thesis, Universität des Saarlandes.

A Ordinary Least Squares (OLS)

OLS-type

Table 12: Regression Analysis OLS-type.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	1	56	57
R Square	0.9156	SS	58742.25	5417.91	64160.16
Adjusted R Square	0.9140	MS	58742.25	96.75	
Standard Error	9.8361	F	607.17		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-8.6311	2.7746	-3.1107	0.0029	-14.1894	-3.0728
Type	22.7896	0.9249	24.6407	0.0000	20.9369	24.6424

OLS-type-DC

Table 13: Regression Analysis OLS-type-DC.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	2	56	57
R Square	0.9293	SS	59621.59	4538.57	64160.16
Adjusted R Square	0.9267	MS	29810.79	92.52	
Standard Error	9.0840	F	361.26		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-3.6836	2.9771	-1.2373	0.2212	-9.6500	-2.2827
Type	22.0675	0.8823	25.0103	0.0000	20.2993	23.8358
DC	-8.3691	2.5638	-3.2644	0.0019	-13.5070	-3.2312

OLS-type-DC*

Table 14: Regression Analysis OLS-type-DC*.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	2	56	58
R Square	0.9788	SS	215600.10	4664.90	220265.00
Adjusted R Square	0.9606	MS	107800.05	83.30	
Standard Error	9.1270	F	1294.09		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Type	21.1231	0.4447	47.5001	0.0000	20.2323	22.0139
DC	-9.9839	2.2171	-4.5031	0.0000	-14.4254	-5.5425

OLS-type-aligner

Table 15: Regression Analysis OLS-type-aligner.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	2	55	57
R Square	0.9266	SS	59452.30	4707.85	64160.16
Adjusted R Square	0.9240	MS	29726.15	85.60	
Standard Error	9.2519	F	347.28		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-11.7907	2.8310	-4.1648	0.0001	-17.4642	-6.1172
Type	22.7918	0.8699	26.1991	0.0000	21.0484	24.5352
Aligner	7.0356	2.4428	2.8802	0.0057	2.1401	11.9310

OLS-type-pallets

Table 16: Regression Analysis OLS-type-pallets.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	5	55	57
R Square	0.9157	SS	58750.46	5409.69	64160.16
Adjusted R Square	0.9126	MS	29375.23	98.36	
Standard Error	9.9176	F	298.66		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-8.8680	2.9153	-3.0419	0.0036	-14.7104	-3.0257
Type	22.6037	1.1330	19.9507	0.0000	20.3332	24.8742
Pallets	0.0944	0.3266	0.2890	0.7737	-0.5601	0.7489

OLS-type-SML

Table 17: Regression Analysis OLS-type-SML.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	3	54	57
R Square	0.9172	SS	58847.84	5312.31	64160.16
Adjusted R Square	0.9126	MS	19615.95	98.38	
Standard Error	9.9185	F	199.40		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-9.0424	3.6538	-2.4748	0.0165	-16.379	-1.7169
Type	22.4276	1.0651	21.0577	0.0000	20.2923	24.5629
Small	0.5483	3.4713	0.1580	0.8751	-6.4113	7.5080
Large	3.1951	3.1895	1.0018	0.3209	-3.1994	9.5897

OLS-type-area

Table 18: Regression Analysis OLS-type-area.

Regression Statistics		ANOVA	Regression	Residual	Total
Observations	58	df	4	53	57
R Square	0.9328	SS	59847.89	4312.26	61460.16
Adjusted R Square	0.9277	MS	14961.97	81.36	
Standard Error	9.0202	F	183.89		

	Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	-10.3694	3.1738	-3.2671	0.0019	-16.7353	-4.0034
Type	22.3930	0.9073	24.6803	0.0000	20.5732	24.2129
DCP Ambient	6.6725	3.2034	2.0830	0.0421	0.2474	130977
DCP Fresh	3.5214	4.0179	0.8764	0.3847	-4.5374	11.5802
DCT Ambient	-5.5913	3.9979	-1.3986	0.1678	-13.6101	2.4275

B Results RCESPP

Table 19 presents the results of the Resource Constrained Elementary Shortest Path Problem (RCESPP). Each line represents an individual dataset. The columns contain the following information.

RT	Reach truck ID.
DC	Distribution centre where this set of pick replenishment orders took place.
Area	Area within the distribution centre where this set of pick replenishment orders took place.
Wood Locations	Optimal wood locations that should have been visited in order to minimise the total costs.
Obj	Objective value (total cost) of the RCESPP, sum of the distance, node and wood costs.
Dist	Costs related to the distances between the pick replenishment orders and the wood locations.
Node	Costs related to sorting the stack of empty pallets at the pick location.
Wood	Costs related to sorting the stack of empty pallets at the wood location.
Running Time	Running time of the solver when using the MIP formulation.
Running Time LA	Running time of the labelling algorithm.
PR/WL	Average number of pick replenishment orders between the visits at the wood location.
Aisle Switch	Number of times we visited the wood locations and switched aisles simultaneously.
Type Switch	Number of times we visited the wood locations in order to avoid an extra pallet type in the stack.
Type3 Switch	Number of times we visited the wood locations in order to avoid adding a pallet type that is in the middle of the stack (and therefore requires more moves and thus higher costs).
Low Ratio Switch	Number of times we visited the wood locations and this extra distance was less than 50% of the direct distance between the pick replenishment orders.

Table 19: Results of the Resource Constrained Elementary Shortest Path Problem (RCESPP).

RT	DC	Area	Wood	Locations	Obj	Dist	Node	Wood	Running Time	Running Time LA	PR/WL	Aisle Switch	Type Switch	Type3 Switch	LowRatio Switch
h165	ldc	hp1	8,16	20,31	765.80	333.37	290.16	142.27	59.36	0.0390	10	1	0	1	1
h181	ldc	hp1	10,39	22,6	1666.16	898.88	528.32	238.96	3600.00	0.1834	8.5				
h184	ldc	hp1	8,12	20,28	1083.90	548.31	333.58	202.01	36.32	0.0635	8	2	0	2	2
h206	ldc	hp1	4,29	8,5	1971.68	1196.61	547.56	261.75	3600.00	0.1449	7.4				
h325	ldc	hp3	11,1	21,3	919.29	414.15	340.08	165.06	10.10	0.0675	10.5	1	0	1	1
h362	ldc	hp3	2,5	4,3	2390.76	1475.03	521.95	253.12	3600.00	0.0679	6.5	5	3	1	5
h412	ldc	hp3	9,1	21,1	1358.63	551.27	545.61	261.75	1809.15	0.0919	9.5	2	1	2	2
hr04	dct	hp1	10,3	20,3	787.91	317.35	305.50	165.06	24.11	0.0303	10	1	0	1	0
hr07	dcp	hp1	8,1	18,2	855.09	290.47	394.83	170.59	355.25	0.0200	7	3	3	0	3
hr19	dct	hp1	9,2	18,2	911.46	432.37	322.66	156.43	31.72	0.0035	7.67				
hr21	dct	hp1	10,1	22,1	915.14	234.46	467.61	213.07	1714.72	0.0099	8.25	1	2	1	1
hr25	dcp	hp1	11,3	20,3	646.35	265.87	338.00	42.48	0.60	0.0028	8.67	1	0	0	0
hr25	dct	hp1	1,3	11,3	1177.71	544.51	402.87	230.33	2079.67	0.0165	5.4	4	3	1	3
hr32	dct	hp1	5,3	10,3	835.54	393.11	286.00	156.43	11.79	0.0033	7.33	1	0	1	2
hr33	dct	hp1	1,1	13,1	701.26	333.51	260.00	107.75	14.22	0.0015	3.33	4	5	0	3
hr34	dct	hp1	2,1	14,1	978.61	429.07	324.74	224.80	305.76	0.0077	7.33				
hr37	dct	hp1	5,2	7,2	916.49	446.12	299.78	170.59	18.09	0.0053	5				
hr40	dct	hp1	6,1	10,3	890.86	386.69	333.58	170.59	66.37	0.0038	6.25				
hr45	dct	hp1	1,1	2,3	1230.78	509.87	513.37	207.54	2175.71	0.0093	7	1	2	2	3
hr47	dct	hp1	1,3	13,3	1059.17	539.13	363.61	156.43	43.27	0.0093	8.33				
hr58	dco	hp1	1,2	10,2	941.77	491.76	325.00	125.01	4.92	0.0025	6.25	3	2	0	0
hr61	dco	hp1	12,1	14,1	1590.03	792.33	558.74	238.96	3600.00	0.0087	9.5	3	1	2	0
hr62	dco	hp1	10,2	22,2	1025.20	376.21	424.19	224.80	566.26	0.0074	9.67				
hr63	dco	hp1	12,2	24,2	1213.14	624.76	454.74	133.64	26.43	0.0076	11	1	0	0	0
hr65	dco	hp1	12,1	20,2	1423.86	670.53	537.16	216.17	91.69	0.0064	10	2	0	1	0
hr68	dco	hp1	5,1	16,1	1353.31	766.19	430.69	156.43	12.85	0.0025	9.33				
hr69	dco	hp1	12,1	16,1	809.01	311.37	364.00	133.64	3.79	0.0050	9.33				
r003	dcz	vp2	11,1	19,1	585.39	169.12	351.00	65.27	0.41	0.0063	9	2	0	0	1
r009	dcz	vp2	8,1	13,1	626.53	209.10	338.00	79.43	0.48	0.0041	6.5				
r019	dcz	vp2	9,1	21,1	764.39	322.12	377.00	65.27	1.67	0.0059	9.67				

RT	DC	Area	Wood Locations			Obj	Dist	Node	Wood	Running Time	Running Time LA	PR/WL	Aisle		Type		LowRatio	
			12,1	17,1	27,1								Switch	Switch	Switch	Switch		
r023	dcz	vp2	12,1	17,1	27,1	647.39	208.33	351.00	88.06	2.76	0.0024	9	1	2	0	0	0	
r030	dcz	vp4	10,1	20,1		520.60	232.28	260.00	28.32	0.38	0.0006	10						
r036	dcz	vp4	9,1	21,1	33,1	986.21	469.15	429.00	88.06	0.74	0.0055	11	2	0	0	0	0	
r038	dcz	vp4	10,1	22,1	34,1	897.91	413.43	442.00	42.48	0.46	0.0140	11.33	0	0	0	0	0	
r046	dcz	vp4	12,1	20,1		579.06	193.08	266.50	119.48	0.73	0.0009	10						
r050	dcz	vp4	9,1	20,1		529.31	218.20	260.00	51.11	0.29	0.0022	10						
r054	dcz	vp4	12,1	24,1	39,1	1407.95	635.25	556.53	216.17	90.19	0.0061	9.75	2	1	2	1	1	
r128	dcz	vp4	12,1	23,1		750.85	290.21	318.27	142.27	3.71	0.0027	11.5						
r159	dcz	vp4	1,1	13,1	25,1	1105.72	424.23	510.90	170.59	13.06	0.0022	9	3	1	1	0	0	
vr02	det	vp3	2,1	10,1	22,1	584.76	210.70	286.00	88.06	2.73	0.0012	7.33	1	1	0	1	1	
vr05	dcp	vp3	1,2	13,3	25,3	1185.85	533.10	468.00	184.75	49.79	0.0166	7.2	4	3	0	3	3	
vr08	det	vp3	7,1	18,1	28,1	751.54	299.48	364.00	88.06	0.90	0.0023	9.33	2	0	0	1	1	
vr17	dcp	vp3	11,3	23,3	33,3	862.09	206.29	476.58	179.22	168.08	0.0021	11	1	1	1	1	1	
vr18	dco	vp2	12,1	23,1	29,1	1065.81	477.22	463.58	125.01	5.36	0.0027	8.75						
vr20	dcp	vp3	9,3	15,2	16,3	854.22	427.00	325.00	102.22	6.09	0.0005	6.25						
vr21	dcp	vp3	12,1	23,1	28,2	698.14	291.66	364.00	42.48	0.50	0.0011	9.33						
vr32	dct	vp2	6,2	15,2	27,2	989.76	330.93	511.03	147.08	53.35	0.0085	9.25	1	3	0	0	0	
vr42	dct	vp2	10,1	22,1	26,1	582.66	202.18	338.00	42.48	0.51	0.0011	8.67	2	0	0	1	1	
vr51	dcp	vp2	3,1	14,1	22,1	1053.75	407.58	507.00	139.17	13.67	0.0156	7.8	4	1	0	0	0	
vr52	dcp	vp2	7,1	19,1	28,1	998.75	389.53	507.00	102.22	3.02	0.0070	9.75						
vr55	dcp	vp2	11,1	22,1	32,1	698.87	240.39	416.00	42.48	0.35	0.0038	10.67						
vr57	dcp	vp2	10,1	22,1		464.13	149.81	286.00	28.32	0.23	0.0004	11	1	0	0	0	0	
vr60	dcp	vp2	12,1	22,1	33,3	805.94	288.88	429.00	88.06	0.87	0.0110	11	2	2	0	1	1	
vr65	dco	vp3	12,2	23,2	35,1	816.05	318.57	455.00	42.48	1.16	0.0020	11.67	2	0	0	0	0	
vr66	dco	vp2	9,3	15,3	25,3	1361.44	642.53	520.00	198.91	232.83	0.0121	6.67	3	3	0	4	4	
vr67	dco	vp3	12,1	23,2		469.63	142.31	299.00	28.32	469.63	0.0016	11.5	0	0	0	0	0	
vr72	dco	vp2	1,3	12,3	15,3	712.92	228.17	322.79	161.96	90.44	0.0013	4.6	2	3	1	2	2	
vr73	dct	vp3	11,1	22,1		521.95	184.84	286.00	51.11	0.31	0.0010	11						
vr74	dct	vp3	11,1	23,1	31,1	1065.03	481.81	481.00	102.22	3.56	0.0060	9.25	3	1	0	0	0	
vr76	dct	vp3	8,1	20,1		470.21	181.89	260.00	28.32	0.17	0.0012	10						
vr78	dct	vp3	10,1	20,1		503.82	169.92	260.00	73.90	0.45	0.0014	10						

RT	DC	Area	Wood Locations			Obj	Dist	Node	Wood	Running Time	Running Time LA		PR/ WL	Aisle Switch	Type Switch	Type3 Switch	LowRatio Switch
			3,3	15,3	21,3						Time	LA					
vr85	dco	vp2	3,3	15,3	21,3	531.44	193.17	273.00	65.27	0.89	0.0004	7					
vr86	dco	vp2	10,1	15,1	20,3	772.37	367.94	325.00	79.43	3.30	0.0010	6.25	3	2	0	2	
vr89	dco	vp2	4,3	16,3	26,3	1273.77	587.86	515.32	170.59	48.93	0.0050	9.25	3	1	1	3	
rp01	ldc	hp1/ hp3	1,47	2,45	12,25	2922.86	1301.87	1114.80	506.24	3600.00	0.9355	7.6	9	4	4	9	
rp02	ldc	hp1/ hp3	39,5	51,49	62,50	2880.31	1454.80	950.69	474.82	3600.00	0.3901	6.27	9	8	3	8	
rp03	ldc	hp2	4,46	5,45	16,23	1002.20	461.40	407.16	133.64	4.45	0.0450	10	2	0	0	1	
rp04	ldc	hp2	50,50	52,50	60,50	941.33	413.53	394.16	133.64	8.08	0.0519	9.67					
rp05	ldc	hp2	4,1	15,1	27,1	1933.69	833.51	764.53	335.65	949.14	0.3293	9.33	4	2	1	3	
rp06	ldc	hp2	11,1	16,1	26,1	1,300.30	558.68	511.29	230.33	62.33	0.0425	7.8					

C Results Additional Wood Locations

Table 20 presents the results of the Resource Constrained Elementary Shortest Path Problem (RCESPP), where we added some additional wood locations, as described by Section 6.4. WL_{\pm} stands for the difference between the number of wood locations visited before and after adding the additional wood locations. Obj stands for the objective value (i.e. the total costs), where Obj_{\pm} is the absolute difference and $Obj\%$ the absolute difference between the objective values before and after adding the additional wood locations.

Table 20: Results of the RCESPP with additional wood locations.

RT	DC	Area	Wood Locations	WL_{\pm}	Obj	Obj_{\pm}	$Obj\%$	Time
hr62	dco	hp1	9,3 10,3 22,3 29,3	1	920.88	-104.32	-10.2%	0.0141
hr63	dco	hp1	4,3 16,3 17,3 28,3 33,3	2	1163.71	-49.43	-4.1%	0.0589
hr65	dco	hp1	11,3 20,3 32,3 40,3	0	1264.74	-159.12	-11.2%	0.0047
hr68	dco	hp1	5,3 16,3 18,3 28,3	1	1275.90	-77.41	-5.7%	0.0161
hr69	dco	hp1	12,1 16,1 28,1	0	809.01	0	0.0%	0.0034
hr58	dco	hp1	11,3 22,3 25,3	-1	932.69	-9.08	-1.0%	0.0015
hr61	dco	hp1	5,3 17,3 26,3 38,3	0	1519.89	-70.14	-4.4%	0.0082
hr07	dcp	hp1	8,1 18,2 19,2 28,2	0	855.09	0	0.0%	0.0032
hr25	dcp	hp1	12,5 15,5 26,5	0	554.31	-92.04	-14.2%	0.0024
vr51	dcp	vp2	4,2 9,2 14,2 22,2 31,2 39,2	1	937.06	-116.69	-11.1%	0.0328
vr52	dcp	vp2	7,2 19,2 28,2 39,2	0	881.37	-117.38	-11.8%	0.0027
vr55	dcp	vp2	11,1 22,1 32,1	0	698.87	0	0.0%	0.0024
vr57	dcp	vp2	10,2 22,2	0	447.75	-16.38	-3.5%	0.0009
vr60	dcp	vp2	12,2 22,2 33,2	0	760.41	-45.53	-5.6%	0.0038
hr04	dct	hp1	5,5 7,5 9,5 16,5 20,5	3	721.59	-66.32	-8.4%	0.0597
hr19	dct	hp1	9,2 17,2 23,2	0	911.46	0	0.0%	0.0136
hr21	dct	hp1	1,1 2,1 6,1 10,1 22,1 23,1 33,1	3	915.14	0	0.0%	0.0041
hr25	dct	hp1	1,3 11,3 17,3 26,3 27,3	0	1177.71	0	0.0%	0.0078
hr32	dct	hp1	5,3 10,3 22,3	0	835.54	0	0.0%	0.0018
hr33	dct	hp1	1,1 13,1 15,1 18,1 19,1 20,1	0	701.26	0	0.0%	0.0011
hr34	dct	hp1	2,1 14,1 22,1	0	965.61	-13	-1.3%	0.0145
hr37	dct	hp1	5,2 15,5 20,6	-1	902.37	-14.12	-1.5%	0.0027
hr40	dct	hp1	6,1 15,5 25,5	-1	864.96	-25.90	-2.9%	0.0016
hr45	dct	hp1	1,1 2,3 11,3 23,3 35,3	0	1230.78	0	0.0%	0.0031
hr47	dct	hp1	12,5 13,3 25,5	0	1031.86	-27.31	-2.6%	0.0046
vr32	dct	vp2	6,2 15,3 26,3 37,2	0	965.83	-23.93	-2.4%	0.0089
vr42	dct	vp2	11,3 14,3 26,1	0	568.39	-14.27	-2.4%	0.0017
vr02	dct	vp3	2,1 10,1 22,2	0	567.58	-17.18	-2.9%	0.0008
vr08	dct	vp3	7,1 18,1 28,1	0	751.54	0	0.0%	0.0046
vr73	dct	vp3	11,1 22,1	0	521.95	0	0.0%	0.0004
vr74	dct	vp3	11,2 23,2 31,1 37,1	0	970.96	-94.07	-8.8%	0.0077
vr76	dct	vp3	8,1 20,1	0	470.21	0	0.0%	0.0009
vr78	dct	vp3	10,1 20,1	0	503.82	0	0.0%	0.0005
r003	dcz	vp2	11,2 19,2 27,1	0	582.21	-3.18	-0.5%	0.0012
r009	dcz	vp2	8,2 13,2 25,2 26,2	0	626.53	0	0.0%	0.0024
r019	dcz	vp2	9,1 21,1 29,1	0	721.39	-43.00	-5.6%	0.0020
r023	dcz	vp2	9,1 17,2 27,1	0	622.10	-25.29	-3.9%	0.0048
r128	dcz	vp4	12,1 23,1	0	750.85	0	0.0%	0.0103
r159	dcz	vp4	11,3 13,1 25,1 36,1	0	1103.12	-2.60	-0.2%	0.0090
r030	dcz	vp4	10,2 20,2	0	447.18	-73.42	-14.1%	0.0008
r036	dcz	vp4	11,2 22,2 33,2	0	883.68	-102.53	-10.4%	0.0116
r038	dcz	vp4	3,2 15,2 23,2 34,2	1	789.65	-108.26	-12.1%	0.0042
r046	dcz	vp4	12,2 20,2	0	556.29	-22.77	-3.9%	0.0007
r050	dcz	vp4	11,3 20,1	0	506.18	-23.13	-4.4%	0.0005
r054	dcz	vp4	10,3 22,3 27,1 39,1	0	1404.91	-3.04	-0.2%	0.0112