University of Twente

Industrial Engineering and Management

Master Thesis

Evaluating the performance of the loading algorithm for Light Electric Freight Vehicles

Author:
J.D.G. Velthuis

First Supervisor:
dr. B, Alves Beirigo
Second Supervisor:
dr.ir. E.A. Lalla
Company Supervisor:
S. Drenth

March, 2024

CAPE GROEP

UNIVERSITY OF TWENTE.

PREFACE

After studying for six years at the University of Twente in my hometown of Enschede, this thesis will close this chapter of my life. I would like to thank some people whom I met during my time as a student, but also those who helped me with my thesis.

I would like to thank Breno Alves Beirigo and Eduardo Lalla-Ruiz for their time as my supervisors at the University of Twente. Without them, my thesis would not be as academic as it is. Their feedback improved my thesis and especially my figures.

I would like to thank Stefan Drenth for his role as supervisor at CapeGroep. I greatly value his guidance and the welcoming setting that I had at the company.

I look back fondly at the last six years. I made a great number of friends in Enschede, travelled across Europe, and did a board year at AEGEE-Enschede and now AEGEE-Brussel-Bruxelles. Many thanks to the following people: my family, the do-group HGDBB for all the amazing parties and for promoting student activism, my friends from AEGEE for all the travels and late-night talks, my fellow board members for the motivation to do a board year during Corona, and the Financial Directors of Comité Directeur 60 and 61 of AEGEE-Europe for allowing me to work on several projects and develop myself in various ways.

Job Velthuis
Brussels, 04/03/2024

MANAGEMENT SUMMARY

This thesis was performed at Company X in collaboration with the CapeGroep. Company X is a company that delivers both parcels and letters. CapeGroep is a consultancy agency specialising in IT solutions with low-level coding. Company X uses Licht Electrisch Vrachtvoertuig (LEVV)s to deliver parcels in urban areas. An allocation tool is developed to assign parcels to the compartments within the vehicle, solving a Bin Packing Problem (BPP). Currently, there is no manner to test the solution on its performance, but the employees on the work floor are not experiencing problems with loading the parcels as they can use their creativity to fit all the parcels into their assigned compartments. Company X wants to test the current allocation tool in a 3D manner with exact placement to better represent reality as the volume manner is a relaxed version of the BPP. Therefore, the following goal is defined:

Design a 3D placement model to show the best possible placement of parcels and use it to analyse the performance of the current allocation tool.

Context

As stated before, Company X uses the LEVVs to deliver parcels in urban areas. This is done to make the delivery process sustainable and social. These smaller vehicles have less impact on road users as they do not block as much as normal vehicles. The electric vehicles also produce less pollution. This makes the LEVVs very important in the Company X strategy.

To help load the parcels, an allocation tool was designed. The allocation tool assigns parcels based on their volume and the free volume in the compartments and not a specific location within that compartment. Several assumptions and requirements are made for the tool. Two assumptions are important. The first one is that parcels arrive in order for placement. Currently, the parcels arrive randomly but this is difficult for planning. The second assumption is that there are no limitations on where in the vehicle a parcel is placed based on attributes like fragility. Three requirements are important. A customer cannot be split over several trips. The parcels belonging to a customer must be placed as close as possible to each other. Lastly, Parcels must be placed in the delivery order with the first customer being in the top left compartment and the last customer in the bottom right.

Method

The BPP is a well-studied problem. Many heuristics have been developed to solve the problem. To determine the best approach a literature review was done. The approach was chosen based on the similarities between the problem in the article and the thesis. The important aspects were the inclusion of Last-In-First-Out (LIFO), rotation of parcels, and the goal of the problem. The literature showed that the problem of Company X is best solved with a Deepest-Bottom-Left (DBL) approach. The DBL approach does not take optimal rotations into account and is biased to the parcel's initial rotation or orientation. To account for this a second approach was made, the Residual Spaces.

The model is built on the same basic principles as the current allocation tool. The company requires that customers are placed in order of servicing in the route with the first customer

being in the top left compartment and the last in the bottom right. The placing is done per column, so that column 1 is filled first and last column 3. The parcels of a customer must be placed as close to each other as possible. Lastly, the tool should be able to filter out parcels and customers based on a set of company exclusion rules for maximum dimensions, weight, and number of parcels per customer. These rejected parcels are moved to an overflow route to deliver the parcels in another manner. Parcels can also be allocated here if not all of them fit within the roll containers for all 3 trips. Then, the last customer in the delivery order is moved to the overflow route and the algorithm can try again to assign or place all parcels.

The placement within the compartment is done by initialising the first placement as in the DBL point in the container. Every new Extreme Point (EP) is created by taking the corner points of the placed parcel and sorting them to have the one that is DBL as the first in the list. This list is the new set of possible placements. The DBL accepts the first valid placement. The Residual Spaces approach accepts all valid rotations for an EP, then calculates the Residual Space and chooses the minimum value to be the best placement and orientation/rotation of the parcel.

Results

This model was validated with 3 datasets and all yielded good designs with good runtimes, (<3s). Next 289 scenarios are used to evaluate the performances.The data comes from five different cities and is from a period of four weeks during which no major public holidays happened. Three experiments were designed to test feasibility, the impact of problem size, and a comparison between the current tool and the heuristics. The last experiment was created where the 3D heuristics were applied to the solution of the current tool. This is to ensure the comparison is fair and not compare a relaxed version with a strict version.

Four Key Performance Indicator (KPI)s are set up, 1. run-time, 2. utilisation rate, 3. overflow rate, and 4. costs to evaluate the solution performances. For all KPIs, the current tool outperformed the heuristics. However, the feasibility experiment shows that none of the solutions were feasible if 3D and LIFO were taken into account.

The Residual Spaces approach outperformed the DBL approach in every KPI with significance. The significance was tested with ANOVA tests. All the solutions of the heuristics are feasible. The overflow rate shows that in the current tool, almost no parcels were scheduled to the overflow route due to not fitting. This is not possible for the heuristics. The heuristics did not schedule many parcels to overflow routes on average. The costs were on average all close to each other due to the method of cost calculation used by Company X. Figure 1 shows that the utilisation rates are close to each other, but that the current tool outperforms the heuristics.

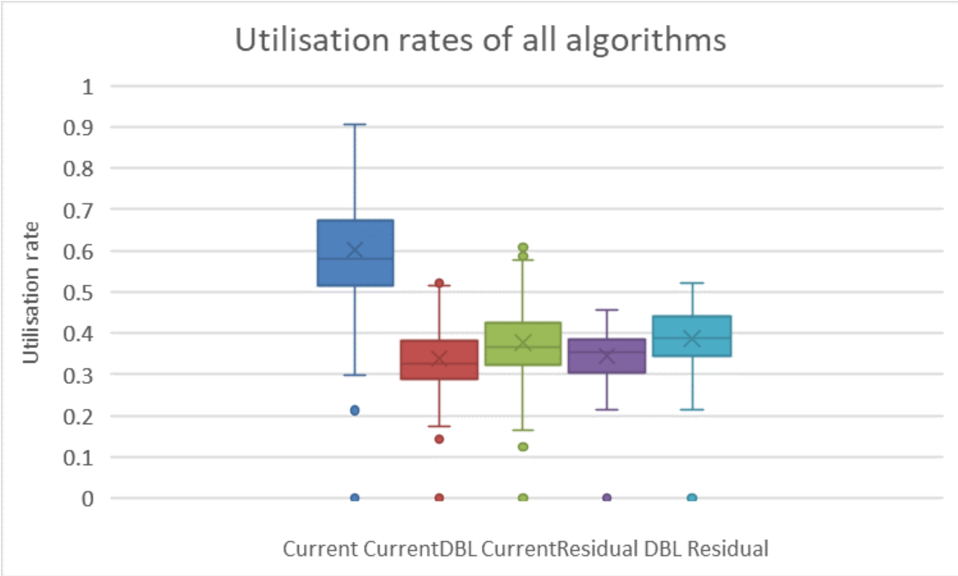


Figure 1: The utilisation rate of the current allocation tool is much higher than the heuristics and their applied cases. The Deepest-Bottom-Left (DBL) application is lower than the Residual Spaces

Conclusion

The first conclusion is that the current allocation tool does behave like a relaxed version of the BPP due to its high performance relative to the heuristics and the applied versions, but does not create feasible solutions. Workers therefore have to be creative in the placement of parcels within the allocated compartments to ensure feasibility. The Residual Spaces approach performs better in terms of utilisation rate, overflow rate, and costs than the DBL version as it optimises over rotation and orientation of parcels. Figure 1 shows that the utilisation rates support these conclusions. Experiments show it has a 2 percentage points less overflow rate, 4 percentage points more utilisation rate, and 0.5 percentage points cheaper. Secondly, the heuristics use more trips and compartments to allocate all the parcels than the current tool. This increase is more than 100% of the current trips and compartments that are currently used. Thirdly, the costs show again that the current tool is better as it has a lower overflow rate, and that the Residual Spaces is better than the DBL. The current tool is 0.2 percentage points cheaper than the Residual Spaces approach ad 0.7 percentage points cheaper than the DBL approach. Lastly, it is known that the current allocation tool provides an infeasible solution but the people on the work floor make the solution work. This means an increase of about 22 percentage points.

Concluding this thesis, the current allocation tool could perform better when looking at it from the 3D heuristics perspective. These solutions are not feasible when checking for 3D and LIFO constraints. However, as personnel on the work floor do not experience difficulty packing the parcels, it is not recommended to switch to a 3D approach. When trying to have a similar utilisation rate, a lot more complexity needs to be added to the algorithms or company rules need to be changed to allow for more freedom in placing parcels. Secondly, the switch is not recommended from a human side. Humans cannot place a parcel on a millimetre exactly. If Company X would change to use robots when loading the roll containers, then it would be helpful to change to a 3D approach and this thesis would recommend the Residual Spaces approach.

Abbreviations

ALNS Adaptive Large Neighbourhood Search.

BPP Bin Packing Problem.

CPP Cutting and Packing Problem.

DBL Deepest-Bottom-Left.

EMS Empty Maximal Spaces.

EP Extreme Point.

GA Genetic Algorithm.

KPI Key Performance Indicator.

LEVV Licht Electrisch Vrachtvoertuig.

LIFO Last-In-First-Out.

MILP Mixed-Integer Linear Programming.

MIP Mixed Interger Programming.

MLIFO Manual Last-In-First-Out.

CONTENTS

1 INTRODUCTION

This chapter will introduce Company X and the experienced problem, the research questions this report will answer and the methodology used for that. In Sections 1.1 and 1.2, the company Company X and its parcel delivery process will be described. In Section 1.3, the problem Company X is experiencing will be explained. Section 1.4 details the research goal, problem, and questions. Lastly, in Section 1.6 the research approach is presented.

1.1 Company description

Company X focuses on providing sustainable services that benefit customers, the environment, and investors. Company X wants to achieve this by digitizing its services, improving working conditions, and reducing its carbon footprint. As part of the efforts, electric vehicles are used for last-mile delivery, resulting in a reduction of CO_2 emissions of 18% per kilometre. The benefits of electric vehicles show the possibility for this thesis to improve the processes surrounding them.

1.2 Parcel delivery process

Company X makes use of the Licht Electrisch Vrachtvoertuig (LEVV) or in English, Light Electric Vehicle to deliver parcels in the city centre of four cities. They do this to make their process more sustainable and have less impact on social life.

Company X had to change the process for loading parcels as the vehicles are smaller and require more attention for loading. Figure 2 shows the changed process for a brief overview.

Company X determined that parcels need to be allocated a place in the LEVV via a digital tool. The LEVV is loaded with 3 roll containers in which the parcels can be placed. Parcels that cannot fit are removed from the vehicle packing list and are allocated to an overflow route. This overflow route is driven by a normal van and does not have a specific set of customers as it can change daily based on the parcels that do not fit.

The digital packing list is sent to the depot. Here the employees will use it when parcels arrive at the loading docks. Employees know where to place the parcels because they have a glove that functions as a scanner. When they scan the parcel it will show a placement code.

The employees will load the parcels into roll containers first and not right into the vehicle. This is because the loading can be done in a different place than where the vehicle is parked. Sorting depots are often outside of cities and the LEVV might not have the range to drive that distance.

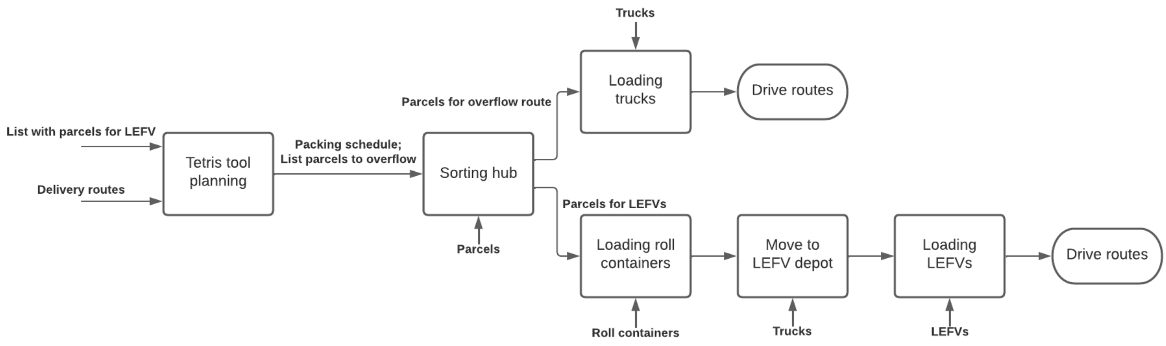


Figure 2: Process flow chart of the delivery process.

1.3 Problem description

This section will give the problem description. Company X has the feeling that the digital allocation tool is underperforming. A brief data analysis is done to detail the current performance. However, Company X does not have a formal method to test the performance against a more theoretical optimum.

1.3.1 Data analysis

The performance data is analysed to better understand and explore the problem. The available data and experience from Company X show that there is volume in the trips and their roll containers left that could be filled with parcels. From talking with Company X the indicators are picked. They think that when manual changes happen to the solution provided by the allocation tool, it is not working properly.

The data is from 4 weeks, resulting in 49,068 parcels that need to be allocated. The parcels and their routes are then analysed for placement and utilisation.

Table 1.1 shows that 19% of the parcels presented to the allocation tool were scheduled to the overflow routes. This is a very high percentage as all these parcels have to be delivered outside of the normal routing, resulting in higher process costs. This overflow scheduling is entirely caused by the internal parcel and customer exclusion rules

Number parcels	#	%
Presented to system	49,068	100%
Scheduled normal routes	39,822	81%
scheduled overflow	9,246	19%

Table 1.1: 19% of all parcels presented to the current allocation tool are moved to the overflow.

Internal rules state that LEVVs cannot be filled more than 95%. Analysis shows that on average only 67% is used. The utilisation rate ranges to around 50% at the low end. This all indicates that improvement is achievable to close the gap between the average and the internal maximum.

Overall, the results show that there are opportunities for improvement which would improve the performance of the current allocation tool. However, there is no formal method to see how much improvement is still possible.

1.3.2 Problem statement

Since the development of the allocation tool, Company X has not created a method of evaluating the solution performance. Company X is experiencing under-utilisation of its resources and over-utilisation of the overflow routes. Therefore, they want to improve the allocation tool but do not know what improvement in performance is possible. Therefore, the problem of the report is defined as:

Company X has developed an allocation tool to help allocate parcels for loading but cannot assess the solution quality.

1.4 Research goal

Company X does not have a method to evaluate the performance of the allocation tool in terms of placement. The allocation tool assigns parcels based on volume but this ignores the dimensions of parcels. Assuming parcels fill a compartment like water leaves the existence of not usable empty spaces out of consideration which could have a real impact on the loading. Therefore, this thesis will design an approach where parcels are placed based on dimensions. The 3D model will be based on current approaches and improvement heuristics. The evaluation is then the solution performance difference between the optimal solution of the 3D model with placement and the current allocation tool solution modified to be 3D based for fair comparison.

Design a 3D placement model to show the best possible placement of parcels and use it to analyse the performance of the current allocation tool.

1.5 Research problem

As discussed in the previous section, the allocation tool is already functioning. This research is thus explanatory to give further insights into the allocation tool and, with that, allow for management to make more informed decisions on further usage and improvement of the existing application to improve performance. The main research question is as follows:

How well does the allocation tool perform compared to a best possible packing plan based on 3D heuristics possible for Company X for the LEVV vans?

Multiple research sub-questions are formulated in this section to answer the research problem. These sub-questions will explore the current process and proposed methods in the literature that can be used to make an exact placement approach for the parcels, design the solution approach, and design experimentation and validation methods.

Current situation
The first question focuses on exploring the current approach Company X is using for this problem. Here, the requirements, limitations, and available data are classified. This leads to the following questions:

i. How does Company X currently load their LEVVs?
   a. *What solution approach is now being used to load the LEVVs?*
   b. *Which assumptions are being made by Company X for the packing plan?*
   c. *What requirements are posed upon the solution in the allocation tool?*

d. *Which Key Performance Indicator (KPI)s are relevant to Company X for the delivery performance analysis of the LEVVs?*

e. *What data is available regarding the performance of LEVVs?*

Literature review
To develop a 3D packing strategy, it is useful to look at the literature to find out what methods are proposed. To start from a fresh point of view, an overview of all relevant methods for solving a loading model will be made together with an explanation of the fundamentals of a loading problem. Following that, a more detailed search is performed to discover what constraints exist in loading problems. Lastly, a conclusion is made about which method is the most relevant to the research problem. This leads to the following questions:

ii. What does the literature propose to solve a 3D loading problem?

a. *Which loading plan methods are used to solve a 3D loading problem?*

b. *What practical applications are there for the 3D loading problem?*

c. *Which constraints are proposed in the literature to simulate real-life limitations?*

d. *Which 3D loading plan methods are most fitting the problem of Company X?*

Design of solution approach
Following the literature search, it is necessary to bring the knowledge into practice to design a 3D placement approach. The practical side is how the current process at Company X can translate into a mathematical model in terms of requirements and the data needed to make a loading schedule. This leads to the following questions:

iii. *How should the 3D packing approach be designed?*

a. *What are the functional requirements of the model?*

b. *What data is needed to have an effective loading schedule?*

c. *How can the current allocation tool be modified to a 3D model to assess the current solution quality*

Evaluation
After the 3D model has been made, the performance should be evaluated. Therefore, questions have been defined to evaluate and validate the developed solution for different scenarios or set-ups:

v. *How does the allocation tool compare to the 3D model in terms of performance?*

a. *How can the allocation solutions from the allocation tool be validated?*

b. *Which design should be used for the experiments?*

c. *How can the allocation tool be compared to the 3D heuristics in a fair way?*

Conclusions and recommendation
Finally, to conclude from the experiments and their evaluation the following questions were defined:

vi. *What are the conclusions and recommendations to Company X?*

a. *How does the allocation tool perform compared to the 3D model?*

b. *What recommendations can be made to Company X based on the experiments?*

c. *What future research can be done based on the results of this research?*

1.6 Research approach

In this section, the overall approach, scope, and limitations of this thesis are discussed. For this thesis, the scope is defined as a boundary set on the project by choice. Limitation is defined as real-life limits that the thesis cannot influence. In the first part, the scope of this project is defined. This limits what will and also what will not be discussed and evaluated in the research. After that, the limitations of the report are discussed. This part is about what barriers the research experiences and will not be considered later on. The Design and Methodology section discusses the build-up of the report itself and what methods will be used to develop the tool.

Scope

The following list details the scope of this thesis:

- The most important scope is that this thesis will stay as close as possible to the set-up of the and decisions made in the current allocation tool. Therefore, the same grouping and sorting choices for parcels are being made here. The grouping determines that all parcels belonging to one customer must be placed as close together as possible and the sorting within the customer is done based on the volume of each parcel, starting with the highest volume in the placement order. No further scenarios other than 3D placement and the impact of orientation will be researched as all other scenarios would stray too far from the current use case. Company X knows their parcels and it determined that their parcels are strongly heterogeneous in nature of dimensions.
- This research will only design a 3D placement tool that makes a theoretical optimum. All real-life constraints except dimensions and volumes will not be considered.
- The delivery personnel is prone to violating the current allocation tool solution of the roll containers. This is also outside the process of how Company X imagined it. Besides that these violations can lead to dangerous circumstances for the driver if they have to brake hard and also possibly harmful for the parcels. As these violations are deviations and only take up time, the research will only try to evaluate the allocation tool solution and the 3D model.
- The research does not explore the performance of the routing of the LEVVs. This is because Company X is already using a heuristic that allows them to have stable usage of their vehicles and personnel. Another reason is that the limited time window Company X has during the days for planning and scheduling the delivery routes leaves little time to solve the often-large routing problems.
- The research does not consider deliveries to rural or low-density areas. This is because LEVVs do not service these areas yet.
- The research focuses on packing vans with parcels to emulate the LEVVs and not stocking a combustion van. This is due to the reason the allocation tool is specifically designed for the LEVVS.
- Parcels arrive in a random order at the packer via a shute. This has an impact on how efficiently the packer can follow a packing solution. The randomness of the arrival of the parcels will not be taken into account for the 3D model.

Figure 3: Roll container with foldable floors

Limitations
The following limitations are impacting the project:

- The parcels have to be ordered in the delivery order within the roll containers from top to bottom.
- The packing schedule must comply with Last-In-First-Out (LIFO) principles as no repacking is allowed during the delivery process.

1.6.1 Design and methodology

The research is split into four phases. They are grouped to form the research design. This will answer the main research question by answering sub-questions in each phase. The first phase answers the first two sub-questions and the other phases answer one each. The phases are: 1. Context analysis and literature study 2. Solution design 3. Solution experimentation and evaluation 4. Conclusion and recommendation

2 CONTEXT ANALYSIS

In this chapter, will answer the research question, *How does Company X currently load their LEVVs*. The current situation is explained in three parts. Section 2.1 will explain the vehicles that are used for the delivery process. In Section 2.2, the allocation tool that is used is explained. Section 2.3 is expanding the data analysis of Chapter 1 to gain more insight into the performance. Section 2.4 will detail the KPIs that are shown to be relevant from the previous section. Lastly, Section 2.5 will give insight into the experiences from the employees when working with the allocation tool.

2.1 Light Electric Freight Vehicle description

Since 2020, Company X has introduced the LEVV to deliver packages to customers in urban areas, both commercial and private. The LEVVs play a big part in Company X's desire to deliver sustainably and socially. From a law side, it is also important to become more sustainable with electric vehicles. Nowadays, environmental policies in city centres are common. These policies discourage the use of combustion engine vehicles. 35% of CO_2 and 10% of particulate emissions in cities are caused by freight traffic (Otten et al., 2015). The LEVVs have a positive effect on the social experience of road usage. Due to their smaller size, the vehicles use less space on the road or pavement when they are parked for delivery. The size allows for easier and safer passing of other road users that would otherwise have to wait due to the narrow roads in historical city centres. Another improvement is that electric vehicles are relatively silent in comparison to current combustion vehicles. In Dutch cities, freight traffic is 20-25% of the road traffic, so LEVVs can have an impact. Another benefit is the experience of the delivery personnel. The small size of LEVVs means they can park closer to the customers' door. This minimises the walking distance which is especially beneficial when heavy or big parcels are delivered. There are also downsides to this type of vehicle. They have less loading capacity and limited speed and range due to recharging requirements. Another problem is that the LEVVs are not allowed to drive on highways which are often needed to reach the bigger distribution centres on the outskirts of the towns(van Amstel et al., 2018).

Figure 4: Small electric delivery van

The vehicles used are small electric vans, as depicted in Figure 4. The build-up of the LEVV is as follows. The vehicle has a cabin in the front where the driver and a passenger can sit. As can be seen in Figure 5, the back is a loading compartment that can be accessed from both sides to unload parcels and from the back to load the roll containers. The loading compartment can fit 3 roll containers at a time. The loading compartment is not filled with the containers as there are wheel wells on both sides together with the chassis which blocks easy access to the lowest compartments of the roll containers. A bar in the loading space of LEVVs is in the way of the top compartments of the roll containers. If a parcel does not fit a compartment in a roll container, then it has to be delivered with a combustion engine vehicle.

Figure 5: The loading space of a LEVV

2.1.1 Delivery process description

The process of parcel delivery is split into three parts: 1. allocating parcels in a digital environment, 2. loading the physical parcels, and 3. delivery. The digital allocation is done first. The resulting schedule is used later by the loading personnel. Figure 2 shows a quick overview of the process and its inputs. The LEVVs have two types of routes:

- Regular: routes that are scheduled regularly with set areas that they cover.
- Overflow: routes that are scheduled irregularly and do not have a set area that they cover but change every time.

When parcels cannot be placed in the regular route, they are placed in the overflow route. This route functions like a buffer for Company X.

Digital allocation tool

The allocation is done by an allocation tool referred to as the Tetris tool by Company X. For this thesis, the tool is referred to as the current allocation tool. This allocation tool sorts and allocates parcels. The process starts with receiving the delivery routes and a list of parcels that need to be allocated. The allocation tool filters out parcels that do not fit in the LEVVs because of size, weight, or limited loading space and allocates them to the overflow routes. The leftover parcels are allocated to a specific compartment in a roll container. Every roll container can have several compartments, with a minimum of one and a maximum of three. The list of parcels for the overflow route and the allocation of the other parcels are given to the loading personnel in the sorting hub.

Loading process description

The loading process is divided into two parts; the first happens in the evening and night, and the second in the morning.

In the evening and night, the parcels are sorted to move them to their destination region and the 33 distribution depots. Of the 33 depots, 27 are for parcel distribution and 6 for letters.

In the morning, the team leaders of the loading personnel run the allocation tool and check the allocation schedule generated. Some routes may be too empty because many parcels were excluded because of company rules by the allocation tool. Then, the diverted parcels will be manually placed back into a roll container. They can do that if there is a need to rerun the allocation tool. This happens when customers are manually added or removed from the route. Sometimes, roll containers cannot be filled due to too little demand. No parcels will be switched from vans to LEVVs to distribute the workload. The delivery routes are optimised to minimise time spent on driving. Moving customers and parcels between routes can negatively impact the time usage.

Parcels are loaded onto the conveyor belt from the depot loading dock. On the belt, the parcels are photographed and then put on an O-shaped conveyor belt that transports parcels to chutes. Every route has its chute. An employee working the chute manually loads the parcels into the vans or for the LEVVs roll containers. The parcels arrive in random order and at relatively high speeds, as the number of parcels handled at a depot is often around 8,000 parcels per hour.

The employee loads the parcels according to the allocation schedule of the allocation tool. The employees at the sorting depot use gloves that have a scanner and a screen. They scan the parcels with the glove that then displays the compartment code where to put the parcel. Only a compartment is displayed to the loading personnel. They are free to determine the placement of the parcel within the compartment. There are best practices on how to place the parcels. One of them is to put the hard cardboard boxes at the borders of the compartment and the soft bagged parcels in the middle as filler. It is sometimes needed to repack the loaded parcels to allow for more efficient packing due to the random arrival order. The parcels are loaded into roll containers that are shipped to the LEVV depot closer to the city centre for delivery.

Delivery process description

The loading of roll containers for the LEVVs is normally done by a different employee than the one who delivers the parcels to their destination. That is because the LEVVs are not able to bridge the distances between the sorting depots and their delivery routes due to the aforementioned reasons.

The delivery personnel does not know where all packages are located. The deliverer needs either a general rule to follow or should be given the information. The current heuristic generally allocates the parcels in the delivery order to the compartments. The first package to be delivered should be in the top left compartment and the last one in the bottom right. Company X uses handheld devices to sign for parcel deliveries by customers. These devices are also used to tell in which compartment the parcel is located. The device makes it possible to move away from the heuristic, allowing parcels to be stored in any compartment.

The delivery personnel receives the roll containers at the place where the LEVVs are parked. The deliverers consider their experience with loading very important. When they receive the roll containers, they sometimes repack the roll containers. The delivery personnel will also unload roll containers into their passenger seat or the sides of the LEVVs next to the roll containers in an attempt to save time on extra trips that are not fully loaded. Unpacking the ideal packing costs time. Experiments by Company X have shown that repacking does not improve capacity usage or save delivery time. It is more efficient to immediately drive the routes and not repack. Another downside is that the parcels are no longer in the compartment shown on the handheld device. This makes finding parcels more difficult and costs time to search.

The LEVVs are loaded with the roll containers and start their rounds. When the drivers want to start their route, they have to get clearance from the depot. This is to have security control over where the parcels are in the process. The clearance will change the location of the parcel from the depot to the van. During the route, the delivery person also picks up packages from stores or collection points to make more use of the route. This is, however, left out of this research as it is not of importance for the morning shift packing the roll containers. At the end of the route, a debrief must be done by the deliverer to restate the location of parcels that were not delivered.

2.2 Allocation tool analysis

The tool is an application made in Mendix, a low-code platform that is used for web apps. Low-code means that coding is done mostly with a graphical user interface. The tool is responsible for the roll containers' packing schedule and the parcels' pre-sorting and is a back-end application, also known as a microservice. The back end is meant to be visited as little as possible. The people who need the tool's output call the logic via a different application called OOM PD, which is not in this project's scope. The users have the input and output of the tool in OOMPD.

The settings of the current allocation tool are split into four groups: 1. general, 2. standard dimensions, 3. maximum dimensions, and 4. container. The general settings are:

Used data description

The relevant data Company X has on the performance of the tool can be divided into 3 parts, the parcel, the trip, and the route data.

On the parcel level, there is data on the following:

- The delivery order number of the customer.
- The volume of the parcel.
- Which compartment the parcel is allocated to? See Figure 6 for the build-up of the compartments.
- Which trips the parcel is allocated to?
- If the parcel is allocated.
- The reason why the parcel is not allocated.

Figure 6: The roll container contains three compartments where the top two have the same dimensions and the bottom one is bigger.

On the trip level, there is data on:

- The number of parcels present in the trip.
- The utilisation rate of the trip, i.e. the percentage of volume used.

On the route level, there is data on the following:

- Number of parcels initially allocated to be on the route.
- Number of parcels moved to the overflow route.
- Percentage of parcels that are allocated to the route.

Assumptions and requirements of the allocation tool
Assumptions are being made in the current tool. They can be split in 3 groups for readability: 1. parcel dimensions 2. exclusion rules 3. allocation logic.

Company X assumes that the measurements must be linked to a dimension based on their value. They also assume that it makes the placement by the allocation tool easier. The tool receives a list of parcels with their measurements as a set of unordered values. The measurements have to be linked to a specific dimension, i.e. height, length, width. The logic behind the linking approach is shown in Algorithm 1. That algorithm reads the values of the measurements and links them to a specific dimension of the parcel or if the measurements are not complete then the standard measurements are used. The following assumptions are made:

1. the smallest measurement is the height, the biggest is the length, and the other is the width.
2. In case the measurements are unknown, the tool will assign an assumed measurement based on an average parcel, in this case, a shoebox.

Company X does not want all parcels or customers handled by the LEVVs. They defined exclusion rules for parcels and dimensions. Algorithm 2 shows how the exclusion rules are implemented and filter the parcel and the customers. The exclusion rules are:

- There maximum number and volume of parcels allowed for one customer. Large customers can be better serviced by traditional vans.
- Parcels that cannot be machine-handled are excluded.
- Parcels cannot be bigger than a manually preset maximum. The dimensions are a manual input.

Lastly, the tool has some assumptions to simplify the allocation algorithm. Note that a route is the complete set of all customers that must be visited. As the volume of the parcels of all the customers is likely to exceed the volume of the LEVV, Company X decided that a route can be split into 3 ”trips”. Where each trip starts at the depot, visits several customers and then return to the depot:

- A route can only be serviced in 3 trips.
- The tool does not take the measurements or orientations of the parcels into account when assigning them to a compartment but treats it like filling a container with volume.
- The fragility of the parcel is unknown to the allocation tool, so it is not considered but only done when packing the vehicle or roll container.
- Fragile items can be placed anywhere.
- The compartments in a roll container are filled to a preset utilisation rate.
- Random arrival order is not taken into account and customers and their parcels are allocated as if the arrival order is known and ordered.
- Compartments are not tested for placement for customers that are later in the placement order. If a parcel does not fit in a compartment, then the compartment is considered closed.

For Company X, there are several requirements for a valid loading pattern. They are also required to stay as close as possible to the current tool for a fair comparison. They are:

- A solution is feasible when all parcels are loaded fully in the compartments.
- The customer’s parcels must be loaded into one trip.
- A customer can only be visited once. If not all parcels belonging to a customer cannot fit into the trips, the customer will be moved to the overflow route.

- Customers must be serviced in the predetermined order.
- Parcels belonging to a customer must be placed near each other in the same compartment or if not possible in the next.
- Customers and their parcels must be placed in such a way that they are in the delivery order and their placement must be so that the first customer in the trip is located in the first roll container in the top compartment, and the last customer in the last roll-container in the bottom compartment.
- The solution must be LIFO compliant and no repacking must be done during the delivery or loading.

Algorithm 1: Function to clean up and standardise measurements of parcels

Data: $ListOfParcels$
Result: Allocation and packing schedule

Function `UpdateParcelDimensions`($ListOfParcels$)
    forall $ParcelinListOfParcels$ **do**
        Read the current dimensions of the parcel
        if $Anydimensions = 0$ **then**
            Assign preset dimensions for Height, Length, and Width
            Break
        Make a list of dimensions
        Sort the list from large to small
        Update Length, Width, and Height attributes of the parcel by the largest to smallest value respectively
    return $ListOfParcels$

Algorithm 2: Function to filter out customers and parcels that do not fulfil company exclusion rules

Data: $ListValidCustomers$

Result: Allocation and packing schedule

Function `FilterCustomerList`($ListOfCustomers$)
    $ListValidCustomers = \emptyset$
    forall $CustomersinListOfCustomers$ **do**
        Count all parcels of the customer
        if $NumberParcels > NumberParcelsAllowed$ **then**
            Break
        Sum all parcel volumes of the customer
        if $SumVolumeParcels > TotalVolumeAllowed$ **then**
            Break
        forall $ParcelsinParcelsOfCustomer$ **do**
            If any parcel is not allowed then the customer is rejected
            if $ParcelAttributeNMG = True$ **then**
                Break customer loop
            if $AnyDimension > Allowed$ **then**
                Break customer loop
        Customer passes all criteria
        Add $Customer$ to $ListValidCustomers$
    All removed customers are moved to overflow routes
    return $ListValidCustomers$

Logic of the current tool

The functioning of the allocation tool starts with receiving a request from OOMPD. The logic of the allocation tool is shown in Algorithm 3. Firstly, the stops are filtered regardless of what route they are in. The filtering is done by checking if the stops have too many parcels, too much volume, or if any parcel cannot be serviced by the machine. The valid stops are kept. A list of all the stops and parcels that should be allocated is made per route. Then, a loop is done over the trips within the route. Per stop, several steps are done. First, an estimate is made if the stop will fit. If the estimation says it will not fit, then the stop is kept for the next trip. If the stop does fit, then the parcels are allocated to compartments. If it turns out that a parcel cannot be allocated then the stop is moved to the next trip. If the stop is successfully planned, the compartments' remaining volume is updated. The list of stops to plan is also updated by removing the successfully planned stops and parcels. When the loop is done, a check is performed. If not all stops have been planned, then two things can happen. If the users check the ”Exclude large parcels” option, the stops with parcels bigger than the mentioned dimensions are moved to the overflow route. This button is not yet used. The whole allocation loop is performed again with the modified stop and parcel list. If the ”Exclude large parcels” option was not checked, the stops not initially planned are moved to the overflow route. The idea of what happens is the same, but different stops and parcels will be moved to overflow depending on the checkbox. At the end of the process, an export to XML is made as an output to OOM PD.

The allocation tool allocates parcels based on a Top-Left-Fill heuristic that uses very simple logic only using the volume of parcels. When a parcel does not fit anymore in a compartment, that compartment is closed, even if a smaller parcel that still needs to be allocated can fit there. The company does not want to change the packing order regarding customer orders. The improvement will therefore not be implemented but mentioned to show that improvement is possible if they want to change their approach.

Algorithm 3: Main logic of the allocation tool

Data: $ListOfCustomers, ListOfParcels, TripList, CompartmentList$
Result: Allocation and packing schedule

begin
    CT := $CurrentTrip$
    t = 0
    CC := $CurrentCompartment$
    c = 0
    $CT \leftarrow TripList[t]$
    $CC \leftarrow CompartmentList[t, c]$
    $CT_{\text{FreeVolume}} \leftarrow CT_{\text{TotalVolume}}$
    $CC_{\text{FreeVolume}} \leftarrow CC_{\text{TotalVolume}}$
    $ListOfParcels$ = `UpdateParcelDimensions(`$ListOfParcels$`)`
    $ListValidCustomers$ = `FilterCustomerList(`$ListOfCustomers$`)`
    forall *Customers in ListCustomers* **do**
        if $Customer_{\textit{Volume}} > CT_{\textit{FreeVolume}}$ **then**
            t = t+1
            c = 0
            $CT \leftarrow$TripList[t]
            $CT_{\text{FreeVolume}} \leftarrow CT_{\text{TotalVolume}}$
            $CC \leftarrow CompartmentList[t, c]$
        Allocate customer to current trip
        $CT_{\text{FreeVolume}} \leftarrow CT_{\text{FreeVolume}}$ - $Customer_{\text{Volume}}$
        forall $ParcelsinParcelsOfCustomer$ **do**
            if $Parcel_{\textit{Volume}} > CC_{\textit{FreeVolume}}$ **then**
                $CT_{\text{FreeVolume}} \leftarrow CT_{\text{FreeVolume}}$ - $CC_{\text{FreeVolume}}$
                c = c+1
                $CC \leftarrow$CompartmentList[t,c]
                $CC_{\text{FreeVolume}} \leftarrow CC_{\text{TotalVolume}}$
            Allocate parcel to compartment
            $CC_{\text{FreeVolume}} \leftarrow CC_{\text{FreeVolume}}$ - $Parcel_{\text{Volume}}$

Logic example

To explain the current approach with a simple example: Imagine a roll container with two compartments, viewed from the front. The strategy is shown in Figure 7.

In the strategy, the idea is to load parcels into the compartments. The rule is to start by placing the parcel on the left side and at the bottom of the compartment. First on the compartment floor, and then as low as possible within the compartment.

Now, consider two customers, customer 1 and customer 2, with a total of 3 packages each labelled 1a, 1b, 1c, 2a, 2b, and 2c. These packages come in two sizes: small and large. We load them in the order 1a, 1b, 1c, 2a, 2b, 2c.

In Figure 7, a compartment is ”closed” as soon as the first parcel in the loading queue does not fit anymore. This means that if a parcel does not fit in a compartment, that compartment is not used anymore.

The roll container is only used to 75% of its capacity because compartments are closed as soon as a parcel does not fit.

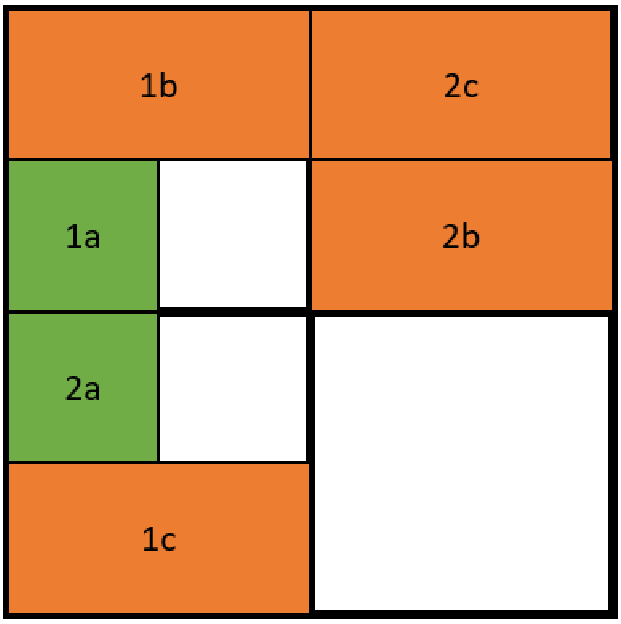


Figure 7: Current heuristic, close compartment when the first parcel in packing order does not fit

2.3 Performance analysis current allocation tool

This section is an extension of Section 1.3.1. The same data set is used for this analysis. To evaluate the indicators, another data set is used, called “Released Routes‘” within Company X. This data set contains information on what parcels were loaded into which routes. This data set is needed for the analysis. Data was filtered by removing all parcels that were not present in both sets. This leaves 44,952 parcels in the allocation tool data set and 44,864 in the ”Released Routes” data set.

Firstly, the reason why parcels are scheduled to the overflow route is analysed in two ways: the rules on exclusion, in Algorithm 2 and based on volume. Exclusion rules are limits set by the company. They include maximum dimensions and volumes of parcels, and the maximum number of parcels per customer. For parcels and customers that violate these limits, it is better to deliver them with bigger vehicles. The rule-based exclusion is analysed first. This is done by comparing the number of parcels moved to the overflow based on the rules by the allocation tool in a route to the total number of the parcels scheduled to the overflow. Table 2.1 shows that all parcels were scheduled to the overflow route by the allocation tool because of the company exclusion rules. A histogram was made to determine the distribution of parcel volumes. As can be seen in Figure 8 the distribution of parcel volumes is exponential. This was expected as mostly small parcels were delivered by LEVVs. An analysis is made on how many parcels are present belonging to the various groups of volumes. This is then compared to the total number of parcels allocated to the overflow route to determine if there is a correlation between volumes and overflow scheduling. Table 2.1 shows that there is no strong correlation between the volume and the scheduling to overflow routes.

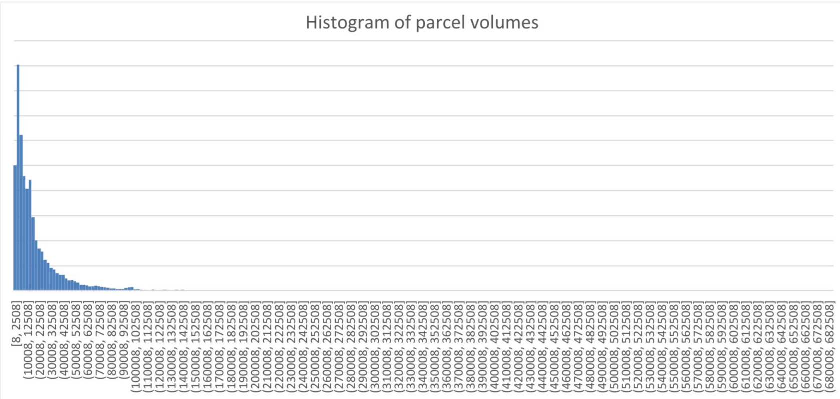


Figure 8: The volumes of parcels are mostly skewed to be smaller than 0.3l (300,000 mm3) being consistent with the company preferencing smaller parcels to be delivered by LEVVs.

Reasons	R^2
Exclusion rule	1
0-20% of max volume	0.3816
20-40% of max volume	0.3266
40-60% of max volume	0.1511
60-80% of max volume	0.0883
80-100% of max volume	0.0654
dummy volumes	0.2308
unknown volumes	0.0285

Table 2.1: The only significant reason (R^2>0.9) for why parcels are removed from their normal route to an overflow route is the exclusion rule. This means that to improve the current allocation tool utilisation rate, the company must change its exclusion rules.

Secondly, the manual changes between the allocation tool solution and the driven route are about equal in how much is planned into the route and how much is taken out. As Table 2.2 shows, both actions are about 11% of all parcels meant to be on the route by both data sets.

Difference scheduled vs reality	Scheduled	Reality
Total correct planned	89%	89%
Total parcels added		11%
Total parcels removed	11%	

Table 2.2: The difference in parcels that are scheduled by the current allocation tool and that are included in the driven routes is about the same in being removed and added to the driven loads. The difference is viewed from the data set of the driven routes

Thirdly, the over-usage of roll containers and trips is analysed. The over-usage is determined by calculating the scheduled capacity and the needed capacity. Then the scheduled capacity is decreased in steps of roll container volumes until the minimum required capacity is reached. Table 2.3 shows that 18 trips and 352 roll containers were scheduled that were not needed if the schedule was more optimal. Note that for roll containers it is possible that only one container was filled but the allocation tool will still schedule three as it wants to fill the van. This is purely to understand the need for hardware better. For this thesis, the trips are relevant and optimisation of available capacity in terms of roll containers is outside of the scope. Table 2.4 shows a more realistic overview of material slack. Here only the scheduled roll containers are taken into account where a trip can be assumed to be loaded by fewer than all three roll containers. Now the not needed material is only 3 containers which is significantly less. The same was done for compartments where two scenarios were used. The first assumes that only the scheduled compartments are analysed which means that a roll container can have fewer than three compartments. The second assumes that all three compartments should be taken into account.

Materials max scheduled vs not needed	Not needed %
Trips	4%
Roll containers	28%
Compartments	34%

Table 2.3: The current allocation tool overschedules the materials at the maximum, i.e. 3 roll containers per trip and 3 compartments per roll container. When looking at the trips only a minimum number is overused. Compared to optimal placement regarding only the volume of parcels and the volume of the compartments.

Materials scheduled vs not needed	Not needed %
Trips	4%
Roll containers	0.3%
Compartments	10%
Compartments full roll container	16%

Table 2.4: The current allocation tool only slightly over schedules the materials with a fair comparison where only the material that contains parcels are accounted for, where over schedules means that parcel volumes could be allocated more efficiently if current packing rules are loosened.

Note that this analysis is unfair to the LIFO approach that Company X uses as it will try to fit parcels anywhere in the trip instead of grouping parcels of a customer. This is more to illustrate the amount of volume that could be used.

2.4 Key Performance Indicators

Company X and CapeGroep have identified several KPI groups in the past that are of importance to the performance of the current allocation tool. They are used to analyse the differences in performances between the 3D heuristics and the current allocation tool. The KPIs are:

- Rate of overflow is used to check how many parcels that could be delivered normally, are not. Having more overflow on a route, means that more time needs to be spend on the parcels and that more costs are incurred for the process
- Utilisation rate is determined by summing the volumes of the parcels allocated to a compartment or trip and dividing it by its available volume. This is important to determine how efficient Company X is working as fewer trips with higher utilisation would mean that they work quicker and more efficiently by having to use fewer vehicles or drive fewer times.
- Number of trips and compartments used is done by counting how many of these of a non-zero utilisation rate per route. Having fewer trips means more efficient delivery.
- The cost of the solution is included to determine if it is beneficial for the company to implement the changes.
- The run time of the program is important for the company to determine if the application is feasible for day to day usage.

Overflow
The first type is about the usage of the overflow route. The algorithm first schedules customers to the overflow if they break the preset company rules. The other time this happens is when customers do not fit within the three trips that are allocated to a route. Therefore three KPIs are the number of customers allocated to the overflow in the presorting, the number allocated to the overflow during the scheduling with the approaches and the number of extra allocations to overflow after applying the approaches to the current tool's solution.

Utilisation rate
The second type is the utilisation rate of the routes, their trips, and the compartments. Three different sets of utilisation rates are used as KPIs: the current tool, the current tool solution with the DBL loading to check validity in terms of 3D, and the DBL loading schedule. These can be used to see how the various loading schedules differ in terms of performance.

Costs
Costs have been chosen to be one of the main solution quality comparisons as this was preferred by Company X. Costs added to the KPIs types identified in Section 2.4. Together with Company X various costs per route have been identified:

- Sorter employee
- Delivery employee
- Overflow costs

The costs of the sorter employee are the amount of time they spend on each route and their hourly salary. Company X assumes that for each route 45 minutes of sorting is required. The costs of the delivery employee are the amount of time they spend on driving and delivering parcels and their hourly rate. For this Company X assumes that no matter how many trips are within a route, they always pay for 8 hours of work.

The overflow costs are determined by the hourly rate of a delivery employee and the amount of time spent driving the overflow route. Company X assumes that a full overflow route takes 1/5th of a working day. To determine how much an overflow route costs, the volume assigned to the overflow route is divided by the maximum volume of a LEVV route. By adding the three cost types per route, the total cost is determined.

Usage of trips and compartments
The third type is the number of trips and compartments used in the current tool and the DBL solutions. As Company X wants to use as few resources as possible it is important to review the solution for this. This one is added as it is relevant for quickly determining if a solution has a lower utilisation rate because of less efficient packing by using more materials or rejecting more parcels from the input data.

Runtime
The last type is due to the is the run-time of the various algorithms. This is important for the company to see if the 3D heuristics are viable to be used daily where long waiting times cannot be accepted.

2.5 Experience from the workfloor

The most important experience is on the loading best practices. Employees use them to make the loading easier and more efficient. The first is to create a place to temporarily place parcels from the belt before packing them into the roll container. This is done in several ways but the most notable is by placing mailbox parcels in the front of a compartment and later fitting them in. Another strategy is placing bigger items in front of the compartment to wait. This allows for more efficient packing later on. A parcel can be allocated to a compartment but does not fit due to deviations in data. Then, the parcel will be placed in another roll container. It will still be delivered by the LEVVs. If the 3rd container is filled then the parcels will be moved to an overflow route. The personnel also packs the roll container by placing soft parcels packed in a bag in the middle and sturdy bigger boxes as an outside ring. This is done because the roll containers are wrapped in plastic, and there could be a chance the parcels can fall out.

Some experiences on the work floor are good for the safety of personnel and parcels but are not taken into account by the allocation tool. The employees indicated to prefer to have the heavy and/or big parcels in the middle compartment row whilst the lightweight parcels can be placed in the top row. This is important as they would exert a lot of strength reaching for those parcels. This can cause injuries in the long term. There is the idea that boxes with wine bottles should be placed in the bottom compartments. If these boxes topple and break, then all parcels below could be damaged, and the costs can be steep. Currently, Company X does not have a penalty assigned when scheduling parcels. Nor do they indicate what amount it should be.

The most important lesson from the workflow is that the employees are already able to efficiently pack the compartments in such way that more parcels might fit in the container than the current allocation tool might suggest. This is because employees and humans have more insight and creativity to orientate parcels in better ways and put them together in more efficient blocks. This is very difficult for machines as it requires finesse. This fact was also taken into account by previous developers of the current allocation tool used by Company X. They mention that further development of the allocation tool was unnecessary than volume-based because humans were packing the compartments. They also mention that humans do not benefit from being told to place a parcel exactly to the millimetre.

2.6 Conclusion

LEVVs are shown to be a sustainable vehicle for city centres. The current way the parcels are loaded into the LEVVs is an approach that can be seen as First-Fit where bins are being closed as soon as the first parcel in the loading list does not fit anymore. Fitting means that the volume of the parcel is smaller than the volume of the free space in the bin. Thus, the parcel volume is assumed to act like a liquid filling a container disregarding the dimensions of parcels and unusable spaces. Besides that, Company X removes certain customers from the LEVV routes who break the exclusion rules.

To answer the research question *How does Company X currently load their LEVVs?*, they make use of their current allocation tool which does two things. It first sorts out parcels based on company exclusion rules. The second aspect is that it assigns parcels to a compartment of a roll container that is transported by the LEVVs. The employees read out that allocation and place the parcel within the compartment as they see fit. This is because no precise placement is given. From the experience, the employees do not have trouble placing the parcels in the assigned compartment.

3 LITERATURE REVIEW

This chapter will outline what current research exists on the Bin Packing Problem (BPP). Section 3.1 gives the problem terminology and classification. Section 3.2 details the constraints that are relevant in the literature and how they can apply to this thesis problem. Section 3.4 lists possible modelling approaches that can be used. Lastly, Section 3.3 gives an overview of methods used in literature together with their problem classification. This helps in choosing the most relevant approach and how to implement it in this thesis.

3.1 Problem classification

The Packing problem or loading problem has been a topic of research for some years now. Dyckhoff (1990) argues that the problem is a subset of the larger Cutting and Packing Problem (CPP) group. The problems are known under various names, e.g., bin packing, knapsack, container loading, cutting stock, and partitioning problems to name a few. The general idea of the CPP is that a large item, a container or a piece of wood, needs to be used by the small items, boxes that must be placed in the container or planks that must be cut from the wood, while using as much as possible of the large item, so to not waste space in the container or waste the wooden piece. The small items need to be fully in the large item and the small items may not overlap. Here the large item can be seen as a container in which small boxes are loaded or as a large sheet of paper from which smaller pieces need to be cut. The large objects are seen as input and the small items as output (Dyckhoff, 1990).

To classify the various subsets, Dyckhoff (1990) developed four characteristics. They are:

1. Dimensionality, the number of dimensions that are needed to define the arrangement of the problem. This can be 1,2,3, or higher.
2. Kind of assignment is between "All large objects and some items" and "Some large objects and all small items". In the first, the number of large objects is set, and the objective is to make as much use of the small items as possible. With the last one, a common objective is using as few large objects as possible whilst including all small items.
3. Assortment of large objects describes the characteristics of the object. Dyckhoff states three main types: (a) only one large object, (b) multiple large objects of the same size and shape, and (c) multiple large objects of different sizes and shapes.
4. And assortment of small items is on the details of the item. Here four main types are detailed. There are differences in the number of items, a lot of items with few different shapes, some items with many different shapes, and congruent shapes.

Wäscher et al. (2007) improved on the typology of Dyckhoff. They deemed the publications on the CPP had increased the knowledge considerably. Therefore, the typology was no longer useful with the new developments. Besides that, Wäscher et al. wanted to introduce a notation system based on terms already in use. The new typology can be found in Figure 9. They agreed with the *Kind of assignment* differentiation of Dyckhoff but renamed his German notations to English ones, output maximisation and input minimisation. The first has the objective to assign as many items to the objects, the latter is using as few objects whilst servicing all items. The next level in Figure 9 splits into the large object(s) dimensions. The last level concerns small items and what kind of shape and how many different shapes there are. This varies from strongly heterogeneous to identical (Wäscher et al., 2007).

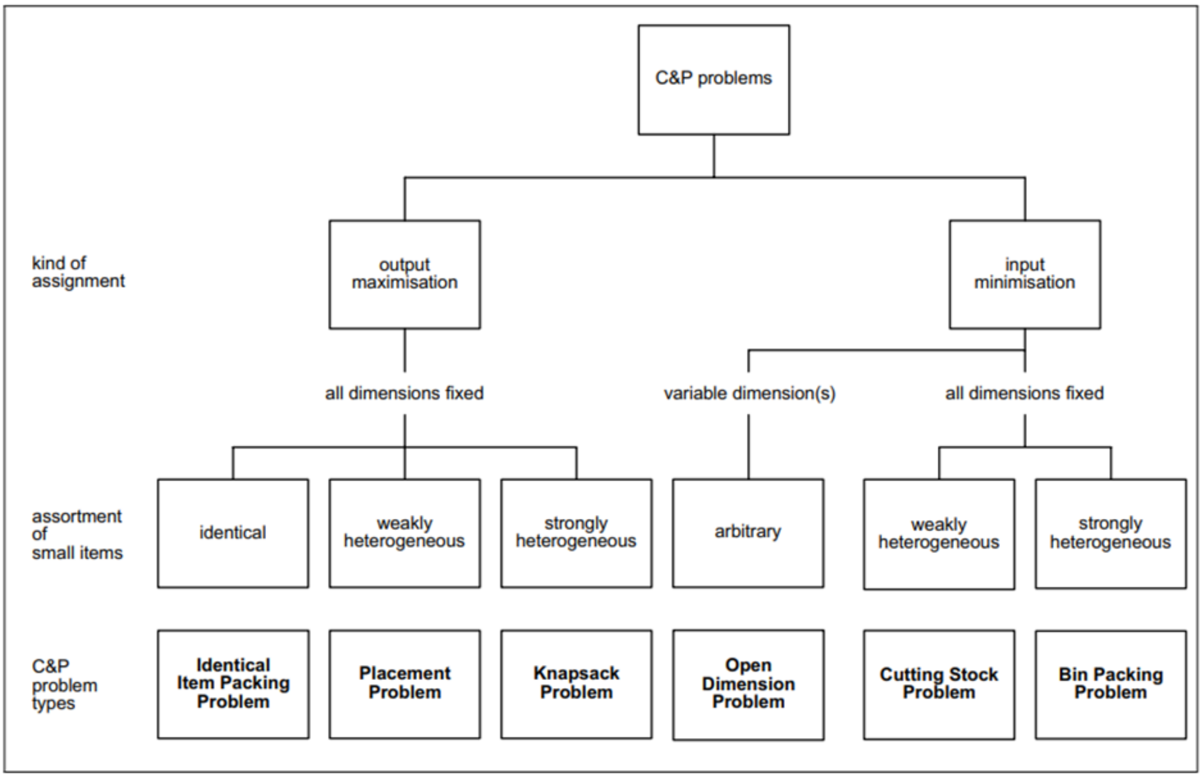


Figure 9: CPP typology according to (Wäscher et al., 2007)

3.2 Constraints

Bortfeldt and Wäscher (2013) performed a review of container loading problems with a focus on which factors were used mentioned by Bischoff and Ratcliff (1995). During their research, they listed the constraints and categorised them into five parts related to an aspect of the problem namely, container-related, item-related, cargo-related, positioning, and load-related constraints. The constraints will be explained using this structure.

Container-related

Weight limit is one of the most frequently used constraints in container loading problems and vehicle routing problems. For the latter, it is often a constraint whether or not to add a stop to a tour by checking whether it is feasible to add the load to the vehicle (Krebs et al., 2021). It is common logic that a container can only be filled with items as long as the weight limit is not exceeded. This kind of constraint is often modelled with a sum of all weights of loaded items that should be smaller or equal to the weight limit (Bortfeldt and Wäscher, 2013)(Chen et al., 1995).

Weight distribution is about spreading the load over the container floor. This improves the safety of the truck and cargo as load shifts are less likely and the axles of the vehicle can carry the weight. Often, this constraint is modelled over the container length as that is frequently the most important direction where weight balance matters with the most common example being cargo planes (Bortfeldt and Wäscher, 2013). Balanced loading is another method to split up weight over the length in two sections left and right. The location of the item determines on which side the weight is allocated. The total weight of a side cannot exceed a certain percentage of the maximum weight (Balakirsky et al., 2010)(Krebs et al., 2021).

Item-related

Loading priorities only arise in loading problems where the value of the load is maximised. In this problem type, there is not enough room for all items, so some are left behind. In reality, some items are better left behind than others due to the service sold, for example. This is often modelled by grouping the prioritised items into a subset that has to be serviced. This constraint can be a hard constraint where all high-priority items have to be loaded first before low-priority can be loaded or a soft one where the objective function can penalise the exclusion of high-priority packages (Bischoff and Ratcliff, 1995)(Krebs et al., 2021)).

Orientation constraints are common in daily practice. An example is the “This side up” sticker often found on fragile packages. An item can have six orientations that can be considered. Bortfeldt and Wäscher (2013) argue that there are five cases of constraints based on the literature:

1. Only one orientation is allowed for each item type, i.e., no rotation is allowed.
2. Only one vertical orientation is allowed, the “This side up” stickers allowing for 90-degree rotations on the horizontal plane.
3. No restriction to the orientation in the vertical direction, but a maximum of two vertical orientations can be forbidden.
4. No restrictions for both vertical and horizontal rotations but a maximum of five orientations can be forbidden.
5. No restrictions for both vertical and horizontal rotations.

Stacking constraints deal with the height of stacking, the load-bearing capacity of packages, and their fragility. They restrict how boxes can be placed on top of each other. The load-bearing capacity is determined by the vertical orientation of the items or the content of the package. Content made out of wood is stronger than that of glass. Another formulation is that boxes have a maximum weight per area that the box can support (Bortfeldt and Wäscher, 2013)(Ceschia and Schaerf, 2013)(Fuellerer et al., 2010)). Fragility is seen as a binary condition; a package is fragile, or it is not. A common formulation of this constraint is that non-fragile boxes cannot be placed on top of fragile ones, but vice-versa is allowed (Bortfeldt and Wäscher, 2013)(Krebs et al., 2021).

Cargo-related

Complete shipment is about requiring that if one item belonging to a subset is loaded then all belonging to that subset must be loaded. If one item does not fit into the container, then none of the items of the subset can be loaded. This constraint is common when the shipment has assembly parts or if a customer cannot be serviced by multiple vehicles on the same day (Bortfeldt and Wäscher, 2013)(Eley, 2003).

Allocation constraints can be split up into two parts: connectivity and separation constraints. Connectivity constraints dictate that items of a subset must be loaded into the same container (Liu et al., 2011). Separation constraints prohibit the loading of items into the same container.

This can be due to various reasons like contamination or food safety standards (Bortfeldt and Wäscher, 2013)(Eley, 2003).

Positioning

Absolute positioning deals with the requirement that packages need to be put in a certain part of the container. A common example is that the size of the item prohibits placement anywhere except near the doors. The item might be too big or heavy and thus must be within reach of a forklift (Bortfeldt and Wäscher, 2013)(Hodgson, 1982).

Relative positioning is similar to the allocation constraint of Section 3.2. The difference is that the Relative positioning is within a container and Section 3.2 between containers. A difference again is placing items close to each other, *grouping constraints,* or placing them away from each other. The former is useful during the loading and unloading of items needed for a customer. The latter can be for the quality of the packages like food and chemicals cannot be placed close to each other as the food will be contaminated.

Multi-drop is a combination of absolute and relative positioning. It recognises the need that subsets of items go to different customers. This constraint takes the arrangement of loading and unloading of the items into account so that no unnecessary handling of other items is needed. A common approach is the LIFO. This would mean that if item i has a customer who visited before the destination of item j, then item j cannot be placed between the doors and item i of or on top of item i (Bischoff and Ratcliff, 1995). LIFO parcels cannot have other parcels on top of them that are delivered later due to unloading with forklifts (Tarantilis et al., 2009). Ceschia and Schaerf (2013) formulated the Manual Last-In-First-Out (MLIFO) where that is allowed.

Not mentioned by Bortfeldt and Wäscher (2013) is the *reachability* of an item. When the package is unloaded then it should be guaranteed that either work equipment or the personnel can reach the package whilst standing as close as possible. An example of the reach is the length of an arm. If an item is out of reach then it should be placed closer to the doors (Krebs et al., 2021).

Load-related

Stability is an important aspect of literature. Unstable loads can cause damage to the shipment or injure personnel when they are handling the cargo. The stability can be split into vertical and horizontal stability.

Vertical stability is to prevent items from falling, i.e., withstand gravitational forces. In reality, this means that the base of an item must be supported by either the floor or the top of another box. This can again be split up into full support or partial support. The former means that the complete box needs to be supported by the underlying item or floor. Partial support means that an overhang of items is allowed. Hemminki et al. (1998) claim that 70% support is sufficient. There is only one problem if multiple items have an overhang, then the centre of gravity of the tower can lay outside of the bottom item and the tower will fall. Robust stability deals with this by introducing multiple overhangs (Ceschia and Schaerf, 2013). The difference is that this method checks the supporting area of all underlying items.

Horizontal stability prevents items not shifting whilst moving the container or their inertia. A simplification of this is placing items next to each other or on the wall. A technique to improve horizontal stability is “interlocking.”Carpenter and Dowsland (1985) indicated three criteria for determining the degree of interlocking:

1. Supportive criterion: The base of a box must touch the top of at least two other boxes.

2. Base contact: At least some percentage of the base of a box must be supported.
3. Non-guillotine: the length of a seam or guillotine cur must not be longer than a certain percentage of the stack's maximum length or width.

Complexity is about how complex a loading pattern is. For manual loading, a complex pattern might be difficult to execute as visualisation might be difficult. For automated loading, a complex pattern might not be suitable for machines as it could lead to additional cost-intensive labour. The "guillotine" pattern is easy as it has many horizontal and/or vertical seams (Carpenter and Dowsland, 1985). This means that there are either towers or layers of items which are simple to visualise for people. However, this pattern is not stable and might require wrapping or fillers. A different pattern is the "robot-packable" pattern. The way of packing is starting in the left corner in the back and successively placing items either in front of, next to, or on top of the previously placed items (Den Boef et al., 2005).

3.3 Literature table

Extensive literature has been written on the BPP. This gives the opportunity to determine what approach is best for the problem of this thesis. This can be done by looking for similarities with the problems of the literature.Table 3.1 is made to compare the problem of this thesis against what is mentioned in the literature. To help this 6 columns are made. The "Objective" column discusses the objective function of the papers. As each problem can have a different goal, it is important to see what is similar and different. The "Solution method" column indicates the methods used to solve the problem. This helps determine which heuristics are commonly used and if exact approaches are used. The "Bin type" column details the bin type as in the BPP it is common to have either only 1 size for all bins or they all differ in size. The next column is the number of bins. For the BPP it is common to have either 1 or many bins depending on the objective of the problem. The LIFO column indicates if the constraint is applied in the paper. The "Rotation" column shows if parcels can be rotated before placement. There are commonly 3 cases, no rotation is allowed, only rotation over the horizontal axis is allowed, 2 rotations, and free rotation as long as the sides of the parcel are parallel to the walls of the bin, 6 rotations.

3.4 Modelling approaches

The modelling approaches that were mentioned in Table 3.1 will be explained in this section.

Off-line vs on-line

Before detailing the various modelling approaches, the difference between off-line and on-line packing problems must be discussed. In the off-line problem, full knowledge of the input items is available. This type of problem uses sequencing of packages or the order in which they arrive for the loading pattern. Besides that, the algorithm manages the empty spaces in the container to determine the possible locations of the next package (Ali et al., 2022). On-line problems are more realistic. Items arrive at the packer and can only be loaded in the order of arrival, with some exceptions where the waiting area is available. The items arrive in a random order (Ali et al., 2022). The only available knowledge is the previously packed items and the ones that must be serviced. This problem type causes packing heuristics to be less efficient due to fewer opportunities of optimising the packing order (Christensen et al., 2017)(Hemminki et al., 1998). This research will from this point on only focus on heuristics and off-line approaches as that is within the scope.

Author	Objective	Solution method	Bin types	Number of bins	LIFO	Rotation
Ceschia and Schaerf (2013)	Min. vehicles used	Random initial, ALNS	1	Many	✓	2
Crainic et al. (2008)	Max. space usage	First and best fit	1	Many	No	0
Eley (2003)	Max. volume utilisation	Greedy block arrangements, treesearch	1	1	No	0
Erbayrak et al. (2021)	Min. used bins	MIP	1	Many	No	6
Gendreau et al. (2006)	Max. space usage	DBL, tabu	M	Many	✓	6
Gzara et al. (2020)	Min. perimeter of overlapping items	Layer building, tree search	1	1	No	0
Ha et al. (2017)	Min. cost of container	DBL	M	Many	No	6
Jin et al. (2003)	Min. vehicles used	Exact	M	Many	No	2
Karabulut and İnceoğlu (2005)	Max. number parcels packed	DBL, GA	1	1	No	0
Koch et al. (2018)	Max. volume used	ALNS, DBL	1	Many	✓	2
Mack and Bortfeldt (2012)	Max. item volume in bin	Layer building	1	1	No	6
Pace et al. (2015)	Min. unused space	Layer building, treesearch	1	Many	No	0
Pan et al. (2009)	Min. wasted space	Wall building, treesearch	1	1	✓	2
Saraiva et al. (2015)	Min. bins used	Layer building, greedy picking	M	Many	No	6
Tarantilis et al. (2009)	Min. vehicles used	Tabu Search, DBL	1	Many	✓	2
Tresca et al. (2022)Wei et al. (2014)	Max. volume used	First fit	1	Many	✓	0
Zhang et al. (2015)	Max. parcels packed	First fit	1	Many	✓	0
Zwep (2023)	Min. unused space	Exact	M	Many	✓	6
THIS THESIS	Max. utility	DBL,EMS	M	Many	✓	6

Table 3.1: Overview of the literature, Homogeneous (1), Heterogeneous (M).

Heuristics

The multi-dimensional loading problem is NP-hard(Bortfeldt and Wäscher, 2013). Therefore, numerous heuristics have been constructed to solve this problem in a reasonable time a near optimally. Several approaches will be explained in this section. For a quick overview and to assess relevance to the problem of this thesis, Table 3.1 is made.

Construction methods
Wall building fills the free spaces with several vertical layers that can be seen as “walls” (Krebs et al., 2021). With this approach, the depth of each layer is normally determined by the first box that is first placed in that layer. It can be used to pack weakly heterogeneous items (George and Robinson, 1980). It works as follows. The algorithm selects a box based on criteria to build the vertical layers (Kocjan and Holmström, 2006). The priority is first to fill the space above and then to the space alongside the layer. If the unpacked boxes do not fit in the remaining space, the space is marked as rejected for some time. When a new layer is made the algorithm combines the new empty space with the previously rejected space (Ali et al., 2022). The walls are created in a sequential manner which allows for dealing with weight distribution (Zhao et al., 2016).

Layer building is a loading approach that places the items into the container in a layer-by-layer procedure. The algorithm first places all items on the floor and then on the next layer until the height of the container is reached (Ali et al., 2022). In this approach, it is useful to try to make a layer out of identically shaped boxes to improve the evenness of a layer (Saraiva et al., 2015). The packing of pallets inspired Bischoff and Ratcliff (1995) to develop this approach. They did not allow more than two different types of boxes to ensure the stability of the pallet. Loh (1992) improved this by grouping weakly heterogeneous boxes by height and then sorting the groups from tallest to shortest. Then the groups would be sorted the same way.

Stack building is a similar approach to wall building but the items are now placed in vertical piles or “stacks” (Gehring et al., 1997). These stacks are then packed in the container where the problem is now reduced to a two-dimensional problem as it is now about placing items on the floor (Ali et al., 2022).

Block building is an approach that tries to combine boxes of one or more types into blocks. These blocks are then packed into the container (Eley, 2003). In the literature, there are two methods of building the blocks. The first one is that blocks consist of one type of item that is oriented in the same way. This approach is useful if the set of items is weakly heterogeneous. The second approach is for building a block out of multiple types of items (Fanslau and Bortfeldt, 2010). Fanslau and Bortfeldt (2010) have formulated a heuristic that makes use of two types of blocks, one with only one type of item and a general block consisting of multiple types. This can again reduce the problem to a two-dimensional problem (Ali et al., 2022).

Deepest-Bottom-Left (DBL) or any other order of these three directions refers to how parcels are positioned in their lowest, deepest, or farthest-back possible spots. This positioning is determined by comparing their x, y, and z values, where ”Left” corresponds to x, ”Bottom” to y, and ”Deep” to z. In the order of DBL, the parcels are placed by first choosing the one with the lowest z value, and if there’s a tie, then the one with the lowest y value is picked, and finally, if there’s still a tie, the one with the lowest x value is chosen (Gendreau et al., 2006).

The locations where parcels can be placed are identified by their x,y, and z coordinates. These coordinates collectively represent the available free space as a cuboid in a container, referred to as Empty Maximal Spaces (EMS) or Extreme Point (EP)s. The ordering of the EMS is done with the DBL priority and is then used in the construction method for quick placement choices. After placing an item the EMS that it is placed in must be updated and other EMSs must be checked if they also must be updated (Crainic et al., 2008).

Parcel orientation is an important aspect that is not regarded in the classical DBL approaches. To optimise over this aspect Mahvash et al. (2018) improved the work of Crainic et al. (2008) and calculated the minimum wasted space of the EMS of each orientation by measuring the distance from each face of the item to the face of the EMS. The minimum value of the following set is taken, where L, W, H indicate the length, width, and height of the EMS and l, w, h the length, width and height of the item:

$$
\begin{aligned}
&((L_ep - l_i), (L_ep - w_i), (L_ep - h_i),\\
&(W_ep - l_i), (W_ep - w_i), (W_ep - h_i),\\
&(H_ep - l_i), (H_ep - w_i), (H_ep - h_i))
\end{aligned}
\tag{3.1}
$$

So first either the L, W, or H axis is removed together with one of the l,w, or h. This first orientation then allows for only two other rotations as only two items in the set remain. By again taking the minimum value the optimal orientation is determined (Mahvash et al., 2018). This approach will be called Residual Spaces in this thesis as no official name was given for this approach.

Tree search is an approach that is mostly focused on building and constructing several solutions by packing items until none fit into the container. This approach is a general one and can be combined with other approaches mentioned above. The overall idea of a tree search is that the algorithm explores the search space and finds a path from the initial state, an empty container, to a terminal state which can be a complete solution. Every node in between is an intermediate state with a list of unpacked boxes, walls, or blocks (Ali et al., 2022).

Local search
Local searches try to optimise the initial solution found by the constructive heuristics. There are various methods used for local searches, but all explore a search space around the initial solution and make small changes iteratively to improve either the quality or the score of the solution (Ali et al., 2022).

Tabu search is an approach that tries to break free from a local optimum by keeping a memory list of past solutions or changes to the solution that cannot be visited again by the algorithm. This avoids cyclic searching and also improves the performance of the algorithm (Kischka et al., 1998).

Genetic algorithms use a fitness function that measures the quality of a solution generation. Then the evolutionary approach combines the most advantageous aspects of the solutions to generate a new parent and use that to create a new generation of solutions. The run time and generally better solution quality make it a preferred approach for three-dimensional packing problems (Ali et al., 2022).

Simulated annealing is an approach where the initial solution is complete but randomly generated. Then a method produces a candidate solution. The acceptance of a new solution is determined by two factors. The first is the question: “Does it improve the current solution?”. The second is where the name gets the part of annealing from. During the process, there is a temperature that decreases with iterations. The temperature symbolises the acceptance chance of a worse neighbourhood solution (Dereli and Sena Das, 2010).

Adaptive Large Neighbourhood Search (ALNS) is an approach that searches the neighbourhood of the solution with different operators so that the size of the neighbourhood can change. The approach works by destroying a solution and repairing it. Due to having multiple destroy operators and repair operators this approach has a higher performance than the standard large neighbourhood search. The operators can be chosen by logic, i.e. after a certain number of iterations with no improvement in the best solution quality, or by randomness (Mara et al., 2022).

Exact

An exact approach provides the optimal solution to a problem. The time required to solve a problem with an exact method increases exponentially with its size. Therefore, it is only used when the item set is small enough or if enough time is available. There are several methods for the exact approach, amongst others, branch-and-bound algorithms and mathematical models also known as analytical models.

For mathematical models, one approach is the Mixed-Integer Linear Programming (MILP) model suggested by Chen et al. (1995). For this approach, the constraints of real-life situations are translated into formulas. In general, the items and container are regarded as parallelepipeds and items must be placed parallel to the wall of the container (Kho, 2018). Common constraints for mathematical models are:

- Parcels cannot overlap.
- Parcels must be placed.
- An item can only be placed once.
- Ensuring that items are placed within the boundaries of the container.

Branch and bound is another exact method to solve an NP-hard problem which is not able to be solved in polynomial time. The bigger problem is divided into subsets of smaller problems that create numerous new problems with every step of solving the bigger problem. The smaller problems are often defined by the aspects of the properties of the subproblem. At every step, the new subproblems are analysed to see if they are discarded based on criteria or explored. The way the steps are explored can be divided into two methods, first, explore every new subproblem of the step called breadth-first, or second explore the most promising subproblem first and down to a possible solution called depth-first (Tomazella and Nagano, 2020).

3.5 Conclusion

The research question *What does the literature propose to solve a 3D loading problem?* will be answered by reading Table 3.1 and comparing it to the aspects of the problem this thesis is handling.

According to Wäscher et al. (2007) the problem of this thesis can be categorised as input minimisation. The goal of the problem is to use as few trips and roll containers, input, as possible. The dimensions of these inputs are fixed but the compartments within the roll containers do vary slightly. The small items, or parcels, are strongly heterogeneous as the dimensions of the parcels vary. Therefore, the problem can be classified as a Bin Packing Problem (BPP) and more precisely a Multiple Bin-Size BPP. The relevant constraints of this problem are: 1. complete shipment, 2. LIFO, and 3. grouping.

According to Table 3.1, the article that is most similar to the problem of this thesis is that of Gendreau et al. (2006). Therefore, the thesis will use the construction and improvement heuristics used in that article, namely the DBL approach. To deal with orientation the Residual Spaces strategy of Mahvash et al. (2018) will also be used. So concluding Mahvash et al. (2018) Gendreau et al. (2006) will be used together as they complement each other.

4 PROBLEM DESCRIPTION AND SOLUTION APPROACH

This chapter details the development of the heuristics chosen in the previous chapter. Section 4.1 details how the problem can be notated in scientific formulae. Section 4.2 explains how the general solution approach will be implemented. In Section 4.3, the constraints will be explained for how they work and are implemented. Lastly, Section 4.4 details how the heuristic will handle optimisation over the orientation of parcels.

4.1 Problem definition

In this problem, a route contains a set of N customers i = 1, 2,..., n. These customers each must be visited in a specific delivery order. Their placement in the order is noted as o_i where a customer i must be visited before customer k being noted as $o_i < o_k$. Each customer has m_i parcels j = 1,2,..., m_i with height, h_j, length, l_j, and width, w_j.

The customers of the route can be serviced by one of the three trips t = 1,2,3 as stated in Section 2.2. Their parcels are being loaded in one of the nine compartments c_t = 1,2,..., 9 in the trip t. The compartments have height, H_c, length, L_c, and width, W_c. Every third compartment in each trip has a different dimension. One could see the trip as a bin. However, for this problem, it is a bin of smaller bins. The traditional bin as used in literature is in this problem is the compartment in which the parcels are placed.

Trips and compartments have three parameters. Each has the parameter volume, which determines how much volume they can hold at a maximum. The next parameter is the volume used. This notes how much volume is used by parcels. Lastly, they have the parameter utilisation rate. This determines how much is filled percentage-wise.

The decision variables are in which trip the customer is placed, i_t, and in which compartment the parcels are placed in the trip, j_tc. The corresponding objective function is to make use of the minimal number of trips and their compartments. An active trip is noted as x_t=*1* and an active compartment as y_tc=*1*. A trip or compartment is active when at least one customer or parcel is allocated to them.

The task is to place the parcels orthogonal, i.e., the faces of the parcel must be parallel to the faces of the bin, into the minimum number of bins considering the LIFO and grouping constraints. Free rotation of the parcels is used but the sides of the parcels must be parallel to the sides of the compartment.

Only one assumption is needed from a mathematical and data point of view. Not all data is perfect. It happens that parcels do not have measurements in their data attributes. Standard dimensions are used for these parcels.

There are two limitations to this notation. The first is that customers cannot be split over two trips, so i_t*<=1* when summed over *t*. The second is that parcels belonging to one customer must be placed close to each other. This is difficult to note but is fixed with how the loops will run so that no other customer can be placed in between them.

4.2 Solution approaches

This section explains the solution approach of the new approach used for 3D scheduling. Figure 10 shows an overview of all algorithms in this chapter. This overview represents the building blocks of the approach and how they function together. Here it shows that the following algorithms below are used inside each other as it zooms in on the various different components. Figure 10 is therefore the main overview of the algorithms as making a pseudo code would not function, because of the zooming in.

As the current allocation tool does not assign parcels based on dimensions, this thesis needs to adapt the tool to consider that, the following two parts detail how this adaptation is done.

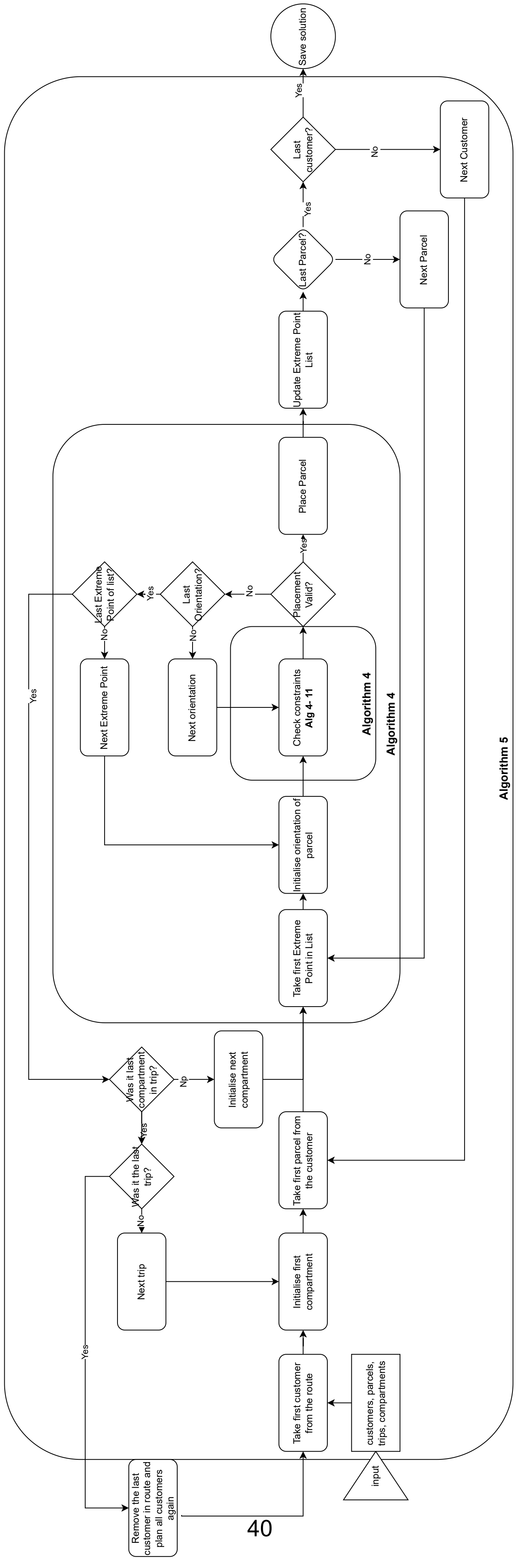


Figure 10: Overview of all included algorithms and basic logic flow.

4.2.1 Deepest Bottom Left

The algorithm will try to place all pre-sorted customers in the three trips in the route. The postal company has requirements for their customers and parcels to fit in the LEVVs. If the algorithm cannot place every customer and their parcels, the last customer in the delivery order is removed and then tried again until all the customers who are not moved to the overflow route are fully placed.

Algorithm 4 is responsible for placing every customer and all their parcels. It also checks that a customer is not split over two trips. The algorithm loops over all customers and within it loop over all their parcels.The placement loop goes in the reverse delivery order to maintain the LIFO constraints and limit possible LIFO violations. For the same reason, the placement within the trips is also in reverse order starting with the last trip in the last compartment. The algorithm checks if there is enough free volume in the trip left for the customer. If this is not the case the next trip is then used for further placement. If it does fit, the algorithm will try to place all their parcels.

Algorithm 4: DBL placement of Customers

Data: $ListOfCustomers, ListOfCompartments, ListOfTrips$
Result: Allocation and packing schedule on DBL

c := $Compartment$ in $ListOfCompartments$
t := $Trip$ in $ListOfCompartments$
i:= $Customer$ in $ListOfCustomers$
n := Last customer in list
j:= $Parcel$ in $ListOfParcels$
m_i := Max parcels of customer i
for i = *1 to* n **do**
    while $ParcelsPlaced = False$ **do**
        for j = *1 to* m_i **do**
            while $Placed = False$ **do**
                $ParcelPlaced$ = `PlaceParcel(`*c, j, Extreme Point List*`)`
                The procedure returns if the Parcel is placed
                if $Parcelisplaced$ **then**
                    Set $Placed$ = True
                    Update $ExtremePointList$
                    Break the while loop
                else
                    Next c, or t, and try again
            if *The trip was changed within customer* **then**
                Remove all their parcels from the previous trip
                Try again all parcels in the new trip
        if *Customer succesfully placed* **then**
            $CustomerListPlaced, ParcelsPlaced$ = True
    if $CustomerListPlaced = False$ **then**
        Not all customers are placed within the three trips
        remove the last customer and try again

Algorithm 5 is responsible for placing each parcel. The placement is done on EPs in the compartment. Each possible EP is checked for each parcel orientation for valid placement according to the constraints. The first valid combination is accepted as the placement of the parcel. If no combination of EP and orientation is valid, the next compartment is used from now on. The new compartment only has one initial EP namely the DBL corner or (0,0,0). If not all parcels of a customer fit in one trip, all placed parcels are removed and tried again in the new trip.

Algorithm 5: DBL placement of parcels

Data: $Compartment, Parcel, ExtremePointList$
Result: Placement of Parcel on DBL

```
Function PlaceParcel(Compartment, Parcel, ExtremePointList)
    c := Compartment
    j:= Parcel
    To determine where the Parcel is to be placed check for all extreme points as possible
      locations
    forall Extreme Points in ExtremePointList do
        For every possible location, check for all possible orientations of the parcel and
          choose the first valid combination of place and orientation
        forall Orientations of j do
            Perform valid placement check
            Valid = ConstraintValidation(c,j,Extreme Point, Orientation)
            if Valid then
                Update the j with Placement and Orientation
                Update the c with the Parcel as placed
                return Parcel as placed



    if No Placement is valid then
        return Parcel as failed
```

4.3 Constraints

Four constraints are relevant to the Company X case: 1. Geometry, 2. Collision, 3. LIFO, and 4. Support. All constraints must be fulfilled before a placement of a parcel can be considered valid. Their relevance will be explained in their respective sections. The general constraint validation is detailed in Algorithm 6. It shows how the several constraints work together. 4 constraints are checked to see if they deny the placement. If one of them determines if the location is invalid, the location is denied.

Algorithm 6: Check if the placement is valid

Data: $Compartment, Parcel, Placement, Orientation$
Result: Boolean indicating if constraints are not broken

c := $Compartment$
j := $Parcel$
$Orientation$ is an aspect of j
$ListItemsPlaced$ is an attribute of c
Function `ConstraintValidation(`$c, j, Placement, Orientation$`)`
    if *Any of the following constraints returns True* **then**
        Check all constraints with the next 4 procedures to see if the placement is valid
        `GeometryConstraint(`$c, Placement, Orientation$`)`
        `SupportValidation(`$ListItemsPlaced, Placement, Orientation$`)`
        `LIFO(`$j, ParcelPlaced$`)`
        `MainCollisionConstraint(`$ListItemsPlaced, j, Placement$`)`
        If any constraint is $True$ then return $False$ for not valid placement
        return $False$
    else
        No problems with constraints so placement is valid
        return $True$

4.3.1 Geometry

The geometry constraint makes sure that all parcels are fully placed within the compartment. This is needed to ensure parcels are not partially loaded and would otherwise be stuck out of the container. The pseudo-code in Algorithm 7 details how the algorithm checks if the parcel breaks one of the walls of the bin. It does it by adding the dimension to the position and see if that value is greater than the wall coordinates.

Algorithm 7: Geometry constraint to place parcels fully in the compartment

Data: $Compartment, Placement, Orientation$
Result: Boolean if parcel breaks the geometry constraint

Function `GeometryConstraint(`$Compartment, Placement, Orientation$`)`
    c:= $Compartment$
    If any of the following placement and orientation is breaking the compartment wall then return True to indicate breaking
    if H_c < $Placement$ + $Orientation[y]$ **then**
        return $True$
    else if W_c < $Placement$ + $Orientation[x]$ **then**
        return $True$
    else if L_c < $Placement$ + $Orientation[z]$ **then**
        return $True$
    else
        No compartment wall is broken
        return $False$

4.3.2 Collision

The collision constraint checks if the parcel being placed will not collide with a parcel that is already placed. This is done by mapping the XY, XZ, and YZ planes of the parcel that is going to be placed and one by one, the already placed parcels. For both planes, the parcels are projected onto their axes. There the midpoint of the parcel sides are determined. If for both axes the distances between the midpoints are smaller than half the dimensions of both parcels in that axis, then that plane shows collision. If there is a collision in all three planes the parcels collide. Algorithm 8 checks if the parcel collides with another already placed parcel. It determines collision with Algorithm 9 which contains the mathematical formulas that check if the parcels collide in all three dimensions.

Algorithm 8: Collision constraint

Data: $ListItemsPlaced, Parcel, Placement$

Result: Boolean if Parcel collides entirely with another parcel

Function `MainCollisionConstraint`($ListItemsPlaced, Parcel, Placement$)
    j:= $Parcel$
    forall *ParcelPlaced in ListItemsPlaced* **do**
        Check for all parcels that are placed if they collide with the ParcelToBePlaced and its current placement
        Collision = `CollisionValidation`($j, ParcelPlaced$)
        if *Collision* **then**
            j collides with a $ParcelPlaced$ so no further checks need to happen
            return $True$
        else
            Check the next $ParcelPlaced$ for collision
    No collision with any $ParcelPlaced$
    return $False$

Algorithm 9: Intersect in planes between parcels

Data: $Parcel, ParcelPlaced$

Result: Boolean if parcel collides with another parcel

Function `CollisionValidation`$(Parcel, ParcelPlaced)$

PP = ParcelPlaced

j := Parcel

$CentreX_{\text{PP}}$ =$PPPlacement[X]$+$PPDimension[X]$/2

$CentreX_{\text{j}}$ = $j.Placement[X]$+$j.Dimension[X]$/2

$CentreY_{\text{PP}}$ =$PPPlacement[Y]$+$PPDimension[Y]$/2

$CentreY_{\text{j}}$ =$j.Placement[Y]$+$j.Dimension[Y]$/2

$CentreZ_{\text{PP}}$ =$PPPlacement[Z]$+$PPDimension[Z]$/2

$CentreZ_{\text{j}}$ =$j.Placement[Z]$+$j.Dimension[Z]$/2

$MidPointX$ = Max($CentreX_{\text{PP}}$,$CentreX_{\text{j}}$)-Min($CentreX_{\text{PP}}$,$CentreX_{\text{j}}$)

$MidPointY$ = Max($CentreY_{\text{PP}}$,$CentreY_{\text{j}}$)-Min($CentreY_{\text{PP}}$,$CentreY_{\text{j}}$)

$MidPointZ$ = Max($CentreZ_{\text{PP}}$,$CentreZ_{\text{j}}$)-Min($CentreZ_{\text{PP}}$,$CentreZ_{\text{j}}$)

if *the value for midpoints are smaller than half the dimensions of both parcels* **then**

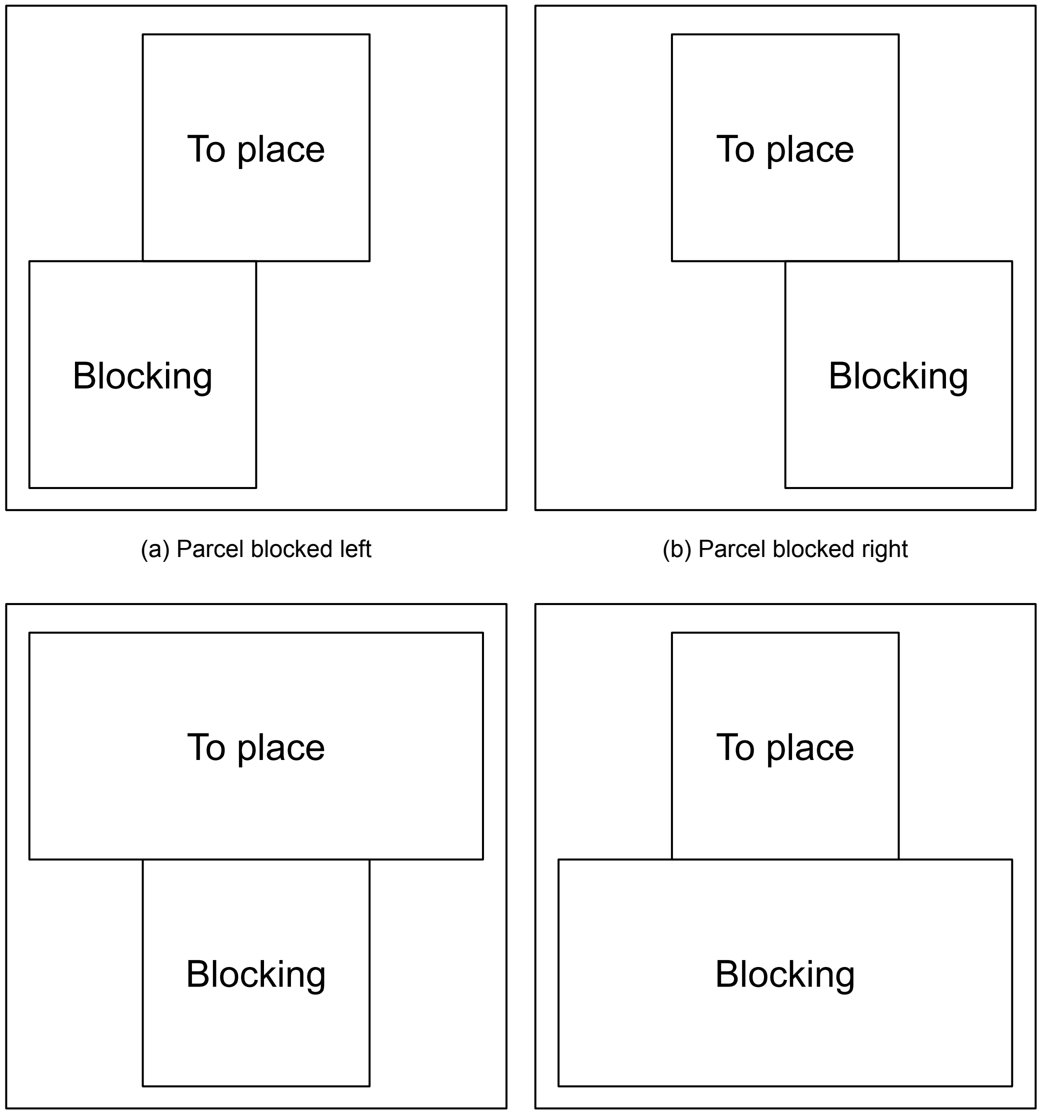
  return $True$

else

  return $False$

4.3.3 Last in first out

The LIFO constraint checks if an already placed parcel blocks a parcel of a customer that is visited earlier. Blocking is done by either being in front or on top of another parcel. Blocking can only happen between two different customers, and only customers who are visited later in the route can block customers who are visited earlier. The logic is shown in algorithm 10 This constraint is executed twice, for blocking on top and blocking in front. The pseudo code details both directions but for simplicity, only the front check will be written here. Firstly the X-axis is checked as this is used for both directions and can be re-used. It checks if parcels already placed share a part of the axis of the to-be-placed parcel. There are four cases of which only one needs to be fulfilled to be blocking. Figure 11 shows the four cases of how a parcel can block another parcel from exiting. These examples also show the commonalities between the cases to make conclusions on how to represent the case in mathematical terms.



(a) Parcel blocked left (b) Parcel blocked right

(c) Parcel blocked by smaller parcel (d) Parcel blocked by bigger parcel

Figure 11: There are 4 cases for blocking parcels in a direction. All parcels must exit at the bottom of the figure

The four cases share the common aspect that blocking happens when:

- The to-be-placed parcel's left point is to the left of the right point of the already placed parcel.
- The to-be-placed parcel's right point is to the right of the left point of the already placed parcel.

In logic this would be:

$$\begin{aligned} ToPlaceParcel_{left} < PlacedParcel_{right} \\ ToPlaceParcel_{right} > PlacedParcel_{left} \end{aligned} \tag{4.1}$$

The parcels that fulfil this filter, are then filtered to check if their back edge has a higher or equal Z coordinate than the front edge of the to-place parcel. This is to make sure the parcels can block and be in front of the to-be-placed parcel. If there are parcels left in this filtered list, then the location is blocked and invalid.

Algorithm 10: LIFO constraint

Data: $ListItemsPlaced, Placement, Orientation$
Result: Boolean if parcel is breaking LIFO constraint

Function `LIFO`$(ListItemsPlaced, Placement, Orientation)$
  Parcels must share a part of the X-Axis regardless of removal direction
  $ItemsBlockingXaxis$ = `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, X - axis)$
  Of the parcels already placed sharing the X-Axis, check if they block in either removal direction
  Check for the Up removal direction
  $LIFOBlocked$ = `LIFOExit`$(Placement, Orientation, ListOfFilteredParcels, Y - axis)$
  if *LIFOBlocked* **then**
    return $True$
  Check for the front removal direction LIFOBlocked = `LIFOExit`$(Placement, Orientation, ListOfFilteredParcels, Z - axis)$
  if *LIFOBlocked* **then**
    return $True$
  return $False$

Algorithm 11 shows the logic for generating the list of parcels that could block the parcel that is going to be placed. It is modified to perform different checks for removal from either the front or top. This last one is to ensure parcels are not left floating after a parcel is removed from the bin. In the end the algorithm checks if there are any parcels left in the list. If no parcels are present, then it is a valid location.

Algorithm 11: Logic to check if parcels block in direction

Data: $Placement, Orientation, ListOfFilteredParcels, axis$
Result: Boolean if the parcel is blocked

Function `LIFOExit`$(Placement, Orientation, ListOfFilteredParcels, axis)$
  Determine the first and second directions based on the removal axis
  if *axis = "front"* **then**
    $LIFOdirection$ = z and $FinalBlocking - axis$ = y
  else
    $LIFOdirection$ = y and $FinalBlocking - axis$ = z
  Filter based on parcels sharing the LIFO axis
  $FrontTopParcels$ = `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, LIFOdirection)$
  Filter based on parcels sharing the removal axis to see if any parcels block the ParcelToBePlaced
  $ParcelsBlocking$ = `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, FinalBlocking - axis)$
  if *ParcelsBlocking list is not empty* **then**
    return $True$
  else
    return $False$

Algorithm 12 shows the logic how a list of parcels is checked if all parcels are blocking. It does so with the help of Formulas 4.1 and applies that formula to all parcels that are in the list of parcels that needs to be checked.

Algorithm 12: Logic to check if parcels overlap in any direction with already placed parcels

Data: $Placement, Orientation, ListOfFilteredParcels, BlockingDirection$
Result: $FilteredlistofblockingParcels$
Function `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, BlockingDirection)$
    ParcelLeft = Placement[BlockingDirection]
    ParcelRight = Placement[BlockingDirection] + Orientation[BlockingDirection]
    forall $ListOfFilteredParcels$ **do**
        Keep Parcels that fulfil the Filter with Formulas 4.1
    return $filteredlist$

4.3.4 Support

Support is defined as how much area of the bottom of the parcel is supported by the parcels below. Algorithm 13 is used to calculate if there is enough support below the parcels. It does so by adding the supporting areas of the parcels below together. The supporting parcels are given by Algorithm 12. It is used to determine if parcels are directly or partially below the parcel placement option. All the parcels that are below, are filtered for sharing their top y coordinate with the y coordinate of the bottom of the to-be-paced parcel. These parcels can provide support. Then the shared area of these parcels is calculated per parcel and added together. If there is enough area as a percentage of the bottom of the parcel, the parcel is supported.

Algorithm 13: Support constraint

Data: $ListItemsPlaced, Placement, Orientation$
Result: Boolean if the parcel is breaking the support constraint
if *Placement Y-level = 0* **then**
    return $False$
First, determine which ParcelsPlaced share a part of the X-Axis
$ParcelsShareXAxis$ = `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, BlockingDirection)$
Then determine those parcels left if they share a part of the Y-Axis
$ParcelsShareYAxis$ = `LIFOOverlap`$(Placement, Orientation, ListOfFilteredParcels, BlockingDirection)$
The remaining parcels are either partially or wholly located beneath the ParcelToBePlaced
Filter ParcelsShareYAxis for parcels that have a top coordinate equal to bottom coordinate placement to be able to support
forall *ParcelsShareYAxis* **do**
    Calculate the supporting area and sum it together
if *There is enough supporting area* **then**
    return $False$
else
    return $True$

4.4 Orientation of parcels

The DBL algorithm has a high preference for the initial orientation of a parcel. This can have an impact on the solution quality. Here, orientation is used to determine the dimensions of a parcel. The orientation is a tuple of 3 values of the dimensions which when reordered makes it a new orientation or rotation. The approach of Mahvash et al. (2018) and Crainic et al. (2008) is chosen as they together account for free orientation and LIFO, which makes it applicable to this thesis based on Table 3.1.

4.4.1 Residual spaces

Mahvash et al. (2018) made use of DBL for the placement and allowed for free rotation. They changed the EP to Residual Spaces to be able to calculate volumes around parcels to determine the best orientation of parcels. The space is defined by a width, length, and height from the EP to the front, right and top walls of the bin or parcel if that is closer. The orientation of Parcel_i is determined by taking the minimum value in the set as determined in Equation 3.1.

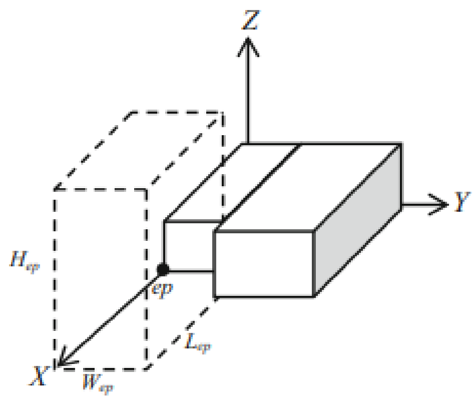


Figure 12: Example of Residual Space Mahvash et al. (2018)

This is done for all EPs and orientations. The first orientation is now chosen for the EP. One axis and orientation of the set is removed and used for filtering the EPs. This leaves four options for values in the set per EP. The minimum value of the set is chosen over all EPs locking the orientation of the parcel. Again, the set is filtered for the axis and orientation and the minimum value is chosen for the choice of EP. The rest of the algorithm is similar to the DBL where only the placement and orientation are different.

4.4.2 Required data

For this algorithm to work, data is needed on the dimensions of the parcels for dimension-based placement. For the LIFO constraint the delivery order of the customers must be known. Besides that, information is needed on the delivery method, mainly the number and sizes of the bins or containers in which the parcels are placed. This is mainly for the geometry constraint.

4.5 Conclusion

This chapter aims to answer the question: *How should the 3D packing approach be designed?*

There are functional requirements of the model important to adhere to the company's logic. They are:

- A customer is never split over two trips.
- Customers must be visited in a certain order and in such a way they must also be loaded into the compartments.
- Parcels belonging to one customer must be placed as close as possible to each other.

The data that is needed for the model is divided into customers and their parcels. For the customer only their ID is needed, how many parcels they have, and when on the route they must be serviced, also known as a customer delivery order number with 1 being the first customer to be visited. For the parcel, their dimension is needed and their ID to be able to identify them.

The modification of the current allocation tool to a 3D heuristic is done by using the DBL approach chosen based on the conclusion of Chapter 3. This answers the main question of this chapter regarding the design. This approach makes use of EP to create possible placement locations for parcels. This approach however preferences the initial orientation of a parcel. To account for this another approach was implemented based on (Crainic et al., 2008) and (Mahvash et al., 2018). It uses Residual Spaces to notate a maximum cube from the EP to either the walls of the compartment or another parcel, whichever is closer. This helps to choose the optimal orientation of a parcel for a EP. New positions can be evaluated to waste as little space from the parcel to the cube and ensure that more parcels should be able to fit between the walls of the Residual Space and the parcel. Both approaches have 4 constraints that must be checked to make sure placement is valid and not breaking company rules.

5 EXPERIMENTAL SETUP

This chapter details how the experimentation and data runs are done. Section 5.1 explains how the data is collected. In Section 5.2 the data is formatted and cleaned for experimental usage. Lastly, Section 5.3 details how the experiments are designed.

5.1 Data collection

Data was collected over 4 weeks, from the 4th of September until the 30th of September. This data was collected from the application where the current allocation tool runs and that is also used as the current input. This is to make sure the data and the experiments reflect reality. This data contains information on what routes are driven, the date of the drive, to which depot the route belongs, the customer with their ID, their stop sequence number, the parcels with their ID, their dimensions, weight, and an indicator if the parcel can be handled by machines. The total data collected resulted in 289 data points that can be divided into scenarios. All the data points will be used as they are all valid representations of the performance of the current tool.

5.2 Data preparation

Two steps are taken for the data preparation. Firstly, the data was extracted from the JSON files to become a structured object with several attributes. Every route, customer, and parcel is transformed into an object. The structure is as follows: a route is an object with attributes route ID, depot, and date. The route was then also given two attributes that are lists, customers and overflow lists. The customer list is exactly what is mentioned, a list of customer objects. The overflow list is a list of parcels that will not be delivered by the normal route. This list is extended in the initial filtering explained later and during the three algorithms mentioned in 4. The customer has the attributes ID, StopsequenceID, and the total parcel volume. The last attribute is the list of parcel objects. The parcel has the attributes ID, customer ID and Stopsequence ID for better referencing, dimensions, weight, and the boolean if the parcel can be handled by machines.

This structure of objects makes sure that referencing is done correctly and objects don't get lost from their parent route.

The customer ID is their zip code. As this is personal information, this will not be revealed or used in the output. The customer ID is only relevant as an internal identifier as stop sequence ID has a high likelihood of duplicates throughout the different routes. This identifier is used anywhere else except identification of objects

The second step is filtering the routes based on the company rules. During this filtering, a list of "valid" customers is made whose parcels all fulfil the requirements. The customers who have at least one parcel that violates the requirements are removed from the valid list and moved to the overflow. Only their parcels are saved there as no further operations need to be done on the parcels and their customers except counting the number of parcels in the overflow list.

Now, the data is clean and can be used as input for the scenarios. Table 5.1 shows that seven scenarios are made to test the approaches. Each scenario contains a certain number of data points. Each data point represents a set of customers and their parcels. Each data point instance represents thus a day of a route that Company X has to drive. So as an example, scenario 1 contains 2 data points. This means that there are 2 routes with a certain number of customers but each route has in total less than 50 parcels. These routes can be from different days and from different places.

Scenario	Parcels	Individual data points
1	0-50	2
2	50-100	44
3	100-150	81
4	150-200	136
5	200-250	21
6	250-300	5
7	all	289

Table 5.1: Overview of the scenarios. Each scenario contains a certain number of data points that are used to gain an average insight into the scenarios.

5.3 Experimental design

All the algorithms are run on a computer with an i7-13700H processor of 2.40 GHz and 32 GB RAM. The algorithms are applied to each route before the next route will be considered. After all routes are put through the algorithms, the solutions of the current tool will also be put through the 3D placement heuristics. After this, the experiments are done and the solutions are evaluated to determine the KPIs per route and then exported to an Excel file.

5.3.1 Fair comparison between current tool and heuristics

Currently, comparing the outcomes of the existing allocation tools and the 3D placement heuristics might seem unfair. The current tool employs a form of Bin Packing Problem (BPP) relaxation, achieved by omitting dimensions from the classical problem formulation. This relaxation simplifies the problem, making it more manageable to solve. Comparing the current tool's results with the heuristics involves assessing a relaxed solution against a non-relaxed one. To address this, the 3D placement heuristics are applied to the packing solution generated by the current allocation tool.

This process involves taking all parcels allocated by the current tool to a compartment and attempting to place them based on their dimensions. Parcels that cannot fit in their assigned compartment are relocated to an overflow area. It is assumed that there are no additional compartments or roll containers available for these extra parcels, and updating information in Company X's data systems during the placement phase is deemed challenging. This complexity is also influenced by connectivity and the constraints of one-stop per-customer.

As a result, the comparison now occurs between solutions that are all 3D-based, ensuring a more equitable evaluation.

5.3.2 Experiments

Four experiments are made for this thesis to evaluate the performance of the approaches. They are:

Feasibility

The purpose of the experiment is to determine whether the solutions generated by the current tool are feasible for placement. The placement will be tested with the 3D attributes and the LIFO constraint as these represent reality. A solution will be regarded as infeasible when not all parcels can be placed in their allocated compartment.

Size

The problem size differs throughout the data set. Not every data point has the same number of parcels. The purpose of this experiment is to determine if there are trends for the KPIs within the allocation approaches. The experiment will test for utilisation rates and run times. These are the most important ones according to Company X.

5.3.3 Fair comparison

As mentioned above, comparing the current allocation tool with the 3D approaches is unfair. Therefore, the experiment is made with the purpose of comparing only 3D applied solutions. The experiment will test for the 1. utilisation rate, 2. overflow, and 3. costs KPIs.

Grouping

Lastly, an extra experiment is done without scenarios. This is done by looking at the entire data set per allocation approach. This experiment will evaluate all KPIs and make a whiskers plot with them to have a broader view of the performances. The purpose is to have an overall trend between the various allocation approaches to determine which has a better performance.

5.4 Conclusion

This chapter answers the questions 1. *How can the allocation solutions from the allocation tool be validated*, 2. *Which design should be used for the experiments*, and 3. *How can the allocation tool be compared to the 3D heuristics in a fair way?*

The solution validation is done by comparing the output of the current tool versus the output of the model made for this thesis. The outputs were of the structure where parcel IDs were linked with a compartment code. The validation was performed over 3 routes driven.

The experimentation design was made together with Company X. Three experiments are made. They focus on feasibility, problem size, and the fair comparison between the current tool which is not 3D and the 3D heuristics.

The fair comparison is done by applying the 3D placement heuristics to the solution of the current tool. This transforms the solution from a relaxed version of the BPP to a strict one. Therefore, it is now able to be fairly used in comparison to solution qualities.

6 EXPERIMENT RESULTS

In this chapter, the results of the experiments are shown. In Section 6.1 the results of the experiments that were defined in the previous chapter are shown. Section 6.2 shows a more general overview of the results of the scenarios when they are grouped together.

Please note that all cost-related data depicted in the charts in this chapter have been anonymised to uphold financial confidentiality. Techniques have been used to bring it all in a range where the standard performance was set at 1 and values have been scaled back to this initial value to indicate decreases and increases in costs. These adjustments ensure the protection of sensitive information while maintaining the integrity and the comparative value of the data. The figures should not be used for exact values but as representative values that illustrate the underlying patterns, trends, and distributions.

6.1 Experimental results

This section details the results from the experiments that were defined in the previous chapter. There are four experiments 1. best performance for utilisation, 2. feasibility, 3. size impact, and 4. 3D comparison.

6.1.1 Best performing approach for utilisation

This experiment is made to determine which approach reaches the best performance in each scenario. Each scenario has a number of data points. This number is the maximum number of best performing cases per approach. Each performance of a data point of all approaches per scenario is compared against the best performing data point of all approaches. As an example, the DBL has the best performance of data point 5. Then all results of all approaches with data point 5 are compared against that best result. If the performance matches, the approach receives a +1 for that scenario. It is therefore possible that all approaches have the same number of optimal cases in a scenario. Table 5.1 shows the number of data points per scenario and therefore the maximum number of data points that reach the best performance. For each data point, the maximum utilisation is taken and compared against the performance of the approaches. It is therefore possible that multiple approaches reach the best performance per data point. Three smaller experiments are made to compare the best performance through different lenses.

The first experiment compares all approaches including the current allocation tool. Table 6.1 shows the number of data points that reached the best utilisation rate. To clarify, in scenario 2, the current approach has 44 data point that reached the maximum utilisation rate of the data points, the applied cases 0, the DBL approach has 7 data points that reached the maximum utilisation rate of their instance, and Residual had 16 data points that had the maximum utilisation rate. This comparison can be seen as unequal as the current allocation tool is a relaxed version of the BPP. However, when looking at Table 6.1 the current tool reaches the best utilisation rate the most often. The table also shows that other approaches only reach the best utilisation rate with smaller problem sizes.

Scenario	Current	CurrentDBL	CurrentResidual	DBL	Residual
1	100%	50%	50%	100%	100%
2	100%	0%	0%	16%	36%
3	91%	0%	0%	0%	4%
4	100%	0%	0%	0%	2%
5	100%	0%	0%	0%	0%
6	100%	0%	0%	0%	0%

Table 6.1: The current tool reaches the best utilisation rate the most often. The other approaches do not reach this best utilisation often and only for smaller problem sizes.

The second experiment only compares the 3D approaches as the current approach always reaches a higher utilisation rate due to the aforementioned reasons. This means that a 3D heuristic has been applied to the current allocation tools solution. Now a better understanding can be reached which approach has the best utilisation rates. Table 6.2 shows a pattern that the Residual Spaces approach is performing better in terms of utilisation rate. The DBL approach only reaches the best utilisation rate for smaller problem sizes.

Scenario	CurrentDBL	CurrentResidual	DBL	Residual
1	50%	50%	100%	100%
2	7%	52%	16%	36%
3	5%	48%	0%	42%
4	0%	14%	0%	88%
5	0%	29%	0%	81%
6	0%	20%	0%	80%

Table 6.2: The Residual Spaces approach reaches the best utilisation rate the most often. The DBL approaches do not reach this best utilisation often and only for smaller problem sizes.

The third experiment only compares the performance of the 3D heuristics, DBL and Residual. Here, the same result as in the second experiment is shown. The Residual Spaces approach is reaching the best utilisation rate more often than the DBL approach. This proofs that the optimisation over orientation that the Residual approach does is beneficial.

Scenario	DBL	Residual
1	100%	100%
2	16%	95%
3	0%	100%
4	0%	100%
5	0%	100%
6	0%	100%

Table 6.3: The Residual Spaces approach reaches the best utilisation rate the most often. The DBL approach does not reach this best utilisation often and only for smaller problem sizes.

6.1.2 Feasibility

This experiment is made to determine if the relaxed version creates a valid loading pattern. This is tested with the 3D heuristics if the solution is feasible for the placement of all the parcels that are allocated to a specific compartment. Feasibility is tested for 3D placement and adherence to LIFO constraints. If not all parcels can be placed in their allocated compartment the solution is infeasible. Table 6.4 shows that no loading pattern is feasible went tested for placement. All data points could not provide a solution where all parcels could be placed in their assigned compartments by the current tool. This means that the 3D heuristics, which do always result in a valid loading pattern, are more reliable when using machines.

Scenario	Current with DBL	Current with Residual Space
	Percentage not feasible	
1	100%	100%
2	100%	100%
3	100%	100%
4	100%	100%
5	100%	100%
6	100%	100%

Table 6.4: A testing is done for the loading pattern current solution tool for feasibility with 3D principles. No loading patterns would be feasible when testing for 3D placement with the LIFO constraint.

6.1.3 Impact of sizes

This experiment is done to determine what impact the problem size has on the utilisation rates of the solutions of the various approaches. Table 6.5 shows an overview of the performances of the current tool, the DBL approach, and the Residual approach. The performances are measured with the KPIs utilisation rate and run time. The DBL and Residual approaches also have a column for the differences in utilisation rate compared to the current approach. This table provides insight into how the performance between the approaches changes with a different number of parcels per data point. Table 6.5 shows that there is a positive trend amongst the three approaches until the fifth scenario. This can be explained that more parcels were needed to fill the compartments for this utilisation rate. The last scenario indicates that this amount of parcels would require more trips or roll containers, which would lower the utilisation rate again. A comparison is made between the current tool and the heuristics. When comparing between the approaches no clear trend in the difference in utilisation rate is shown. Note that the previous experiment states that all solutions of the current tool are infeasible when checked for dimensions. This means the difference in utilisation rate should be read as the improvement employees make as they pack the containers. Reality shows that the current solutions are workable. Table 6.5 shows that the employees improve significantly the utilisation rate with a maximum of around 30%.

Scenario		DBL			Residual spaces			Current	
Number	Num parcels	Utilisation rate	difference current	Runtime(sec)	Utilisation rate	difference current	Runtime(sec)	Utilisation rate	Runtime(sec)
1	0-50	12%	0%	.01	12%	0%	.02	12%	.01
2	50-100	28%	-22%	.05	33%	-17%	.09	50%	.03
3	100-150	32%	-33%	.09	38%	-28%	.14	65%	.07
4	150-200	37%	-22%	.44	40%	-19%	.23	59%	.09
5	200-250	40%	-32%	2.18	46%	-26%	1.75	72%	.10
6	250-300	39%	-30%	8.05	49%	-20%	12.49	69%	.14

Table 6.5: Increasing the problem size shows a positive trend in utilisation rate for all three approaches. That is until a tipping point is reached where more trips or roll containers are needed thus decreasing the utilisation rate.

6.1.4 3D applied comparison

This experiment is made to compare the performance between the relaxed solution of the current allocation tool and the 3D heuristics.

Table 6.6 is about the application of the 3D heuristics to the solution of the current tool. In all scenarios, the utilisation rates decrease. The Residual Spaces approach decreases less. For both applications the overflow rate increases, with the sixth scenario having more than 40% overflow rate. This also shows in the increase in costs. This can conclude that applying the heuristics decreases the solution performance for all three mentioned KPIs. It also means from Table 6.4 that the solutions are feasible. Note that the current tool does not show any overflow. This means that the allocation tool itself is not the problem that the company experiences with overflow parcels but rather the exclusions rules are the problem as mentioned in Section 2.3.

Scenario	Tetris			Tetris with DBL				Tetris wit	
	Utilisation		Costs	Utilisation	Utilisation change	Overflow	Costs	Utilisation	Utilisation ch
1	12%	0	0.99	7%	-4%	0.11	1.00	10%	-2%
2	50%	0	0.95	29%	-22%	0.20	0.99	31%	-19%
3	65%	0	1.07	36%	-29%	0.25	1.14	41%	-25%
4	59%	0	1.08	33%	-26%	0.26	1.17	37%	-22%
5	72%	0	1.22	41%	-31%	0.26	1.33	46%	-26%
6	69%	0	1.20	41%	-28%	0.41	1.33	45%	-24%

Table 6.6: The solution performance of the current tool is tested when 3D heuristics are applied. The utilisation rate increases until the fifth scenario.

Table 6.7 compares the performance of the current tool where DBL is applied and the DBL approach. There is no clear trend over the size scenarios on the impact on the utilisation rates. The only clear trend is that the current allocation tool with DBL increases quicker for the overflow rate than the DBL approach.

scenario	Tetris with DBL				DBL		
	Utilisation		Overflow	Costs	Utilisation	Overflow	Costs
1	7%	-4%	0.11	1.00	12%	0.00	0.99
2	29%	1%	0.20	0.99	28%	0.00	0.95
3	36%	4%	0.25	1.14	32%	0.00	1.07
4	33%	-4%	0.26	1.17	37%	0.03	1.09
5	41%	1%	0.26	1.33	40%	0.15	1.26
6	41%	2%	0.41	1.33	39%	0.39	1.32

Table 6.7: The current tool with DBL is compared against the DBL approach. No clear trend in utilisation is shown over the various scenarios. It does show that the current tool with DBL increases quicker for overflow rates.

Table 6.8 compares the performance of the current tool where Residual Spaces is applied and the Residual Spaces approach. There is no clear trend over the size scenarios on the impact on the utilisation rates. The only clear trend is that the current allocation tool with Residual Spaces increases quicker for the overflow rate than the Residual Spaces approach. The Residual Spaces approach does not have any overflow until the fifth scenario showing that all parcels could be delivered by its solution.

scenario	Tetris with Residual				Residual		
	Utilisation		Overflow	Costs	Utilisation	Overflow	Costs
1	10%	-2%	0.07	0.99	12%	0.00	0.99
2	31%	-2%	0.18	0.99	33%	0.00	0.95
3	41%	3%	0.21	1.13	38%	0.00	1.07
4	37%	-3%	0.22	1.16	40%	0.00	1.08
5	46%	0%	0.22	1.32	46%	0.05	1.24
6	45%	-4%	0.45	1.32	49%	0.49	1.27

Table 6.8: The current tool with Residual Spaces is compared against the Residual Spaces approach. No clear trend in utilisation is shown over the various scenarios. It does show that the current tool with Residual Spaces increases quicker for overflow rates. The Residual Spaces only have an overflow from the fifth scenario.

6.2 Overview with all scenarios averaged

This section details the results from the KPIs that are tested and formulated in Chapter 2. There are five experiments 1. utilisation rate, 2. overflow rate, 3. costs, 4. trip and compartment use, and 5. runtime.

6.2.1 Utilisation rates

The utility here is calculated by dividing the volume used by the available volume, which in combination with the results from Section Hardware where more hardware is used, leads to lower percentages. This is also expected because of the KPI in the previous section.

Figure 13 shows the utilisation rates of the different approaches. It is shown that the current allocation tool has a better performance in terms of route utility than the heuristics. The boxplot also shows that the maximum values for the 3D placement heuristics are lower than the 25th percentile of the current allocation tool. When the approaches are applied to the current allocation tool solution this difference disappears. The averages are now close to each other with the heuristics being better. The minimum and maximum values of the approaches are within the bounds of the minimum and maximum values of the approaches applied to the solution. This means that a more stable solution quality is achieved with less variation.

This can also be explained by taking the lost space into account which does not happen in the current allocation tool. It was to be expected that the 3D approaches have a lower utility rate because here the current approach can be regarded as a relaxed version of the bin packing problem.

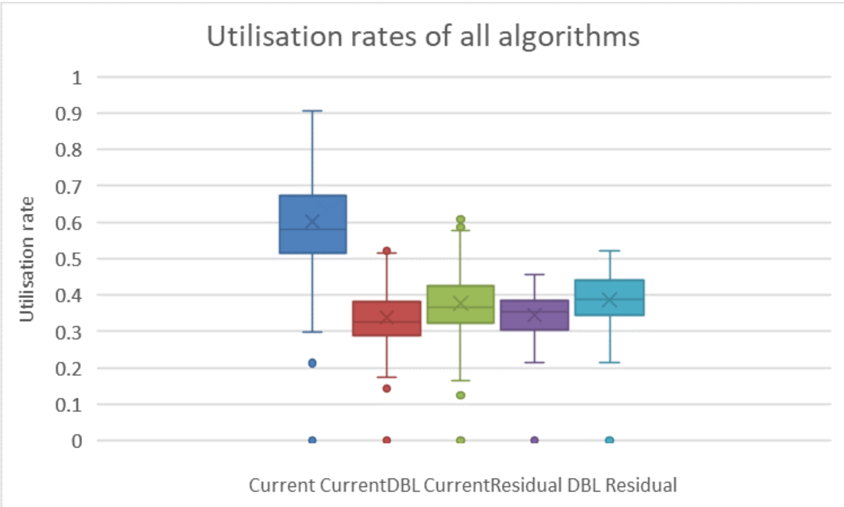
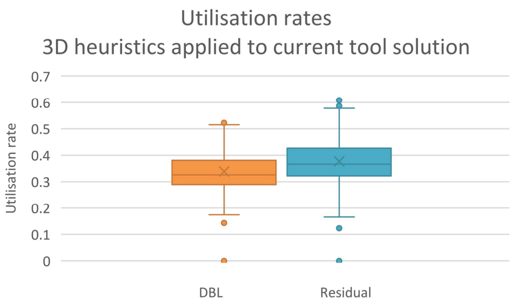
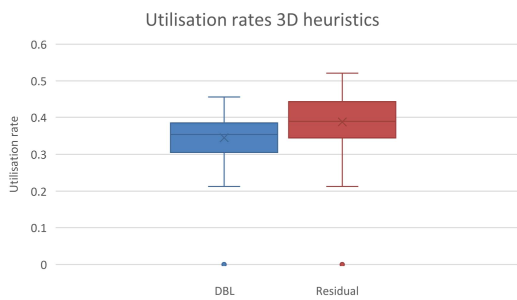


Figure 13: The current allocation tool is outperforming the 3D heuristics as is expected from a relaxed version of the Bin Packing Problem. When comparing the heuristics the difference disappears and are more similar.

Figure 14 shows the utilisation rates when comparing the applied cases and 3D heuristics cases separately. When comparing the two heuristics, it shows that in both the applied case and the normal case the Residual Spaces approach outperforms the DBL heuristic. This was expected as the Residual Spaces approach has an extra step trying to optimise the parcel placement.



(a) Utilisation rate of the 3D heuristics in the applied case



(b) Utilisation rate of the 3D heuristics

Figure 14: The Residual approach where the solution takes optimal orientation of parcels is considered to outperform the Deepest-Bottom-Left approach and is also significant by the ANOVA test.

6.2.2 Overflow

Table 6.9 shows that the exclusion rules of Company X already move a reasonable percentage of parcels to the overflow route. This already implies that some changes could be made here to improve the utility rate.

Figure 15 shows that if only the current allocation tool with the 3D heuristics is compared, the current allocation tool has a lower rate of moving parcels to the overflow routes. This is expected to happen as the utility rates are lower meaning that space is used less efficiently leading to fewer parcels being placed and therefore must be moved to the overflow route.

If a fairer comparison is made by comparing the 3D heuristics with the current allocation tools solution after applying the same approaches to it, the heuristics have less overflow usage on average. This is expected as if a parcel cannot be placed in its assigned compartment it is assumed it cannot fit in its trip and therefore must be serviced by the overflow route.

%	Rules
mean	12.02%
st dev	11.09%
min	0.00%
max	100.00%

Table 6.9: Rate of overflow usage based on company rules compared to all parcels allocated.

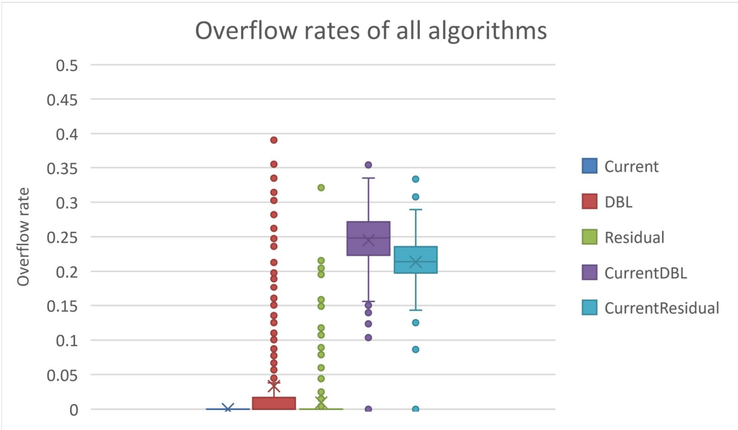


Figure 15: Box plot of the overflow usage per algorithm. The relaxed version of the Bin Packing Problem as the current allocation tool has the lowest overflow rate with the Deepest-Bottom-Left performing worse than the Residual version. The applied solutions have the highest as parcels that could not be placed by the heuristic are moved to overflow.

6.2.3 Costs

Based on the previous KPIs it is expected that the costs are higher based on the calculations from Section 2.4. Figure 16 shows indeed that the costs are all close to each other, but that the applied versions are more expensive. This was expected as only the overflow costs are variable due to the assumptions of costs by the postal company. Considering this, the current allocation tool outperforms the 3D placement heuristics. When the fairer comparison is used the 3D heuristics are better again because of the KPIs.

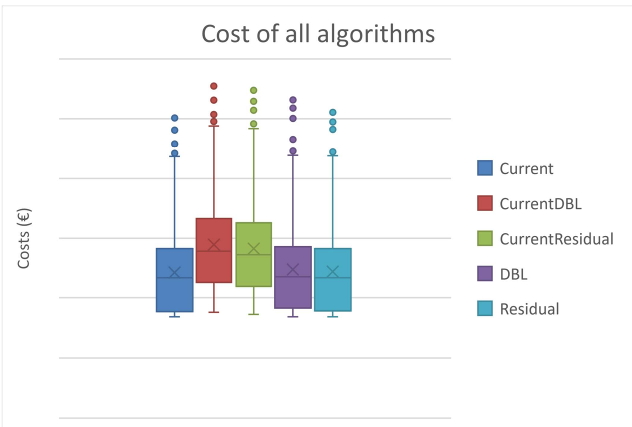
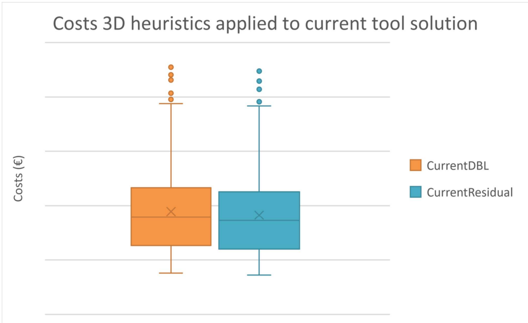
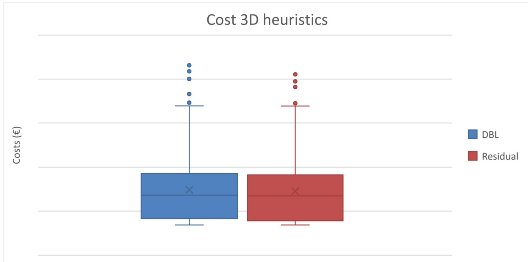


Figure 16: Average costs analysis of all algorithms. The current allocation tool is cheaper on average than the heuristics. In the applied case they are more expensive than the heuristics.

Figure 17 shows the costs when comparing the applied cases and 3D heuristics cases separately. It shows that in the heuristics version, the Residual Spaces approach has on average lower costs and even lower 25 percentile of costs. This difference however is minimal when looking at the applied case of the heuristics because of the higher usage of overflow rates.



(a) Costs for the solutions by the 3D heuristics applied to the current allocation tool solution

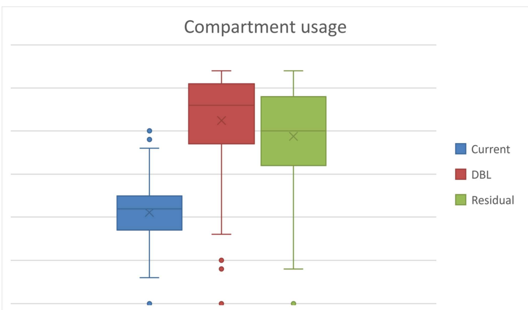


(b) Costs for the solutions by the 3D heuristics

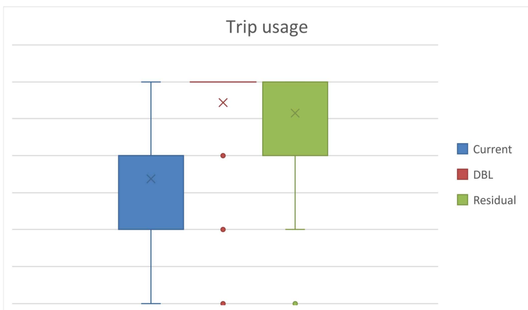
Figure 17: Box plot of the difference in the costs between the algorithms. The plot should be interpreted that a lower value is a cheaper solution. The heuristics are more expensive than the current allocation tool but cheaper than their applied version and the results are also significant by the ANOVA test.

6.2.4 Trip and compartment use

Figure 18 shows that the current allocation tool has a lower trip and compartment usage on average. This is expected as more parcels can be placed in one compartment in the current approach than in the 3D heuristics also seen in the utilisation rates. The route can have a maximum of 3 trips and 27 compartments allocated.



(a) Utilisation rate of the 3D heuristics in the applied case



(b) Utilisation rate of the 3D heuristics

Figure 18: The Residual approach where the solution takes optimal orientation of parcels is considered to outperform the Deepest-Bottom-Left approach and is also significant by the ANOVA test.

6.2.5 Runtimes

As can be seen in Table 6.10, the run times of the 3D placement heuristics are higher on average than the current allocation tool. This is reasonable as their algorithms are more complicated and have more checks done for the various placements within containers than merely checking the volumes of parcels versus the free space in compartments.

seconds	Current	DBL	Residual spaces
mean	0.08	0.55	0.51
stdev	0.03	1.35	2.04
min	0.00	0.00	0.00
max	0.25	15.46	27.34

Table 6.10: Run time analysis of the placement heuristics. The relaxed heuristic of the current allocation tool is the quickest as the least operations have to be done with the Residual spaces taking the most time whilst also having the most operations. Deepest-Bottom-Left (DBL)

6.3 Conclusion

This chapter answers the question *How does the allocation tool compare to the 3D model in terms of performance?*. Concluding, the 3D heuristics have a poorer utilisation rate compared to the current allocation tool in the case where a relaxed approach is compared. However, experimentation shows that the current allocation tool delivers infeasible solutions looking at 3D principles. Experiments show that the problem size does not have a great impact on many KPIs except the utilisation rate. The utilisation rate increases until the number of parcels reaches 250.

If the comparison is made equally when the solution of the current allocation tool is used in combination with the 3D approaches, the 3D heuristics have in general a better performance for utilisation rate, overflow rate, and costs. Between the two 3D heuristics, the Residual Spaces approach where orientation is accounted for performs better in terms of utilisation rate, overflow rate, and costs. This was to be expected as this heuristic specifically aims to minimise the lost space within a compartment or bin.

7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

This section concludes the thesis and answers the main research question. For this thesis, the question from Company X was how a 3D heuristic would impact their current load planning tool. Such a change was requested to get a better idea of how well the current allocation tool performs and if a change would be beneficial. This was formulated into the following research question:

How well does the allocation tool perform compared to a best possible packing plan based on 3D heuristics possible for Company X for the LEVV vans?

The thesis started by detailing the current process. Company X developed an allocation tool. This tool has two functions, allocation of parcels and filtering parcels. The first step is filtering parcels. Parcel data is used to filter out customers and their parcels based on company exclusion rules. The LEVV vehicles are smaller than normal vehicles, so they want to exclude large or heavy parcels, but also customers who have many parcels as they could be better serviced by a larger vehicle. These customers are put into a so-called "overflow" route. This route does not have a predefined set of customers every day.

The second function of the tool is assigning locations to parcels. The tool assigns parcels to a compartment in a roll container based on the parcel volumes and the remaining free volume of a compartment. It uses only the data on the parcel and its customer to assign a place. In the current process, Company X assigns compartments to parcels and not specific locations. There are 3 compartments in a roll-container and each vehicle can hold 3 roll-containers. Each route can only be assigned 3 trips, i.e. be divided into 3 parts, that are driven by the vehicle. This general way of assigning means that the personnel on the work floor have a higher degree of freedom to place parcels within a compartment. The personnel experience little to no problems placing the parcels in their assigned compartments.

The KPIs relevant to Company X have already been implemented in the current allocation tool. The KPIs are the utilisation rate and percentage of volume used for each compartment, trip, and route. Another KPI is the percentage of parcels put to the overflow route.

To better understand the underlying problem behind the tool a literature review was performed. Articles that dealt with 3D loading problems are reviewed specifically. First, the problem classification was done to better understand the problem at hand. The problem is defined as an input minimisation problem, in which a few compartments, roll containers, and trips are to be used. Parcels vary in size which according to Wäscher et al. (2007) means that the problem is a BPP problem. Company X also wanted to have free rotation of parcels so they could be placed optimally, and that the parcels must be removable in a LIFO manner. These qualifiers were corresponding to the problem by Gendreau et al. (2006). Therefore, this thesis used the DBL approach to load the parcels. This approach however does not optimise over rotation or orientation of parcels so the approach of Crainic et al. (2008) and Mahvash et al. (2018) was used to also consider that and not have a bias in the solution.

Before implementing the found approaches, more company requirements were analysed. This was needed to identify the requirements of Company X and the constraints needed for a viable solution. The constraints were all standard for 3D placement, placed wholly within the compartment, with no collision, and enough support below the parcel so that none are floating. The added constraint was the LIFO constraint, to make sure that parcels can be easily removed at their stop.

Company X required that parcels of one customer must be loaded as close together as possible and that the customers were loaded in order of delivery stops. This means that the first customer is placed in the top left compartment in the 3x3 roster of compartments, when that one is full the customers are loaded in the one below that etc., and the last customer is loaded in the bottom right compartment. A customer cannot be split over multiple trips as the multi-dropoff was not allowed.

These constraints and requirements were adapted into the existing logic of the DBL and the EMS of the Residual Spaces. The constraints were implemented by determining if a combination of location and rotation/orientation was valid. The Company X requirements were met by changing the way the customers and their parcels are sorted.

Complexity in both logic and code was added to switch from the current approach to the 3D heuristics. This was required as the current approach only took volume as a criterion but the 3D approaches need to keep track of EMS or EPs and the loading constraints to ensure a valid loading pattern.

Experimentation was done on the developed approaches. To validate the base model, three data sets were used. This was compared to the output of the current allocation tool. Experiments were done on a data set of 289 routes that took place between the 4th of September and the 30th of September. These days do not have extreme demands as no public holidays or major sale events took place in that period.

From the experiments, it is concluded that the current allocation tool almost always has a higher utilisation rate. This is also reflected in the overflow rate and costs, which are also better being both lower in value. As the number of parcels scheduled to be placed increases, the current allocation tool outperforms the 3D heuristics increasingly. It was also concluded that the overflow problem is not caused by the current allocation tool, but rather the exclusion rules that were determined by the company.

The current allocation tool can be regarded as a relaxed version of the bin packing problem. The current tool only uses volume instead of dimensions. Therefore, it is expected that the 3D heuristics would have a lower utilisation rate than the current loading tool.

When comparing the two 3D heuristics, the Residual Spaces approach does outperform the DBL approach as can be expected in terms of utilisation, overflow rate, and costs. This is mainly due to the orientation or rotation optimisation that the Residual Spaces approach has. No other logic between the two algorithms is different.

In conclusion, the current allocation tool performs better than the heuristic in terms of utilisation rate and overflow rate. This is when the freedom of employees is taken into account. This freedom results in a 30% increase on average in performance for utilisation rate. This is tested with the difference between the current tool solution and the 3D heuristic applied. However, experiments show that these solutions are not feasible if we account for 3D practices. Therefore the comparison is made between the current allocation tool with the 3D heuristics applied and the 3D heuristics. In both applications, the Residual Spaces approach is better in terms of overflow rate and utilisation rate.

The conclusion can be defined as that the current allocation tool performs well in terms of utilisation rate, overflow rate, and costs, but the solutions are infeasible for 3D placement. The Residual Spaces approach performs similarly in terms of utilisation rate when the current tool solution is adapted for 3D. The Residual Spaces perform better in terms of overflow rate and still create a valid solution.

7.2 Recommendation

This thesis recommends not changing the current allocation tool due to the current process with loading the parcels. It performs better than the best possible packing plans of the heuristics tested in this thesis. The current tool does create infeasible solutions which is recommended to investigate when a change is needed. In reality, with the current allocation tool, parcels are being loaded on their trip. When humans pack the parcels, creativity and experience help the loading efficiency, as mentioned in Section 2.5. This was proven on the work floor of the visited warehouse that humans can pack the parcels in their assigned compartments without breaking geometry rules. Strict adherence to LIFO was not checked in this case.

Another reason why the 3D heuristics are not feasible for the current situation is the random arrival of parcels from the conveyor belt. This randomness creates problems when parcels have to be stacked. If parcels arrive in the wrong order, then parcels would need to be handled twice, once to place temporarily and secondly to the final place. It is recommended to research if it is possible to make the arrival process pre-determined by changing how the parcels are transported within the sorting depots.

This thesis recommends Company X to only further research 3D heuristics if the company wants to automate the loading process as only then it is necessary to have a very detailed loading schedule. Previous developers have noted that the level of freedom is important for humans to retain enjoyment in the job and their innate ability to pack more parcels and optimally orientate them. Also, it is recommended to loosen some of the company rules regarding the placement and ordering of parcels within the roll containers and compartments. They currently restrict the solution space and therefore limit the performance of the 3D heuristics.

If Company X wants to further develop the current allocation tool to improve the utilisation rate, the thesis recommends applying improvement heuristics for swapping and moving parcels between the compartments to maximise the utilisation rate of each compartment. This retains the possibility for humans to pack the roll containers with creative freedom in orienting and placing parcels.

7.3 Future research

As mentioned before, the current approach works best for loading processes with human actors, but the impact of speed and efficiency should be investigated when the automation of these processes is done. It is important to first determine if the current roll containers that are used by Company X are packable by robots or if other types of bins should be used if Company X wants to automate. Besides that research must be done on the parcels itself. If the parcels are too fragile or oddly shaped they might not be loadable with machines except when using loading trays as currently seen in other automated warehouses. This would have an impact on the utilisation rates and might be combined with different roll containers that could better facilitate the loading trays.

If Company X wants to increase the utilisation rate of the LEVVs, further research should be done on loosening the current constraints and restrictions. The most constraining one is the proximity requirement for parcels belonging to one customer. The other requirement that has an impact on the loading schedule is the ordering approach where the first customer in the route is placed top left and the last customer bottom right. Company X already has the necessary tools to facilitate easy unloading at a customer with their handheld device that indicates the compartment where the parcel is located. The major impact of this change would be the time required to unload at the stop and collect the parcels. This time increase would be compared against the utilisation rate and possible route efficiency if more customers could be serviced with one trip. Loosening these company requirements would create more flexibility for the heuristics. Currently, the heuristics cannot reach the same utilisation rate as the current tool could within a reasonable time. This is because the current version is a relaxation of the BPP which creates an optimum solution, requiring more computational effort than Company X could spend on a process that happens many times on a daily basis.

REFERENCES

Ali, S., Ramos, A. G., Carravilla, M. A., and Oliveira, J. F. (2022). On-line three-dimensional packing problems: a review of off-line and on-line solution approaches. *Computers & Industrial Engineering*, page 108122.

Balakirsky, S., Proctor, F., Kramer, T., Kolhe, P., and Christensen, H. I. (2010). Using simulation to assess the effectiveness of pallet stacking methods. In *Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference, SIMPAR 2010, Darmstadt, Germany, November 15-18, 2010. Proceedings 2*, pages 336–349. Springer.

Bischoff, E. E. and Ratcliff, M. (1995). Issues in the development of approaches to container loading. *Omega*, 23(4):377–390.

Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.

Carpenter, H. and Dowsland, W. (1985). Practical considerations of the pallet-loading problem. *Journal of the Operational Research Society*, 36(6):489–497.

Ceschia, S. and Schaerf, A. (2013). Local search for a multi-drop multi-container loading problem. *Journal of Heuristics*, 19:275–294.

Chen, C., Lee, S.-M., and Shen, Q. (1995). An analytical model for the container loading problem. *European Journal of operational research*, 80(1):68–76.

Christensen, H. I., Khan, A., Pokutta, S., and Tetali, P. (2017). Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79.

Crainic, T. G., Perboli, G., and Tadei, R. (2008). Extreme point-based heuristics for three-dimensional bin packing. *Informs Journal on computing*, 20(3):368–384.

Den Boef, E., Korst, J., Martello, S., Pisinger, D., and Vigo, D. (2005). Erratum to “the three-dimensional bin packing problem”: robot-packable and orthogonal variants of packing problems. *Operations Research*, 53(4):735–736.

Dereli, T. and Sena Das, G. (2010). A hybrid simulated annealing algorithm for solving multi-objective container-loading problems. *Applied Artificial Intelligence*, 24(5):463–486.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European journal of operational research*, 44(2):145–159.

Eley, M. (2003). A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum*, 25:45–60.

Erbayrak, S., Özkır, V., and Yıldırım, U. M. (2021). Multi-objective 3d bin packing problem with load balance and product family concerns. *Computers & Industrial Engineering*, 159:107518.

Fanslau, T. and Bortfeldt, A. (2010). A tree search algorithm for solving the container loading problem. *INFORMS Journal on Computing*, 22(2):222–235.

Fuellerer, G., Doerner, K. F., Hartl, R. F., and Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3):751–759.

Gehring, H. et al. (1997). A genetic algorithm for solving the container loading problem. *International transactions in operational research*, 4(5-6):401–418.

Gendreau, M., Iori, M., Laporte, G., and Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350.

George, J. A. and Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3):147–156.

Gzara, F., Elhedhli, S., and Yildiz, B. C. (2020). The pallet loading problem: Three-dimensional bin packing with practical constraints. *European Journal of Operational Research*, 287(3):1062–1074.

Ha, C. T., Nguyen, T. T., Bui, L. T., and Wang, R. (2017). An online packing heuristic for the three-dimensional container loading problem in dynamic environments and the physical internet. In *Applications of Evolutionary Computation: 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part II 20*, pages 140–155. Springer.

Hemminki, J., Leipala, T., and Nevalainen, O. (1998). On-line packing with boxes of different sizes. *International journal of production research*, 36(8):2225–2245.

Hodgson, T. J. (1982). A combined approach to the pallet loading problem. *IIE Transactions*, 14(3):175–182.

Jin, Z., Ito, T., and Ohno, K. (2003). The three-dimensional bin packing problem and its practical algorithm. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 46(1):60–66.

Karabulut, K. and İnceoğlu, M. M. (2005). A hybrid genetic algorithm for packing in 3d with deepest bottom left with fill method. In *Advances in Information Systems: Third International Conference, ADVIS 2004, Izmir, Turkey, October 20-22, 2004. Proceedings 3*, pages 441–450. Springer.

Kho, M. (2018). *Analysis of new models of the three-dimensional bin packing problem for the application to the parcel delivery service industry*. PhD thesis, University of Groningen.

Kischka, P., Lorenz, H.-W., Derigs, U., Domschke, W., Kleinschmidt, P., Möhring, R., Bortfeldt, A., and Gehring, H. (1998). Applying tabu search to container loading problems. In *Operations Research Proceedings 1997: Selected Papers of the Symposium on Operations Research (SOR'97) Jena, September 3–5, 1997*, pages 533–538. Springer.

Koch, H., Bortfeldt, A., and Wäscher, G. (2018). A hybrid algorithm for the vehicle routing problem with backhauls, time windows and three-dimensional loading constraints. *OR spectrum*, 40:1029–1075.

Kocjan, W. and Holmström, K. (2006). *The AUTOPACK project-Algorithms for container loading*. Citeseer.

Krebs, C., Ehmke, J. F., and Koch, H. (2021). Advanced loading constraints for 3d vehicle routing problems. *OR Spectrum*, 43(4):835–875.

Liu, J., Yue, Y., Dong, Z., Maple, C., and Keech, M. (2011). A novel hybrid tabu search approach to container loading. *Computers & Operations Research*, 38(4):797–807.

Loh, T. (1992). A packing algorithm for hexahedral boxes. In *Proceedings of the Industrial Automation 92 Conference, Singapore*.

Mack, D. and Bortfeldt, A. (2012). A heuristic for solving large bin packing problems in two and three dimensions. *Central European Journal of Operations Research*, 20:337–354.

Mahvash, B., Awasthi, A., and Chauhan, S. (2018). A column generation-based heuristic for the three-dimensional bin packing problem with rotation. *Journal of the Operational Research Society*, 69(1):78–90.

Mara, S. T. W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., and Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, page 105903.

Otten, M., Meerwaldt, H., and den Boer, E. (2015). Segmentering co2-emissies goederenvervoer in nederland.

Pace, S., Turky, A., Moser, I., and Aleti, A. (2015). Distributing fibre boards: a practical application of the heterogeneous fleet vehicle routing problem with time windows and three-dimensional loading constraints. *Procedia Computer Science*, 51:2257–2266.

Pan, L., Chu, S., Han, G., and Huang, J. (2009). A tree-based wall-building algorithm for solving container loading problem with multi-drop constraints. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 538–542. IEEE.

Saraiva, R. D., Nepomuceno, N., and Pinheiro, P. R. (2015). A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company. *IFAC-PapersOnLine*, 48(3):490–495.

Tarantilis, C. D., Zachariadis, E. E., and Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):255–271.

Tomazella, C. P. and Nagano, M. S. (2020). A comprehensive review of branch-and-bound algorithms: Guidelines and directions for further research on the flowshop scheduling problem. *Expert Systems with Applications*, 158:113556.

Tresca, G., Cavone, G., Carli, R., Cerviotti, A., and Dotoli, M. (2022). Automating bin packing: a layer building matheuristics for cost effective logistics. *IEEE Transactions on Automation Science and Engineering*, 19(3):1599–1613.

van Amstel, W. P., Balm, S., Warmerdam, J., and Peters, T. (2018). Urban technology research programme lefv logic: Research on light electric freight vehicles.

Wäscher, G., Haußner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130.

Wei, L., Zhang, Z., and Lim, A. (2014). An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *IEEE Computational Intelligence Magazine*, 9(4):18–30.

Zhang, Z., Wei, L., and Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological*, 82:20–35.

Zhao, X., Bennell, J. A., Bektaş, T., and Dowsland, K. (2016). A comparative review of 3d container loading algorithms. *International Transactions in Operational Research*, 23(1-2):287–320.

Zwep, L. (2023). Optimizing the packing strategy for parcel delivery vans.