Report number: 2021.MME.8540

An optimization model for 3D loading space for the transport of large steel structures

Master Thesis - Jasper Krombeen (4464745)

Delft University of Technology

An optimization model for 3D loading space for the transport of large steel structures

by

Jasper Krombeen

Master Thesis

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology

to be defended publicly on Friday September 17, 2021 at 10:00 AM

MSc track:	Multi-Machine Engineering	
Supervisor:	Dr. ir. X. Jiang	
Thesis committee:	Dr. ir. D. L. Schott	TU Delft committee Chair, 3mE
	Dr. ir. M. B. Duinkerken	TU Delft committee member, 3mE
	K. Oudshoorn	company Supervisor, Oostingh
	G. de Vree	company Supervisor, ASK Romein
Date:	September 30, 2021	

Preface

My first experience with the programming of algorithms was during the first year of my master Multi-Machine Engineering - which is not too long ago - during a course called Quantitative Methods for Logistics. It was an obligatory subject for me, but I have considered it one of my favorite subjects ever since. To finish the very same master by creating an algorithm completely from scratch is great. What makes it a fantastic opportunity is that I was able to do work on a real-life problem, at the company of ASK Romein, while at the same time gaining experience.
This master thesis also marks the end of my time as a student, and the start of my working career. I am looking forward to the next step in my life, and am determined to finish my time at the TU Delft with a great result.

This research assignment is part of the second year of the master Mechanical Engineering, for the master track Multi-Machine Engineering, at Delft University of Technology.

I would like to thank everyone I worked with at ASK Romein Roosendaal and ASK Romein-Oostingh Katwijk, and would in particular like to express my gratitude to Glenn de Vree, Kees Oudshoorn, Leon van der Plas and Erron Estrado for their guidance during this assignment. Their comments were of great value and I feel that their support helped me take this report to a higher level. I would also like to mention my supervisor, dr. ir. Xiaoli Jiang, for her feedback.

Jasper G. P. Krombeen
Bergen op Zoom, August 2021

Front page image: *loading sequence ASK Romein Roosendaal*

Abstract

In this report, a model is presented that automates and optimizes the loading processes for the transport of steel structures at the company of ASK Romein. This transport can be captured in two main loading processes: the loading division, which consists of allocating items to different trailers, and the loading sequence, which considers placing the items onto their respective trailer.
The possibility of automating the process of the loading division is investigated. This includes the digital generation of the loading sequence process. Achieving a form of automation would both mean a reduction in time required to create a loading division as well as allow for optimizing the number of required trailers.
Using an extensive literature review as well as an in-depth investigation of the current situation at the company, a model is developed, which consists of an ALNS heuristic responsible for the loading division, and a new function, the layer heuristic, proposed in this report, that digitally generates the loading sequence.
The proposed model is validated using various experiments on real-life data, including several sensitivity analyses. For the used data set, the model is able to reduce the existing loading division by at least 24%, and the computation time is superior to the current time required to create a loading division. Because the loading sequence process is created digitally, the model is capable of checking all the loading conditions, such as axle loads, even before the actual loading sequence has taken place.

Contents

List of Figures

1 Introduction

The upcoming section serves as an introduction to this report. First, the subject of this research is given, after which the research question is provided. Next, the report structure is presented by defining the sub questions.
This master thesis is executed at the company ASK Romein. ASK Romein is a major player in the Dutch and Belgian steel construction industry. They have the expertise to design and develop large steel structures, like data centers, football stadiums, distribution centers and more.

1.1 Subject of research

For each project carried out at ASK Romein, there are many different types of steel assemblies, in all kinds of shapes. This makes the cargo loading process a complex procedure: it is not as straightforward as loading a set of boxes.

When it comes to loading, there are two different processes distinguishable:

1. First, during the calculation phase of a project, all components and assemblies are assigned to a transport lot. In the remainder of this report, this will be referred to as the loading division process. The loading division conditions (discussed in subsection 2.3.2) are evaluated during the loading division process.
2. The second process concerns the position and orientation of the assemblies when placed onto the trailer. This process is currently executed during the production phase, when the loading occurs. It is in this phase that the loading sequence conditions (subsection 2.3.3) can be evaluated. This process will be referred to as the loading sequence process.

Both processes will be further explained in section 2.

This research project revolves around the two loading processes. The possibility of automating the process of loading division is investigated, to decrease the number of trailers needed for transport and decrease the time needed to create the loading sequence. Furthermore, the loading sequence process is generated digitally and implemented in the loading division process. This means that all loading conditions can be checked as early as the calculation and design phase.

1.2 Research question

The following research question is defined for this assignment:

> How can the loading processes of trucks for steel structure transport be automated and optimized?

1.3 Report structure

The aforementioned research question is accompanied by a set of sub questions, that are discussed throughout this report.

In the next section, the existing loading processes at ASK Romein are analyzed. The two loading processes are reviewed in more depth, and an overview of the required loading conditions is discussed. The research problem definition is provided, and an answer is given to the first sub question:

1. What is the state-of-the-art of the loading processes at ASK Romein?

The next sub question is discussed in section 3:

2. What are the existing solutions of 3D space maximization problems?

This section contains several 3D problems from the literature. In section 4, the proposed model is presented. Sub questions 3 and 4 are answered in this section:

3. How can the loading division process be automated, accounting for the loading conditions?

4. How can the presented loading sequence problem be implemented in a 3D optimization model?

In section 6, the verification and validation are given, so that sub questions 5, 6 and 7 can be answered:

5. How can the loading division and loading sequence models be verified?

6. How can the loading division and loading sequence models be validated?

Finally, experiments are conducted on the proposed model, to provide an answer for sub question 7:

7. How do the parameters of the model influence the solution quality and computation time?

Finally, a conclusion is provided in section 7.

Figure 1 is a graphic representation of the structure of this report.

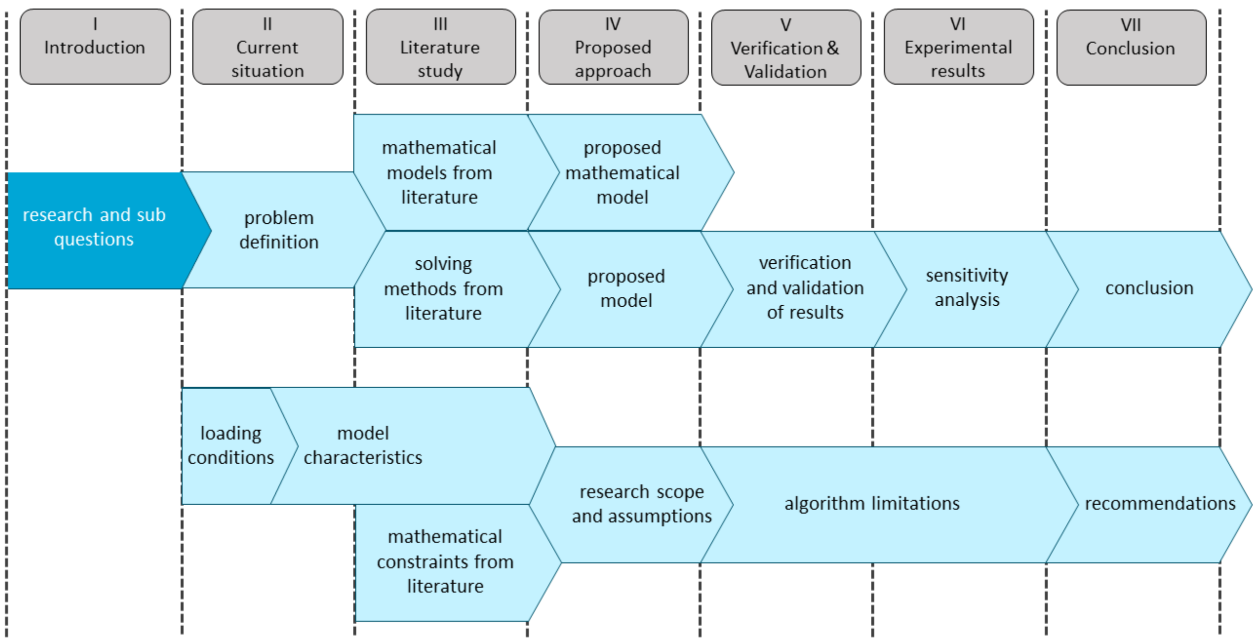


Figure 1: Report structure

2 Current loading processes at ASK Romein

In this section, the existing loading processes at ASK Romein are evaluated. Figure 2 shows the main topics of this section.
First, the different project phases at ASK Romein are briefly discussed. The two loading processes are reviewed in more depth. Next, the problem definition is given, and the loading conditions required for truck loading are discussed. Finally, with the problem definition and the loading conditions in mind, the model characteristics are presented.

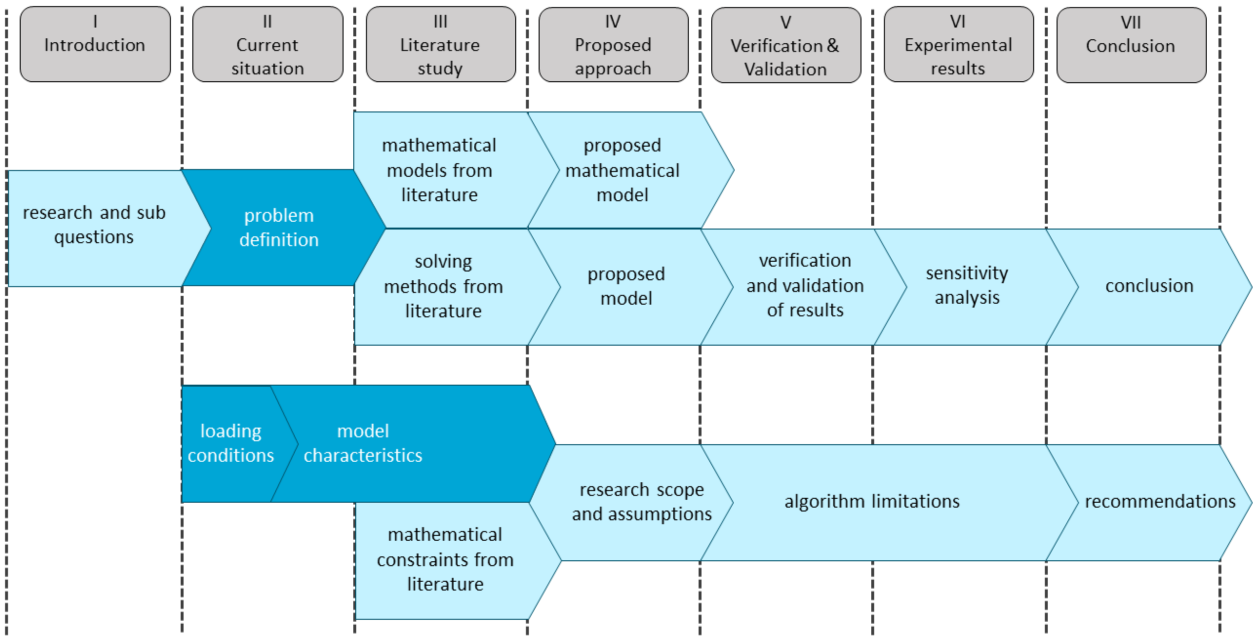


Figure 2: Report structure

2.1 Introduction to ASK Romein

ASK Romein is a major player in the Dutch and Belgian steel construction industry. As mentioned before, they design and develop large steel structures, like data centers, football stadiums, distribution centers and more. ASK Romein operates from multiple locations (see figure 3) in The Netherlands and Belgium, but they execute projects in many other countries, such as Denmark and Italy.

Figure 3: Overview of ASK Romein facilities [1]

[1]Image courtesy of ASK Romein.

At ASK Romein, multiple departments are involved during a project. Each department has its own role and responsibility. The different phases of a project will be discussed in the next paragraphs. Currently, there are two different processes related to transport planning. They will be presented in their respective phases.

2.1.1 Calculation and design

Usually, a client approaches ASK Romein with a rough design idea. At the calculation department, an estimate for the costs is determined. At the engineering department, the detailed design is created. This consists of drawing the separate components and calculating loads.
Next, the department of planning makes sure everything is ready for the construction phase. One of the functions of this department is the loading division process to prepare the loading division.

Loading division

The loading division is an important factor in the design process at ASK Romein. Besides loading the transport trucks according to this division, production also takes place in the order created by the loading division. Therefore, a good loading division is required as early as possible, preferably right after the design has been established.
Currently, the department of planning determines the loading division manually. The design of a project is created in a building information model (BIM) environment called Tekla. A built-in function is used to assign different construction items to a transport lot, one item at a time, conform the conditions mentioned in section 2.3.2. Most of the time, these conditions are the dominant constraint. Figure 4 shows an image of the current loading division process. A project design is shown in Tekla. The different colored items represent different components. Items are assigned to a transport lot. In the example, the items in a certain transport lot are highlighted in yellow.

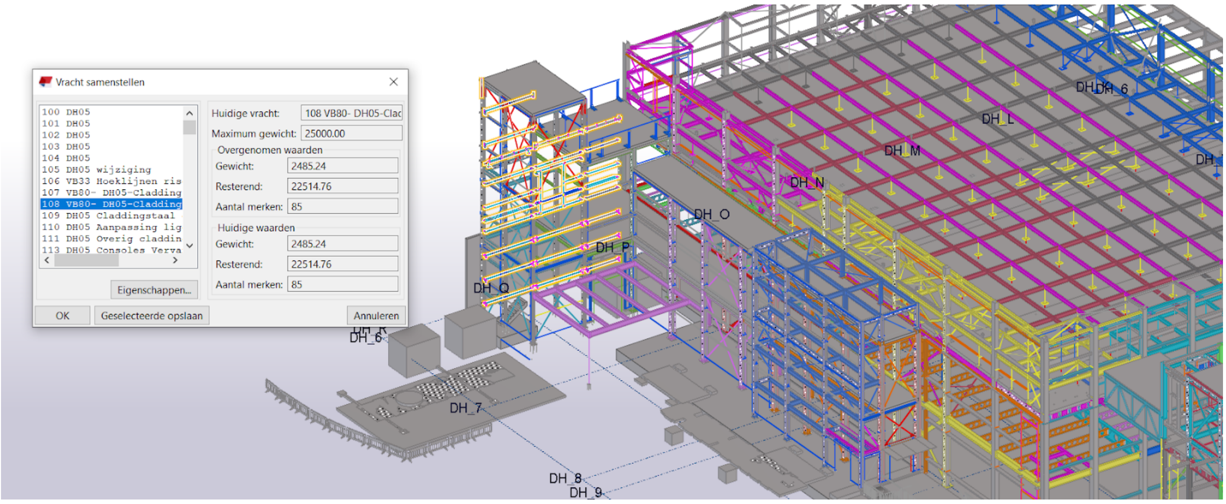


Figure 4: Current loading division process [1]

As each lot is created manually, the loading division process is very time-consuming, especially for large-scale projects. This is the first problem of this research.

[1]Image courtesy of ASK Romein.

2.1.2 Production

During the production phase, the designed assemblies and other components are produced. An assembly enters the factory as a standard steel profile, called position. As mentioned before, positions enter the factory halls in the same composition as the lot they have been assigned to in the loading division process. Next, the positions are cut to the right size, holes are drilled, and steel plates called gusset plates are attached. A finished component is now called 'assembly'.
Subsequently, the conservation system can be applied to the component. Sometimes, conservation of assemblies takes place in the same location as the production facility. In other cases, when conservation is outsourced, the assemblies have to be transported to another coating company.

When all assemblies belonging to the same lot have been produced, the loading onto the trailer conform the loading sequence can be started.

Loading sequence

Currently, the loading sequence process is executed manually. The items are placed via the use of large overhead cranes.

Besides the loading division criteria that were mentioned before, there are many more requirements for the loading of a trailer: these are the conditions related to the loading sequence, to be discussed in section 2.3.3. These requirements can only be checked when the loading sequence has taken place. When this process is automated, using a model that determines the 3D orientation of all assemblies on a trailer, all criteria can be checked using the digital representation. Additionally, this model can be used for the optimization of 3D loading space.

Figure 5: Loading sequence: item placement on a trailer [1]

[1]Image courtesy of ASK Romein.

2.1.3 Construction

When the assemblies arrive on the construction site, they will be put together to form a steel structure, like, for example, a data center. For the department of construction, it is important that the items arriving on site are in the same building phase, to ensure a fast erection process and avoid having to store large amount of components that cannot yet be used.

2.2 Problem definition

Figure 6 shows the current project phases, including a graphical representation of the loading division and loading sequence processes, and when these processes occur.

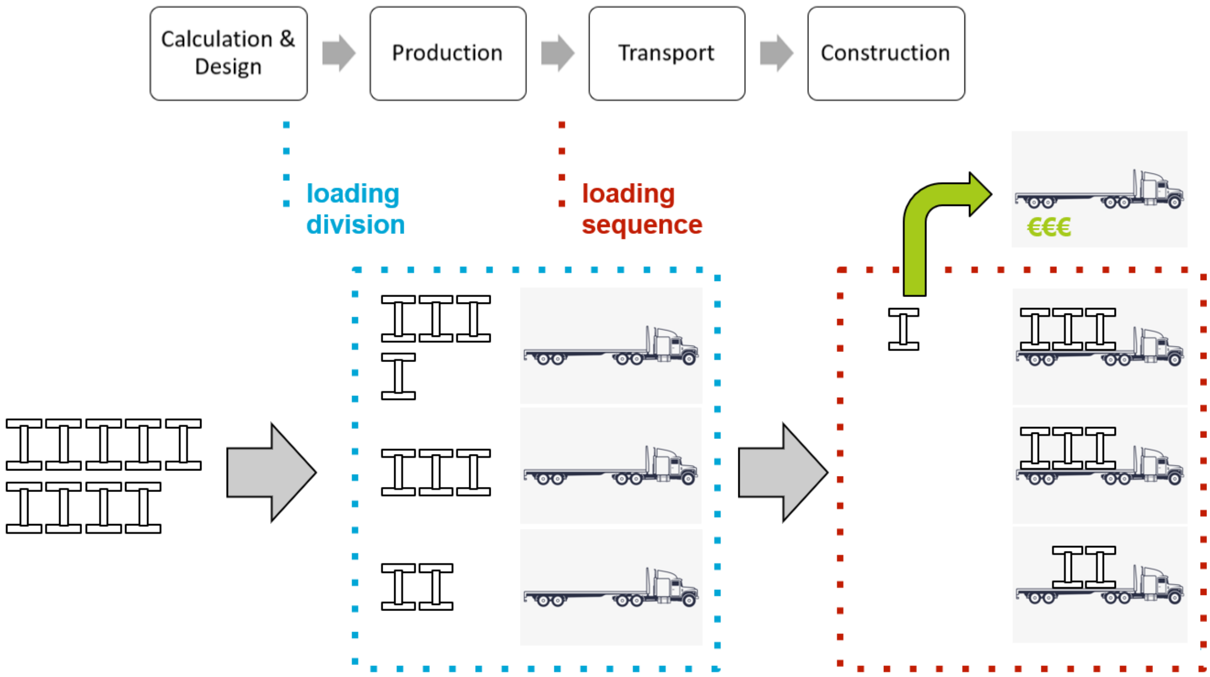


Figure 6: Overview of current loading processes

- In the example shown in figure 6, the project contains 9 items. At the end of the calculation and design phase, these items are allocated to a certain transport lot during the loading division process.
- Next, the 9 items are cut, welded and modified in the production halls during the production phase. When all items of a one transport lot are finished, the loading sequence is started.
- In this example, it occurs that the first trailer is not able to fit all the items allocated to its transport lot: the available trailer space has been overestimated. This mistake is recognized at the end of the production phase. A fast solution is required to prevent disrupting the production and construction phases, which usually results in ordering an extra trailer.
- A last-minute extra trailer is quite costly and is not considered in the initial cost analysis. However, what is worse is that the third trailer on this example appears to have space left: its available space has been underestimated. Had the overestimation of the first trailer been known beforehand, re-arranging the transport lots maybe would have resulted in only needing three trailers instead of four.

In this section, two main problems have been defined:

1. Manual execution of the loading division, which costs time.
2. Over- and underestimation of the available loading space during the loading division process, which costs money.

Figure 7 shows a proposed solution to the given problems:

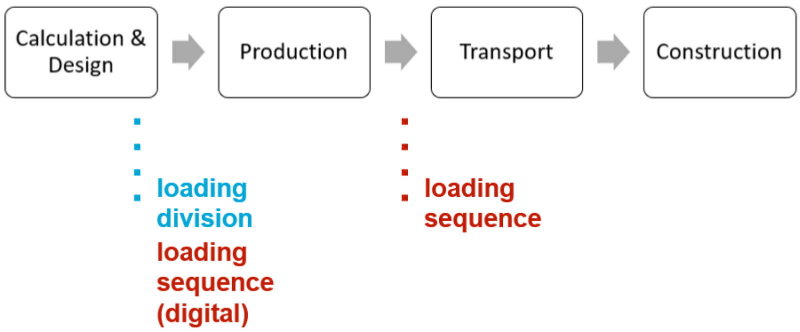


Figure 7: Overview of proposed loading processes

The solution consists of two components:

1. Automation of the current loading division process.
2. Inclusion of digital generation of the loading sequence in the loading division process.

These solutions form the basis of this research assignment.

Digitally generating the loading sequence process means that the loading sequence conditions can be checked. All loading conditions will be discussed in the next subsection. These are the conditions that are part of the current loading division process and loading sequence process at ASK Romein.

2.3 Loading conditions

In the previous paragraphs of this section, the term loading conditions has been mentioned multiple times. There are two types of loading conditions. The fact that the second type, the loading sequence conditions, cannot be checked during the loading division process, is the direct reason for one of the defined problems, the over- and underestimation of loading space.
An example of such a condition, that cannot be checked during the loading division, is that all items have to fit inside the trailer boundaries. Usually, the total weight of all items, which is a loading division condition, gives a good indication for the loading constraints: when the weight is under a certain value, all assemblies fit on a standard trailer. But there are exceptions. For example, for lightweight complex-shaped assemblies, volume is the restricting constraint instead of weight.
The transport planners responsible for the loading division use their experience to identify these exceptional cases. However, since they do this manually, sometimes, the exception is not recognized: this results in the aforementioned overestimation of the available loading space, and thus, ultimately, in an increase in costs.

In this subsection, the involved stakeholders and all conditions concerning the loading processes will be discussed. The first list of conditions are currently checked during the loading division process. The second list of conditions need to be verified after the loading sequence process has taken place.

2.3.1 Stakeholders

In this section, an overview of the involved stakeholders of this research is given, using the power-interest matrix defined by Mendelow [7]. Figure 8 shows the stakeholders:

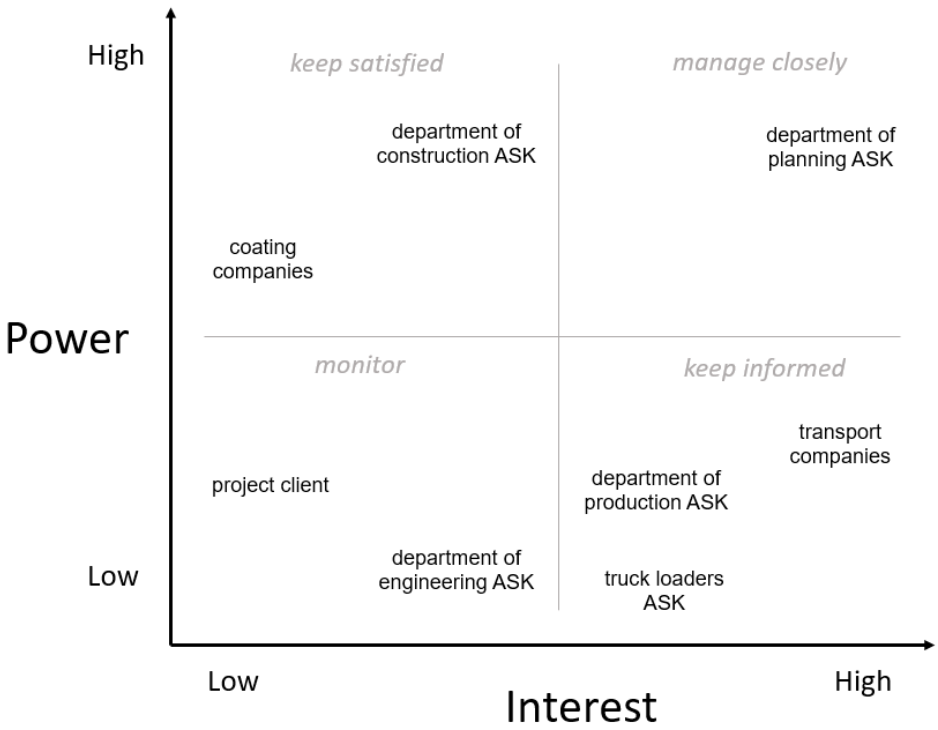


Figure 8: Stakeholders loading division and loading sequence [1]

In the top right area of figure 8, the most important stakeholders are listed: the department of planning of ASK Romein. They have the highest interest, because of their responsibility for the current loading division, and the highest level of influence on the shape of this research.
Moving to the top left, the high power - low interest stakeholders are the department of construction and the coating companies. They are not really interested in a new loading division, but have a high influence on the result. For example, the coating companies require only one conservation system per transport lot, and the department of construction requires a loading division according to the erection sequence, both of which introduce extra constraints for the model.
In the bottom right corner, the department of production, transport companies and truck loaders are located. These stakeholders have little influence on the project, but a high interest: for the production department and truck loaders, an improved loading division can have a positive effect on the work flow, because less time is needed for the loading sequence. Transport companies may receive less last-minute truck orders due to the proposed model.
Finally, the stakeholders with the least interest en power are shown in the bottom left. These are the project client, which only benefit from a new loading division by paying less transport costs, and the department of engineering at ASK Romein.

2.3.2 Loading division conditions

The conditions considered when creating the loading division are the maximum weight, conservation system, erection sequence and trailer dimensions:

[1] Image adapted from Mendelow [7]

- Maximum weight
  The transport of goods is constrained by a maximum weight. This maximum weight depends on multiple characteristics:
  - Dimensions of the trailer: the trailers used at ASK Romein can be extended (figure 9), to allow longer assemblies. An extended trailer naturally has a lower maximum weight tolerance. Some trailers carry a crane (figure 10), which reduces the available loading space.

Figure 9: Double extension trailer [1]

Figure 10: Trailer-crane combination [1]

[1]Image courtesy of ASK Romein

- Characteristics of the trailer: the trailers employed by ASK Romein make use of a flatrack, which eases the loading process, but adds extra weight.
  Furthermore, sometimes auxiliary measures, such as wooden packing sticks, are required to secure the assemblies on a trailer. These have a relatively small weight.
- Legislation: each country that is visited along the transport route has stated a maximum allowed total weight of a trailer and truck combination. The legislation of the maximum weight for standard trailers for three countries is shown in table 1:

Country:	Maximum weight (tonnes):
The Netherlands	50
Belgium	44
Germany	40

Table 1: Maximum allowed total weight of truck combination for several countries in Europe

- Conservation system
  All steel parts for a project need a conservation finish before they can be erected. The finish could be one of many types of coating or galvanizing. Because some coating types are applied in different locations, and each type has different characteristics (number of layers, process time), it is important to load components that have the same finish together. The conservation systems used in this report are:
  - No treatment (code —)
  - Machine blasting (code M)
  - Fire resistant coating (code BW), treated for 30, 60, 90 or 120 minutes
  - Galvanizing (code T)
  - Paint coating (code N), for 1 up to 5 layers.

  So for example, if the conservation system is MN2, it means that an assembly receives machine blasting as well as 2 layers of paint. If the conservation system is BW60, the assembly receives 60 minutes of fire resistant treatment.
- Erection sequence
  On site, the assembly of several components happens in the order of construction: the erection sequence. It is beneficial that the loading order corresponds to this erection sequence. The importance of following the erection sequence also depends on the available space during the assembly process: if the available unloading space is little, the erection sequence should be strictly followed, as re-ordering of components is not possible on site.
- Trailer dimensions
  Naturally, a component to be loaded onto a trailer should never exceed the maximum allowed loading dimensions of the trailer. If the assembly's length exceeds the allowable dimension, the trailer can be extended. For excessively large items, such as trusses, exceptional transport may be necessary.

2.3.3 Loading sequence conditions

The loading sequence conditions are the maximum dimensions of the total cargo, weight distribution, order of (un)loading, indivisible load and item securing:

- Maximum dimensions of total cargo
  When placing multiple assemblies on a single trailer, one should pay attention to the overall volume of the assemblies: sometimes, lightweight assemblies have complex shapes, meaning only a few of them can be placed on a trailer, long before the maximum weight is reached.

- Weight distribution
  Besides the total weight of the cargo, the way the weight is distributed is also constrained. The load distribution can be characterized using the fifth wheel and axle load. These loads on the fifth wheel and rear trailer axles are related to the maximum weight as well as the location and orientation of the items on a trailer (center of mass). The fifth wheel load is defined as the loading on the pin of the trailer.

- Order of (un)loading
  As mentioned before, the loading of items during the loading division process is usually executed using an overhead crane: vertical loading. However, on a construction site, unloading can both happen in vertical (crane) and horizontal (forklift) direction. This means that all items should be placed with sufficient spacing both in horizontal and in vertical direction.

- Indivisible load
  Another important factor to be considered for the loading of large trailers is the rule of indivisible load. This means that for trailers longer than 13.60 m, the length of the longest item placed on the trailer cannot be exceeded by a combination of shorter items.

- Item securing
  Most components placed on a trailer are extremely heavy, which means that they cannot move easily. However, in the case of emergency braking, it is still possible that sliding occurs if the items are not secured correctly. To restrict item movement, securing of items requires that all items are placed against the headboard, and that strap bands are used to tighten the items in width direction.

2.4 Model characteristics

In table 2, the aforementioned conditions are converted to model characteristics. The table also shows which characteristics are currently included in the loading division process, and which characteristics are included in the proposed loading division. The characteristics listed in this table will be used in the remainder of this report.

Characteristic:	Description:	Included in traditional loading division:	Included in proposed loading division:
	trailers		
1. Maximum weight	limit on overall truck weight	✓	✓
2. Axle and fifth wheel loads	limit on axle loads		✓
3. Length	multiple trailer sizes	✓	✓
4. Width, height	fixed trailer dimensions	✓	✓
	items		
5. Flatrack and trailer gap	smaller items can fall through trailer gap		✓
6. Item securing	restrict item movement		✓
7. Item stability	stacking items on larger other items		✓
8. Item orientations	fixed orientation for some items		✓
	processes		
9. Conservation finish	different finishes to different coaters	✓	✓
10. Erection sequence	limited storage on construction site	✓	✓
11. Loading and unloading	(un)loading can be horizontal or vertical		✓

Table 2: Model characteristics

2.5 Discussion

The sub question defined for this section is the following:

1. What is the state-of-the-art of the loading processes at ASK Romein?

The purpose of this question is to give insight to the situation at ASK Romein, to provide a problem definition, and to present a starting point for this research.
In this section,an introduction has been given, consisting of the different project phases at ASK Romein, as well as the two different transport-related processes. Both processes, the loading division and loading sequence, use many conditions. These loading conditions have been used to obtain a list of model characteristics that can be used to define the model constraints later in this report.

3 Available methods for loading division and loading sequence models

As figure 11 shows, in this section, a literature review is provided to gain insight in the existing solutions to create a loading division and a loading sequence. In the field of optimization engineering, studies related to these types of solutions are called cutting and packing problems, and in particular the Multiple Bin Size Bin Packing Problem (MBSBPP). First, an introduction is given to the typology of cutting and packing problems, to connect the literature definitions to the typology of this report. Subsequently, available mathematical models are investigated, after which several solution methods applicable to the problem of this study are presented and evaluated, so that the most promising methods for this research are identified.

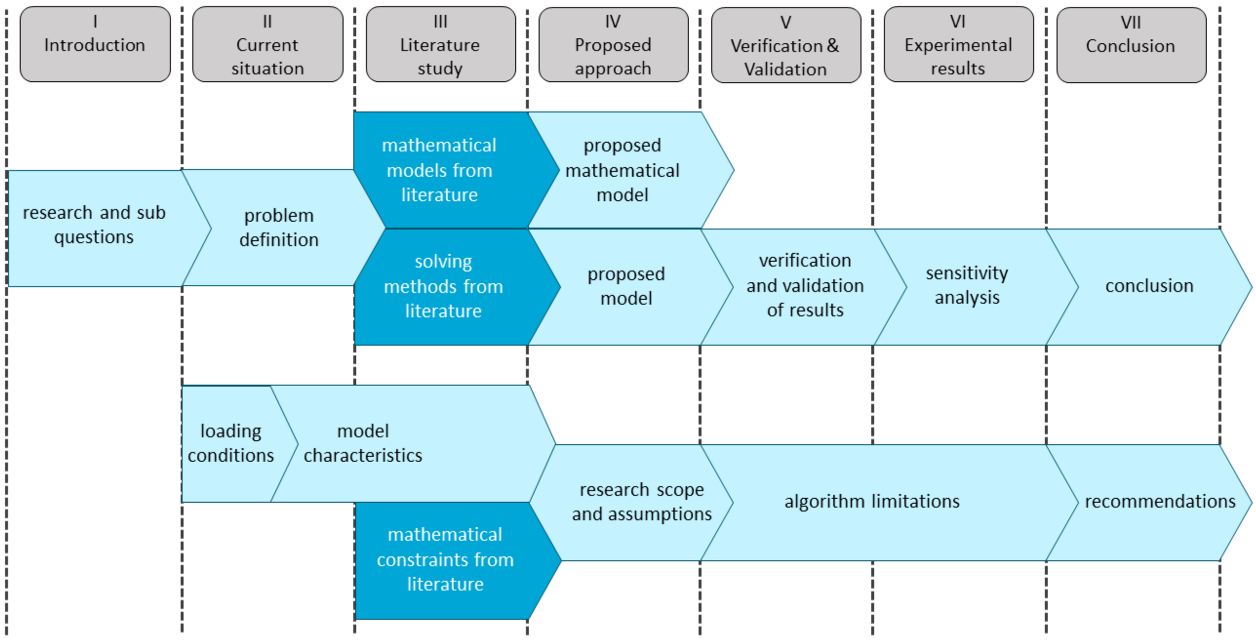


Figure 11: Report structure

3.1 Terms and definitions

Distributing and packing

Zhao et al. [10] use the term 'distributing' of items when speaking about the loading division process discussed in section 2. Packing is the collective name of arranging the items and is related to the loading sequence process from section 2. To keep things clear, in the remainder of this report, the terms loading division and loading sequence will be used.

Bins and containers

In the proposed typology by Wässcher et al. [9], the terms 'bin' and 'container' are used to describe the transport object where items can be packed into. For this research, the bins are defined as the trailers.

Items

An item of a problem refers to the products that will be placed onto the trailers. For this research, the items denote the assemblies, plates and other steel components. A mix of these terms will be used in this report to improve readability, but all refer to the same object.

Wässcher et al. [9] have described a large number of cutting and packing problems. They are grouped by their desired objective, dimensionality, characteristics of the items to be loaded, and characteristics of the loading objects (trailers). This information will be used to help define the scope of this literature review.

There are two possible objectives when it comes to cutting and packing problems. The first, input minimization, requires to fit all items in the lowest amount of space possible. For the second, output maximization, the amount of space is fixed, and the goal is to load as many items as possible. [9]
Naturally, the goal at ASK Romein is to transport all assemblies and components of a project, meaning input minimization is the objective.

For input minimization problems, there are two possible options to be considered when it comes to packing trailers: fixed trailer dimensions, or one or more variable dimensions.
The trailers used by ASK Romein have fixed width and height restrictions, but they can be elongated when necessary. This length extension could be considered as a variable dimension.

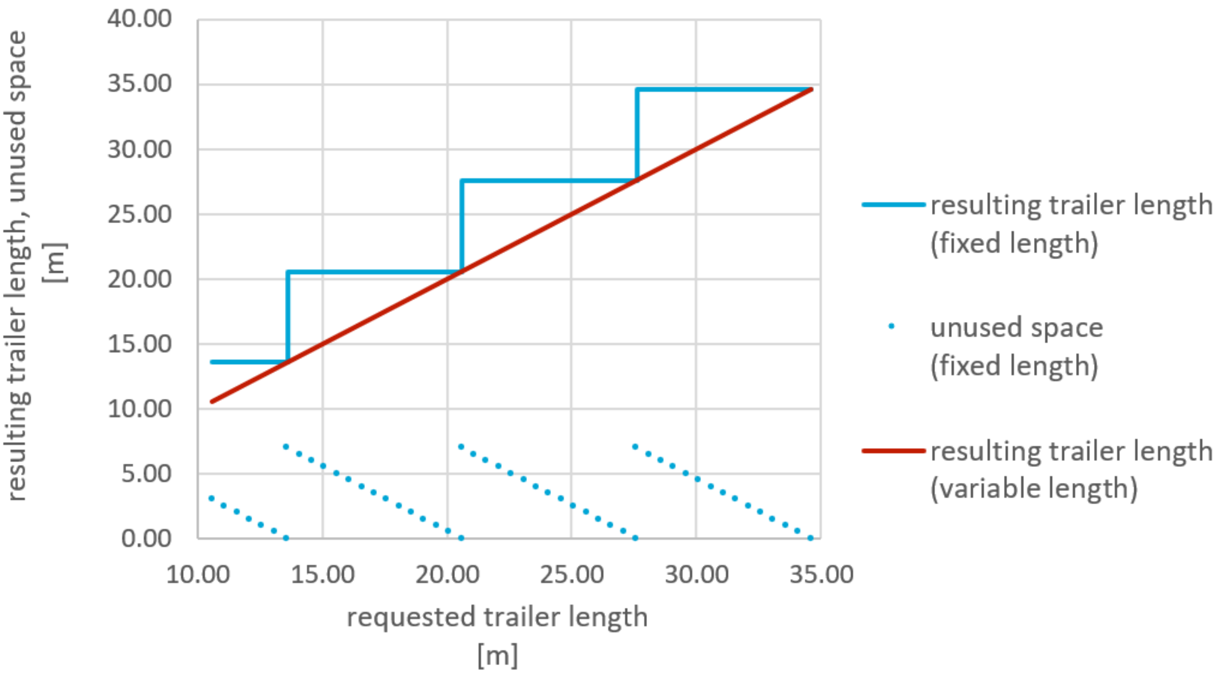
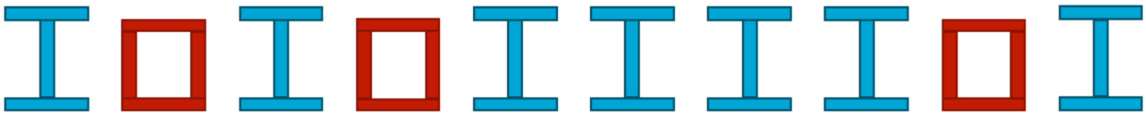


Figure 12: Downside fixed trailer lengths compared to variable lengths [1]

As figure 12 shows, using only a fixed number of trailer lengths will result in unused loading space. For example, if one of the largest item measures 15 m, the resulting trailer length would be 20.60 m, meaning 5.6 m of space would be unused. This space cannot be filled due to the fact that the load has to be indivisible (explained in section 2.3.3). Even though the use of trailers of fully variable lengths could be an interesting topic for the future, it is not applicable for the case of this research.
Fortunately, there is another variant available to define the problem. This involves the trailers to be able to extend, but not arbitrarily. There are fixed positions to which the trailers can be extended. In this perspective, the problem can be considered to contain a set of trailers with different lengths of fixed dimensions, thereby converting it to a fixed dimension problem.
Looking at the characteristics of the items and trailers, according to Wässcher et al. [9], both can be sub divided into three groups, as is shown in figure 13. Items and trailers can be identical, weakly heterogeneous and strongly heterogeneous. Identical means all objects are exactly the same. Weakly heterogeneous means there is a relatively low amount of different types of objects compared to the total amount of objects, while strongly heterogeneous means there are many different objects compared to the total number of objects.

(a) Identical items [1]



(b) Weakly heterogeneous items [1]

(c) Strongly heterogeneous items [1]

Figure 13: Overview of item characteristics

For this study, the trailers can be marked as weakly heterogeneous: they can only vary in a few lengths, and the number of trailers used is high. The items loaded on the trailer are strongly heterogeneous: there are a lot of different components and assemblies. They can vary in length, width and height, and can have steel plates attached to them in various places. Usually, only a few item copies exist within a project.

An overview of the aforementioned characteristics that describe the problem of this research:

- Objective: input minimization
- Dimensionality of the trailers: fixed dimensions, multiple fixed lengths available
- Characteristics of the trailers: weakly heterogeneous
- Characteristics of the items: strongly heterogeneous

As can be seen in figure 14, the corresponding approach, with strongly heterogeneous items and weakly heterogeneous trailers, is defined as the Multiple Bin Size Bin Packing Problem (MBSBPP). Literature concerning this specific cutting and packing problem type will be considered in the remainder of this section.

[1] Own work

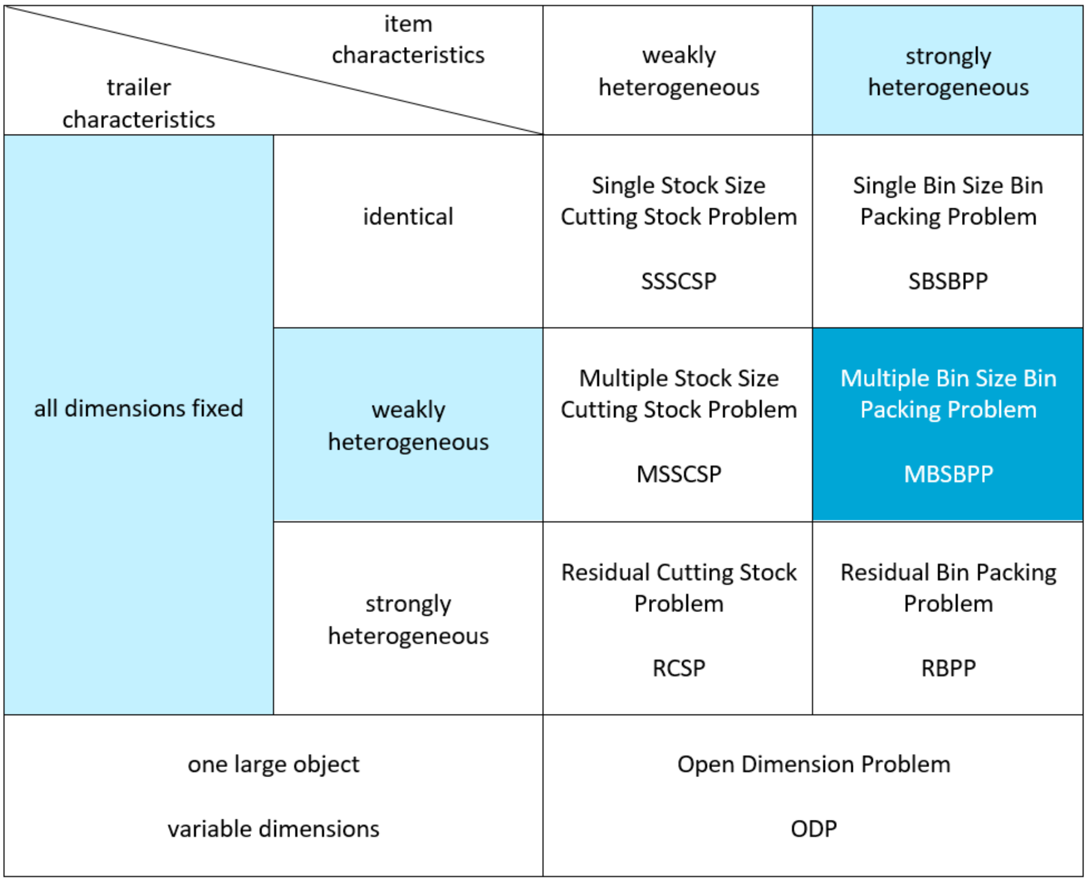


Figure 14: Schematic overview of cutting and packing problems [1]

When multiple trailers are involved in a packing problem, according to Eley [4], three strategies are distinguishable:

1. Sequential strategy: the trailers are filled one at a time. A new trailer is 'opened' when the other is full. A disadvantage of this approach is that the last trailers to be filled may have poor volume utilization, because larger, irregularly shaped items may end up last.
2. Pre-assignment strategy: first, the loading division takes place: all items are distributed over all the trailers. Next, a heuristic is applied for the loading sequence. If, ultimately, an item doesn't fit, it is placed into a different trailer in the next iteration.
3. Simultaneous strategy: loading division and loading sequence take place at the same time: the items are assigned and packed one by one, over multiple trailers. This requires more computation strength.

Referring to the state-of-the-art at ASK Romein (section 2), a manual version of the pre-assignment strategy is the best way to describe the current process. This version considers only one iteration: one for distributing the items over the trailers, and one for packing the items on the trailers.

A surprising result was achieved in Eley (2002) [5]: contrary to what may be expected, the sequential strategy outperformed the simultaneous strategy.

In the remainder of this section, existing literature regarding the MBSBPP with the pre-assignment strategy is discussed. First, information concerning the mathematical model is presented. Next, available solution methods are evaluated.

[1] Image obtained from Wässcher et al. (2007) [9].

3.2 Mathematical model

In this subsection, existing mathematical models will be discussed. The symbols used to describe the model parameters, decision variables and objective function may differ per author. For simplicity, throughout this report, the same type of variable or parameter, are expressed with consistent symbols, which may be in contrast to the symbols used by the authors themselves.

3.2.1 Parameters and decision variables

De Almeida and Figueiredo [1] and Jin et al. [6] solve a large-scale Three-dimensional Bin Packing Problem (3DBPP), which is a more general form of the MBSBPP. Regarding the mathematical model, De Almeida and Figueiredo and Jin et al. use the following notation for the model parameters:

$$
\begin{aligned}
&\mathrm{l}_i, \mathrm{w}_i, \mathrm{h}_i && = \text{item length, width, height}\\
&\mathrm{L}_i, \mathrm{W}_i, \mathrm{H}_i && = \text{trailer length, width, height}\\
&\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i && = \text{coordinates of the origin of an item}
\end{aligned}
$$

It should be noted that the origin of an item is located in the left-bottom-back corner.

To denote the orientation of an item, the following binary variables are defined by Jin et al. [6]:

$$
\begin{aligned}
\mathrm{lx}_i &= \begin{cases} 1 & \text{if the length of item i is parallel to the x-axis} \\ 0 & \text{otherwise} \end{cases}\\
\mathrm{ly}_i &= \begin{cases} 1 & \text{if the length of item i is parallel to the y-axis} \\ 0 & \text{otherwise} \end{cases}\\
\mathrm{wx}_i &= \begin{cases} 1 & \text{if the width of item i is parallel to the x-axis} \\ 0 & \text{otherwise} \end{cases}\\
\mathrm{wy}_i &= \begin{cases} 1 & \text{if the width of item i is parallel to the y-axis} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

It follows from the above equations that the upside of items is locked: rotation about the length and width direction is not allowed. One should note that because the upside is locked, the above four orientation decision variables could be reduced to just two, lx_i and ly_i, as figure 15 shows:

(a) lx = 1, wy = 1 (b) ly = 1, wx = 1

Figure 15: Decision variables orientations

To determine the relative position of one item to another, the following binary variables are used by all authors. Figure 16 shows a graphical representation of these decision variables.

$$a_{ij} = \begin{cases} 1 & \text{if item i is on the left side of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ij} = \begin{cases} 1 & \text{if item i is on the right side of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$c_{ij} = \begin{cases} 1 & \text{if item i is behind item j} \\ 0 & \text{otherwise} \end{cases}$$

$$d_{ij} = \begin{cases} 1 & \text{if item i is in front of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$e_{ij} = \begin{cases} 1 & \text{if item i is below item j} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{ij} = \begin{cases} 1 & \text{if item i is on top of item j} \\ 0 & \text{otherwise} \end{cases}$$

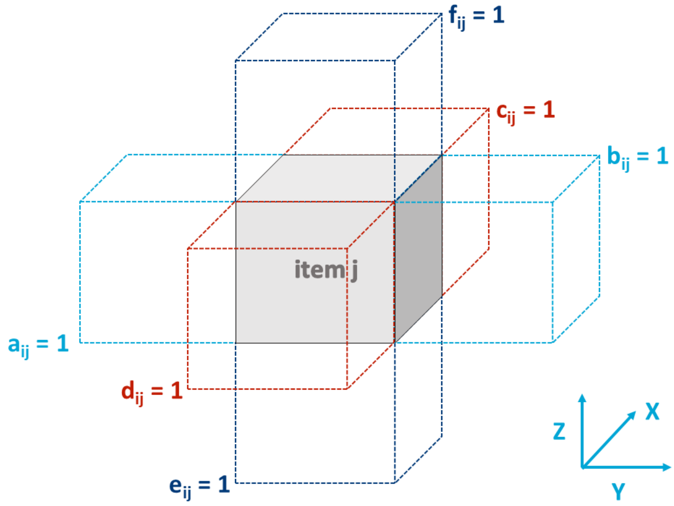


Figure 16: Orientation of item j with respect to item i, and corresponding decision variables

Additional binary variables are introduced to check if a trailer is in use and to allocate the items to the trailers. Here, de Almeida and Figuerido specifically mention both X_{it} and X_{jt}, to help denote when two items belong to the same trailer ($X_{it}\ X_{jt} = 1$).

$$X_{it} = \begin{cases} 1 & \text{if item i is packed in trailer t} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_t = \begin{cases} 1 & \text{if trailer t is included in the solution} \\ 0 & \text{otherwise} \end{cases}$$

3.2.2 Objective function

The MIP formulation of de Almeida and Figuerido [1] and Jin et al. [6] is shown in equation 1. The objective is to simply minimize the number of trailers used.

$$\text{Minimize} \sum_{j=1}^{m} Y_t \tag{1}$$

This objective function is extended by Ceschia and Schaerf [2]. They present an objective function that consists of four components:

$$f = w_1 f_{C1} + w_2 f_{C2} + w_3 f_{C3} + w_4 f_{C4} \quad (2)$$

In equation 2, f_{Ci} represents the cost function of component C_i, and w_i is the corresponding weight:

$C1$	Cost of items that could not be loaded
$C2$	Cost of using a trailer
$C3$	Empty space that can be used for loading unforeseen items
$C4$	Sum of number of destinations of the items present in each trailer

Instead of simply counting the number of trailers used, costs of items that could not be loaded are added, together with the option of reserving empty space for unforeseen items. Finally, C4 allows for a multi-drop constraint, where items on the same trailer can have different destinations, but this is out of the scope of this research.

3.2.3 Constraints

In this subsection, different constraint types are presented. According to Zhao et al. [10], there are three basic constraints applicable to all loading models concerning rectangular-like items. These constraints are also applied to all papers presented in this literature review:

- Items may only be placed in the loading space with their edges parallel to the edges of the loading space. For example, when loading a trailer, all items should be aligned with the edges of the trailer. This makes that only 90° rotations are possible. This constraint is embedded in the definition of orientation variables like lx_i.
- Items may not intersect each other. These constraints are called overlap constraints.
- All items can only be placed while they are entirely within the boundaries of the loading space. These constraints are discussed in the trailer bound constraints paragraph.

Besides these basic constraints, some other general constraints will be presented, after which additional constraints will be given that are applicable to specific cases.

Overlap constraints

An important type of constraints are overlap constraints: these prevent the overlap of items on the same trailer. An example from Jin et al [6] is shown in equation 3, where M is a very large number:

$$\mathrm{x}_i + \mathrm{l}_i \cdot \mathrm{lx}_i + \mathrm{w}_i \cdot (1 - \mathrm{lx}_i) \leq \mathrm{x}_j + (1 - \mathrm{a}_{ij}) \cdot M, \text{ for } i < j \quad (3)$$

Constraint 3 is applied if item i is on the left side of item k: in any other situation, the equation is satisfied instantly because M is a large number. The condition is met if the x-coordinate of item i, added up with the dimension of item i in x-direction (either the length l or the width w), is lower than the x-coordinate of item k, as equation 4 shows for a_{ij}= 1:

$$\mathrm{x}_i + \mathrm{l}_i \cdot \mathrm{lx}_i + \mathrm{w}_i \cdot (1 - \mathrm{lx}_i) \leq \mathrm{x}_j, \text{ for } i < j \quad (4)$$

If, for example, the orientation of the particular item states that $\mathrm{lx}_i = 0$, equation 4 can be further reduced to equation 5:

$$\mathrm{x}_i + \mathrm{w}_i \leq \mathrm{x}_j, \text{ for } i < j \tag{5}$$

The overlap constraints from de Almeida and Figuerido [1] show similarities with Jin et al in their basic principle, but are defined differently:

$$\mathrm{a}_{ij}(\mathrm{x}_i + \mathrm{w}_i) + \mathrm{b}_{ij}(\mathrm{x}_j + \mathrm{w}_j) \leq \mathrm{a}_{ij}x_j + \mathrm{b}_{ij}\mathrm{x}_i \tag{6}$$

In equation 6, both the left and right side option are incorporated in the same equation, meaning only three constraints are required instead of six. Furthermore, the use of a large number M is no longer needed.
If for example item i is placed to the left of j (and $\mathrm{a}_{ij} = 1$, $\mathrm{b}_{ij} = 0$), the above equation is reduced to equation 7, which is equal to equation 5:

$$\mathrm{x}_i + \mathrm{w}_i \leq \mathrm{x}_j \tag{7}$$

One should note that in the paper from de Almeida and Figuerido, only one orientation was considered, which explains the absence of decision variable lx_i.

Trailer bound constraints

The next constraints make sure that an item is packed within the bounds of the trailer. Jin et al. [6] use equations 8 - 10:

$$\mathrm{x}_i + \mathrm{l}_i \cdot \mathrm{lx}_i + \mathrm{w}_i \cdot \mathrm{wx}_i \leq \mathrm{L}_t + (1 - \mathrm{X}_{it}) \cdot M \tag{8}$$

$$\mathrm{y}_i + \mathrm{w}_i \cdot \mathrm{wy}_i + \mathrm{l}_i \cdot \mathrm{ly}_i \leq \mathrm{W}_t + (1 - \mathrm{X}_{it}) \cdot M \tag{9}$$

$$\mathrm{z}_i + \mathrm{h}_i \leq \mathrm{Z}_t + (1 - \mathrm{X}_{it}) \cdot M \tag{10}$$

Here, constraint 10 is a simplification of equations 8 and 9, due to the fact that rotation is only allowed around the z-axis and thus the available orientations have no effect on the height direction. Because de Almeida and Figuerido do not consider any rotation, constraints 8 - 10, have been reduced to constraints 11 - 13:

$$\mathrm{X}_{it}(\mathrm{x}_i + \mathrm{w}_i) \leq \mathrm{W}_i \tag{11}$$

$$\mathrm{X}_{it}(\mathrm{y}_i + \mathrm{l}_i) \leq \mathrm{L}_i \tag{12}$$

$$\mathrm{X}_{it}(\mathrm{z}_i + \mathrm{h}_i) \leq \mathrm{H}_i \tag{13}$$

General constraints

The next group of constraints defines the general rules of a Mixed-Integer Programming model. For example, Jin et al [6] define equation 14, which ensures that overlap constraints like constraint 3 are only applied to items on the same trailer:

$$\mathrm{a}_{ij} + \mathrm{b}_{ij} + \mathrm{c}_{ij} + \mathrm{d}_{ij} + \mathrm{e}_{ij} + \mathrm{f}_{ij} \geq \mathrm{X}_{it} + \mathrm{X}_{jt} - 1, \text{ for } i < j \tag{14}$$

If item i and j are on the same trailer, trailer t, the right hand side becomes 1 for that particular trailer. Else, the right hand side becomes 0 or -1, and the constraint can be considered inactive.
For this constraint, de Almeida and Figuerido take a slightly different approach, as is shown in equations 15 and 16:

$$\mathrm{a}_{ij} + \mathrm{b}_{ij} + \mathrm{c}_{ij} + \mathrm{d}_{ij} + \mathrm{e}_{ij} + \mathrm{f}_{ij} \geq \mathrm{X}_{it}\mathrm{X}_{jt} \tag{15}$$

$$\mathrm{a}_{ij} + \mathrm{b}_{ij} + \mathrm{c}_{ij} + \mathrm{d}_{ij} + \mathrm{e}_{ij} + \mathrm{f}_{ij} \leq 3\mathrm{X}_{it}\mathrm{X}_{jt} \tag{16}$$

The result of these constraints is that if two items are on the same trailer, then the left hand side of both equations, should be ≥ 1 and ≤ 3. If both items are not on the same trailer, the left hand side should equal 0. Equations 15 and 16 therefore are only active when both item i and j are on trailer t: in this case, $\mathrm{X}_{it}\mathrm{X}_{jt} = 1$ for trailer t, and 0 otherwise.

Finally, equations 17 and 18 from Jin et al. guarantee that each item is placed on one trailer and one trailer only (eq. 17), and that a trailer is flagged as 'in use' ($\mathrm{Y}_t = 1$) when an item is placed onto it (eq. 18):

$$\sum_{t=1}^{T} \mathrm{X}_{it} = 1 \tag{17}$$

$$\sum_{t=1}^{T} \mathrm{X}_{it} \leq M \cdot \mathrm{Y}_t \tag{18}$$

Additional constraints

In this paragraph, additional constraints are presented that are designed for cases with specific demands.

- Weight capacity and weight distribution: weight on the axles is a critical measure for loading a vehicle. The total weight can also be the binding constraint instead of volume. This asks for a specific constraint that limits the loads.
- Load support: to maintain item stability, three different options are available [10]:
  1. Full support: the entire base of an item fits on the top of the item beneath
  2. Percentage of overhang: a certain percentage of overhang is allowed on each side, for example 80%.
  3. Centre of gravity is supported: Overhang is allowed, as long as the centre of gravity is supported.

  According to Zhao et al. [10], even though load support is an important aspect in real-life applications, it is rarely considered in the literature.

- Multi-drop: if a trailer has to be unloaded in a number of different locations, a so-called multi-drop constraint can be introduced. This however is not the case for this research.
- Separation of boxes: the separation of boxes denotes a situation where items of two different types must be placed on separate trailers. This is the case for different conservation systems and different building phases for example.
- Complete shipment: when all items from a certain type have to be transported on the same trailer, the complete shipment constraint applies. Nevertheless, this is not required for this research.

A total weight capacity constraint has a very straightforward form: the sum of item weights m_i placed on the trailer has to be less or equal than the maximum allowable weight $\mathrm{m}_{\text{legislation}}$:

$$\sum_{i}^{\mathrm{I}} \mathrm{m}_i \leq \mathrm{m}_{\text{legislation}} \tag{19}$$

The weight distribution constraints may be a bit more tricky, as they require the combined center of mass in the length direction of a trailer. Using simple physics, the moment around one of the axles can be taken to obtain the load on the other axle, after which the load on the first axle is taken from the difference of the item weight and the second axle.
Unfortunately, to the knowledge of the author of this report, there are no mathematical model constraints available in the current literature. A separation of boxes constraint can be of the form presented in equation 20:

$$\mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{type}_i = \mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{type}_j \tag{20}$$

Here, if and only if item i and j are placed on the same trailer, Y_{it} $\mathrm{Y}_{jt} = 1$ and the equation reduces to equation 21:

$$\mathrm{type}_i = \mathrm{type}_j \tag{21}$$

And this induces that both items should be of the same type when placed on the same trailer.

3.3 Solution methods: placement and improvement heuristics

Zhao et al. [10] define the placement heuristic as construction of a solution on trailer level. In this report, the placement heuristic is responsible for the loading sequence process.
The solution generated by the placement heuristic can be improved by a so-called improvement heuristic: a way to find better solutions, usually within the neighborhood of the existing solution. The improvement heuristic can be considered as the algorithm version of the loading division process.

3.3.1 Loading division methods

In this subsection, the available improvement heuristics, responsible for the loading division, are evaluated. There are three methods: tabu search (TS), guided local search (GLS), and adaptive large neighborhood search (ALNS).
The tabu search is one of the most commonly known heuristics, and also a popular choice for many cutting and packing problems. It is implemented in the 3D packing problems by Jin et al. [6] and Crainic et al. [3]. The main aspect of the tabu search is that it keeps a tabu list of previous moves to guide the search.
Guided local search also makes use of memory, but instead of keeping a list, the objective function is readjusted. A typical objective function for GLS contains penalty terms. When investigating a

promising region of a local space, if a local minimum is reached, the penalty terms are (slightly) adjusted so that the cost function changes and the local minimum can be escaped.
The ALNS heuristic is the least known method of the three. This method uses destroy and repair operators to destroy an existing solution and then repair it by swapping items. It is beneficial because multiple destroy and repair operators can be added, and therefore the ALNS heuristic can apply multiple strategies to build a new solution, making it flexible to different cases.

The table in figure 17 shows the obtained scores for all three methods. First, the methods are scored according to their computation time. The computation time is an important factor in model design. Since multiple iterations will be required, a large computation time can easily turn into excessive waiting times. This is not acceptable, and therefore this criterion receives the highest possible weight (3).
Looking at the three methods, both TS and GLS need a large amount of memory to 'remember' previous solutions, meaning they probably take longer to process and thus receive a low score. ALNS on the other hand only has to remember the current best solution, and can adapt itself during execution to reach an optimal solution sooner. The ALNS heuristic is not known in the field of cutting and packing problems, but for vehicle routing problems it is known to outperform the computation times of other heuristics such as TS and GLS. For example, Zulj et al. report that for their vehicle routing problem, ALNS is approximately 2.5 - 3 times as fast as TS [11], which is significant and thus makes it safe to assume that ALNS will also be faster for cutting and packing problems. As a result, TS and GLS will receive a low score (1), and ALNS the highest score (3).

	tabu search (TS)		guided local search (GLS)		adaptive large neighborhood search (ALNS)	
	score	total	score	total	score	total
computation time (3)	1	3	1	3	3	9
complexity (1)	3	3	2	2	1	1
flexibility (2)	1	2	1	2	3	6
reliability (3)	3	9	3	9	1	3
total	**8**	**17**	**7**	**16**	**8**	**19**

Figure 17: Available loading division methods. Scores range from 1 (bad) to 3 (good), weights range from 1 (less important) to 3 (very important).

A complex model requires a lot of programming effort and needs a large documentation, which may become hard to understand as time progresses. This of course should be considered, but on the other hand, it has no negative effect on the outcome of the model, which is why a low weight (1) is given to this criterion.
Both TS and GLS use a single heuristic to obtain a solution, which reduces the complexity of the model. Tabu search is a simple branch-and-bound heuristic, guided local search has additional functions, meaning the scores are a 3 and 2, for TS and GLS, respectively.
The ALNS heuristic is made up of multiple sub heuristics that work together. This requires more lines of code and makes the ALNS heuristic the most complex of the three, so that it receives the minimum score (1).

The flexibility of a method is directly related to the ALNS heuristic. As mentioned before, this heuristic consists of multiple sub heuristics. These heuristics can compete against each other, so that the model can calibrate itself as the execution is ongoing. A flexible model is able to reach good solutions earlier and help the model overcome local minima. This a useful trait, and therefore this criterion receives weight 2. Naturally, since ALNS is the definition of this criterion, it gets the highest score (3), while the other methods, which only use a single heuristic, get the lowest score (1).

Finally, the reliability of a method is defined as a combination of its robustness and whether a method is common in the field of cutting and packing problems. If the latter is true, the chance of success may be more guaranteed. This criterion receives the maximum weight, because it is important to have a robust solution.
TS and GLS are both methods which are commonly used in benchmark literature regarding cutting and packing problems, and specifically MBSBPP. Therefore, they get a maximum score (3). ALNS is, as mentioned before, not yet known as a solution to any cutting and packing problem, and therefore the outcome and reliability of this method can only be estimated, meaning it receives a low score (1).

Adding up the weighted scores of all criteria, the method which receives the highest score is the adaptive large neighborhood search (ALNS) heuristic, with 19 points. Nevertheless, the other methods follow closely, which means that an analysis of the results is required. Looking at the ALNS method, its strong points are, as has been discussed, its computation time and flexibility. On the other hand, this heuristic receives low scores for complexity and reliability, which are the strong points of both other methods.
Complexity here is by far the least important criterion, which is also why it received the lowest weight. Therefore, the fact that ALNS does not score well on this aspect is not very important. Looking at the reliability, the main reason that the ALNS heuristic does not score well is the fact that it has not a known method in the field of cutting and packing problems. However, it is not a new heuristic, as it has proven itself in other fields such as vehicle routing.
As a result, it can be concluded that even though the ALNS method has its weak points, its top score can be justified, and this method is the best option for a loading division heuristic.

3.3.2 Loading sequence methods

The available placement heuristics or loading sequence methods are discussed in this section. They are the orientation Mixed Integer Programming (orientation MIP), 3D raster Mixed Integer Programming (3D raster MIP), the sub volume heuristic and the irregular shape heuristic.

The orientation MIP is a regular mathematical formulation that can be solved by a software program. This method requires a solid mathematical formulation consisting of parameters, decision variables, an objective function and constraints. The software program solves the problem.
The mathematical formulation is also the basis of the 3D raster MIP, but here the solver uses a 3D raster or pixel representation of the trailer to place the items. For example, if the chosen precision is 1 mm, a pixel is generated on each mm, that can have either a value 1 (occupied) or 0 (vacant). If an item is placed on a trailer, its exact shape is generated and for each pixel it occupies, it is checked if no other item is already there.
The sub volume heuristic has a completely different approach: when an item is placed on the trailer, three new sub volumes are created on the item side, front and top. New items can be placed in these spaces. Previous sub volumes may be merged to create larger spaces.
Finally, the irregular shape heuristic is designed especially for items with complex geometries. The method first generates the entire shape of an item by bounding it. Next, the optimal placement is calculated. Due to the complex formulas for the item bounding this method has only been used for two-dimensional problems.

The table in figure 18 shows the different scores for the four methods. Similar to the improvement heuristics, the methods are scored according to their computation time. and again this criterion receives the highest possible weight (3).
Both MIP methods have to consider all possible positions (in mm) for all items to be placed on a trailer, even for non-feasible solutions. Furthermore, the orientation MIP needs to check all constraints for these positions, whereas the 3D raster MIP has to check every point in the raster for every new position. Especially for the latter, an excessive computation time can be expected, as for a trailer of standard dimensions, with a precision in mm, there are 136 billion coordinates. It is easy to conclude that the scale of this research simply is too large for a MIP approach, and the same is concluded for many other practical examples in the literature [6], [10]. Because the 3D raster MIP is significantly worse than the orientation MIP, the resulting scores are 1 and 2, respectively.
The computation time of the irregular shape heuristic is probably comparable to the orientation MIP. This can be explained due to the fact that the exact bounding of each item has to be generated. For instances with a large amount of items this requires a lot of computation strength, meaning the score for computation time is 2.
Finally, the sub volume heuristic takes a vastly different approach: a solution is built one item at a time, placing new items in the spaces (sub volumes) created by the previous items. This method only allows for feasible solutions, meaning the possible solution space is drastically decreased, as is the computation time. Therefore, a maximum score for the computation time is given (3).

	orientation MIP		3D raster MIP		sub volume heuristic		irregular shape heuristic	
	score	total	score	total	score	total	score	total
computation time (3)	2	6	1	3	3	9	2	6
complexity (1)	3	3	3	3	2	2	1	1
feasibility (3) - 3D - item support - inter-flange placement	1 yes no no	3	3 yes yes yes	9	2 yes yes no	3	1 no no yes	3
financial (1)	1	1	3	3	3	3	3	3
total	**7**	**13**	**10**	**18**	**10**	**17**	**7**	**13**

Figure 18: Available loading sequence methods. Scores range from 1 (bad) to 3 (good), weights range from 1 (less important) to 3 (very important).

The next criterion again is complexity: the required programming effort, with weight 1. Here, both MIP methods receive the highest scores. This is not a coincidence: overall, less complex methods require more computation strength, and more complex methods have a lower computation time. This is because for the MIP methods the actual solving is executed by the software: only input in the form of parameters, decision variables, an objective function and constraints is needed. For more complex models, the solving is implemented in the method itself.
Using this information, a high score is given to both MIP approaches (3). Both the sub volume heuristic and irregular shape heuristic require the implementation of solving steps, and thus are more complex. The irregular shape heuristic can be considered even more complex due to the item bounding function, which is hard to achieve. This results in a medium score for the sub volume heuristic (2), and a low score for the irregular shape heuristic (1).

The placement heuristic needed for the loading sequence has to account for specific characteristics. These characteristics are key to obtain a feasible solution:

- 3D: many methods found in the literature consider only one-dimensional or two-dimensional packing problems. This research requires three dimensions.
- Item support: a constraint that is often overlooked in benchmark literature is the item support constraint. This constraint is essential for truck transport and should be incorporated in the placement heuristic.
- Inter-flange placement: this characteristic is unique for this problem. As many components for ASK Romein are H-shaped profiles with plates welded to them, it is a common practice to place items next to each other with plates from one item between the flanges of another item. However, this is not easy to implement in a model, and some methods are incapable of capturing this feature.

The feasibility criterion is the number of key characteristics each method is able to incorporate: the higher the score, the more characteristics a method is able to incorporate.

The final criterion is financial: the orientation MIP requires a licensed software program, which costs money. This has no effect on the outcome of the model, but it should still be considered, which is why the criterion receives a weight of 1. Since the orientation MIP is the only method known to this author that requires licensed software, it is the only method that receives a minimum score (1).

Looking at the resulting scores, one could conclude that the 3D raster MIP method is the best option, as it has obtained the highest score. When looking more in-depth, it also appears that this method is the only heuristic that is able to account for all feasibility characteristics, and thus the only method that is capable to correctly represent the problem of this research: all other methods lack at least one characteristic.
Nevertheless, as was mentioned before, the 3D raster MIP method has one major flaw: its computation time is unacceptable.
It seems that there are two options:

1. Choose the 3D raster MIP method, which is a feasible method with an unacceptable computation time
2. Choose any of the other methods, which has an acceptable computation time, but is unable to capture all characteristics of the problem

Since neither of the options results in a desired solution, it should be concluded that there is a literature gap, and therefore, a new method is presented in the next section, section 4.

When comparing tables 17 and 18, different criteria are used. For table 17, the criteria feasibility and financial are missing. The absence of financial can easily be explained: none of the listed methods requires the use of licensed software. Feasibility is left out, simply because it is related to whether a method is able to account for all characteristics of the problem, and this is the case for each of the methods. This would result in identical scores for all methods, meaning adding this criterion is useless.
Subsequently, in the table in figure 18, the criteria flexibility and reliability are omitted. Flexibility is related to whether a method can use different strategies during execution. This can or cannot be a built-in feature for all loading sequence methods, but is unrelated to the method type, which is why this criterion is not included.
As the result has already shown, none of the methods presented in table 18 have a known application related to the problem of this research. Therefore, the reliability cannot be evaluated for any of the methods, which is why this criterion is left out.

3.4 Discussion

In this section, the scope of the literature review has been evaluated by looking at the benchmark typology regarding cutting and packing problems. The problem of this research has been defined to have an input minimization objective, fixed dimensions, weakly heterogeneous trailers and strongly heterogeneous items, making it a Multiple Bin Size Bin Packing Problem (MBSBPP). The packing strategy applicable to this research is the pre-assignment strategy, where first all items are distributed over the trailers (loading division), and next the items are placed on the trailers (loading sequence).
Next, several examples of mathematical model features have been given. These examples can be used to define the mathematical model of this research in section 4.
The sub question of section 3:

2. What are the existing solutions of 3D space maximization problems?

In the latter part of this section, the available solution methods have been discussed and evaluated: regarding the loading division methods, the ALNS heuristic appears to be the most promising method, due to its flexibility and fast computation time. When it comes to the loading sequence methods, none of the available methods appears to truly fit the problem of this research, meaning a new heuristic is required to fit the literature gap.
This new heuristic will be presented in the next section, together with the mathematical model and a description of the ALNS heuristic.

4 Proposed approach

With the results from section 3 in mind, in this section, the proposed methods are presented. As figure 19 shows, this section starts with defining the research scope. Next, a mathematical formulation of the model is provided, after which the proposed model is discussed.

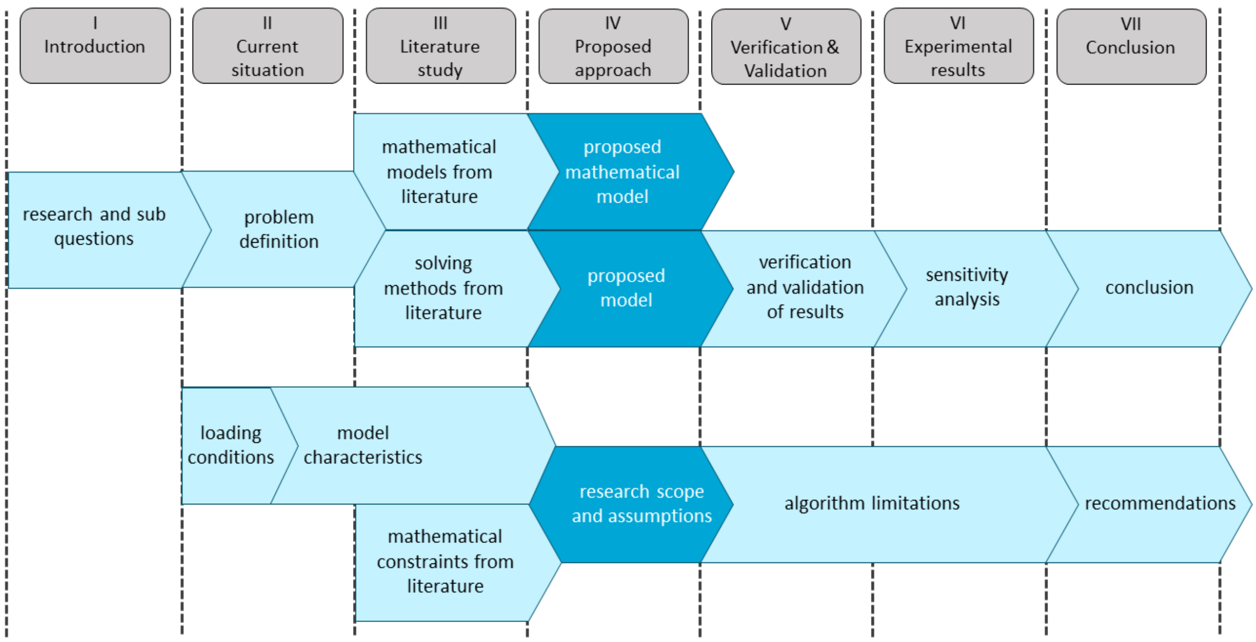


Figure 19: Report structure

4.1 Research scope

In this subsection, the assumptions and simplifications related to the characteristics from section 2 are given, to identify the research scope. Table 3 shows the constraints and their corresponding assumptions:

1. Maximum weight: for now, the legislation regarding the maximum allowed total weight will be checked for three countries: The Netherlands, Belgium and Germany. If a transport route contains one or more of these countries, the strictest legislation is taken. Legislation for other countries can easily be added later.
2. Axle and fifth wheel weight: similar to the maximum weight, the country in a route with the strictest legislation is used to define the constraint for the axle and fifth wheel loads.
3. Trailer length: in total, there are four possible trailer lengths to be considered. There is an exception: some trailers carry a crane, which removes 1.10 m of the available loading space (in brackets):
   - Standard: 13.60 m (12.50 m)
   - Single extension: 20.60 m (19.50 m)
   - Double extension: 27.60 m (26.50 m)
   - Triple extension: 34.60 m
4. Trailer width and height: the trailer width and height are assumed to be fixed. The standard dimensions according to European legislation are 2.50 m and 4.00 m, respectively. These are the dimensions for which no additional permit is required, and are taken as the standard dimensions for this model.

Characteristic:	Description:
trailers	
1. Maximum weight	legislation NL, B, DE
2. Axle and fifth wheel loads	legislation NL, B, DE
3. Length	4 (+3) options
4. Width and height	fixed width and height (legislation)
items	
5. Flatrack and trailer gap	assume fixed flatrack position
6. Item movement	securing of items against headboard and by height
7. Item stability	items on larger other items
8. Item orientation	2 orientations
processes	
9. Conservation finish	1 conservation system per trailer
10. Erection sequence	1 building phase per trailer
11. Loading and unloading	both vertical and horizontal (un)loading allowed

Table 3: Model constraints

5. Flatrack loading: all items are assumed to be loaded onto a standard flatrack used by ASK Romein. On a standard trailer size, the flatrack is placed against the headboard. In case the trailer has one or more extensions, the position of the flatrack moves forward to cover for the gap created in the middle of the trailer.

6. Item movement: to prevent the movement of items during transportation, all items have to be secured. To achieve this, all items are placed against the trailer headboard or in contact with other items placed against the headboard. Furthermore, items of similar height are placed in the same layer, so that upper layers exert pressure on them and hold them into place.

7. Item stability: for different layers on top of each other, in length direction, only full support is allowed. This means that a layer of items should be placed on another layer without overhang.

8. Item orientation: only two orientations are considered. Most items would not fit within standard trailer dimensions in other orientations, and fixed orientations also reduce the solution space.

9. Conservation finish: since the location where the assemblies are shipped to is directly dependent on the conservation system, all finishes are considered. The problem is simplified by allowing only one coating type per trailer, allowing the use of the separation of boxes constraint from section 3.

10. Erection sequence: the allocation of items depends on their building phase: only items from the same building phase can be placed onto the same trailer.

11. Loading and unloading: to prevent that loaded assemblies cannot be unloaded on site, for example because they are placed too close to one another to attach an unloading cable, the different items are separated by separation wood. These wooden plates are assumed to have a fixed thickness of 50 mm, and this distance is added to the overlap constraints. For horizontal unloading (via forklifts), wooden beams of 100 mm thickness are assumed to be placed between item layers.

4.2 Mathematical formulation

In this subsection, the mathematical model for this research is presented, starting with the indices. Next, the decision variables, parameters, objective function and constraints are given. The decision variables of a mathematical model change throughout the iterations. An example is the value for the x-position of an item, that depends on where the item is placed in a certain iteration. The parameters on the other hand are values that are fixed throughout the model, such as the length of an item, that does not change in between iterations.

4.2.1 Indices

i	items	0, ..., I
j	items	0, ..., J
k	sub items	0, ..., K_i
t	trailers	0, ..., T

with

I	number of items
J	number of items
K_i	number of sub items of item i
T	number of available trailers

4.2.2 Decision variables

The first decision variables are similar to Jin et al. [6] and determine the (relative) orientations and coordinates of the items and sub items. Note that the main item has index k = 0. Furthermore, $lx1_i$ and $lx2_i$ are an extension of lx_i for non-symmetrical items (figure 20).

x_{ik}	x-coordinate of origin of sub item k of item i
y_{ik}	y-coordinate of origin of sub item k of item i
z_{ik}	z-coordinate of origin of sub item k of item i

$$lx1_i = \begin{cases} 1 & \text{if the longest dimension of item i is in the trailer length direction} \\ & \text{and the item origin is towards the headboard} \\ 0 & \text{otherwise} \end{cases}$$

$$lx2_i = \begin{cases} 1 & \text{if the longest dimension of item i is in the trailer length direction} \\ & \text{and the item origin is away from the headboard} \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij} = \begin{cases} 1 & \text{if item i is on the left side of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ij} = \begin{cases} 1 & \text{if item i is on the right side of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$c_{ij} = \begin{cases} 1 & \text{if item i is behind item j} \\ 0 & \text{otherwise} \end{cases}$$

$$d_{ij} = \begin{cases} 1 & \text{if item i is in front of item j} \\ 0 & \text{otherwise} \end{cases}$$

$$e_{ij} = \begin{cases} 1 & \text{if item i is below item j} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{ij} = \begin{cases} 1 & \text{if item i is on top of item j} \\ 0 & \text{otherwise} \end{cases}$$

(a) lx1 = 1 (b) lx2 = 1

Figure 20: Decision variables orientations

Next are two binary decision variables that state if an item is packed and if a certain trailer is used:

$$X_{it} = \begin{cases} 1 & \text{if item i is packed in trailer t} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{t} = \begin{cases} 1 & \text{if trailer t is included in the solution} \\ 0 & \text{otherwise} \end{cases}$$

4.2.3 Parameters

l_{ik}	length of sub item k of item i $0 \leq i \leq I$, $0 \leq k \leq K$
w_{ik}	width of sub item k of item i
h_{ik}	height of sub item k of item i
xk_{ik}	relative distance between the x-coordinates of the origins of main item i and its sub item k
yk_{ik}	relative distance between the y-coordinates of the origins of main item i and its sub item k
zk_{ik}	relative distance between the z-coordinates of the origins of main item i and its sub item k
cmx_i	x-position center of mass of item i (including sub items)
m_i	mass of item i (including sub items)
L_t	trailer length
W_t	trailer width
H_t	trailer height
w_ver	vertical stoppage wood thickness
w_hor	horizontal stoppage wood thickness

Bx_t	distance between trailer front and bogies of trailer t
Fx_t	distance between trailer front and fifth wheel of trailer t
M_t	empty mass of trailer t
Mflatrack	flatrack mass
$\mathrm{mb}_{\mathrm{legislation}}$	allowable rear bogie axle load
$\mathrm{mf}_{\mathrm{legislation}}$	allowable fifth wheel load
$\mathrm{m}_{\mathrm{legislation}}$	allowable item load
csrv_i	conservation system of item i
bphs_i	building phase of item i
M	large number

4.2.4 Constraints

Naturally, following the above parameters and the decision variables for the different orientations, equations 22a - c, used to obtain the origin coordinates of the sub items, hold for all k. These overlap constraints are related to the model characteristics regarding the item orientation (8) and loading and unloading (11) from table 3.

$$\mathrm{a}_{ij} * \left(\mathrm{y}_{ik} + \mathrm{lx1}_i * \mathrm{w}_{ik} + \mathrm{w\_ver}\right) - \mathrm{b}_{ij} * \left(\mathrm{y}_{ik} + \mathrm{lx2}_i * \mathrm{w}_{ik}\right) \leq \mathrm{a}_{ij} * \left(\mathrm{y}_{jl} - \mathrm{lx2}_j * \mathrm{w}_{jl}\right) \tag{22a}$$
$$-\mathrm{b}_{ij} * \left(\mathrm{y}_{jl} + \mathrm{lx1}_j * \mathrm{w}_{jl} + \mathrm{w\_ver}\right)$$

$$\mathrm{d}_{ij} * \left(\mathrm{x}_{ik} + \mathrm{lx1}_i * \mathrm{l}_{ik} + \mathrm{w\_ver}\right) - \mathrm{c}_{ij} * \left(\mathrm{x}_{ik} - \mathrm{lx2}_i * \mathrm{l}_{ik}\right) \leq \mathrm{d}_{ij} * \left(\mathrm{x}_{jl} - \mathrm{lx2}_j * \mathrm{l}_{jl}\right) \tag{22b}$$
$$-\mathrm{c}_{ij} * \left(\mathrm{x}_{jl} + \mathrm{lx1}_j * \mathrm{l}_{jl} + \mathrm{w\_ver}\right)$$

$$\mathrm{e}_{ij} * \left(\mathrm{z}_{ik} + \mathrm{h}_{ik} + \mathrm{w\_hor}\right) - \mathrm{f}_{ij} * \mathrm{z}_{ik} \leq \mathrm{e}_{ij} * \mathrm{z}_{jl} - \mathrm{f}_{ij} * \left(\mathrm{z}_{jl} + \mathrm{h}_{jl} + \mathrm{w\_hor}\right) \tag{22c}$$

Figure 22a shows an example for the y-direction overlap constraint.

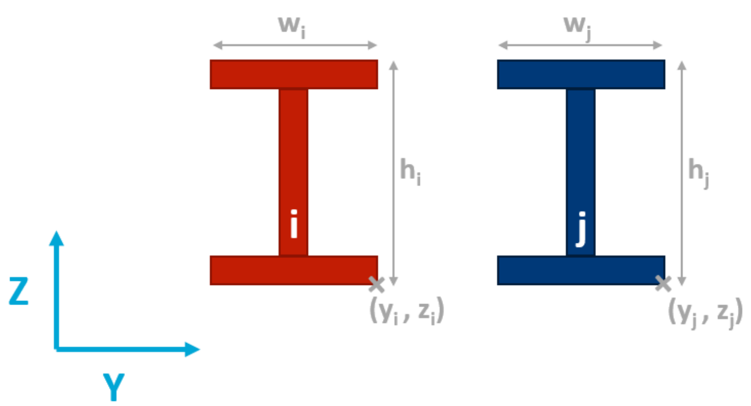


Figure 21: Example constraint 22a [1]

[1]Own work.

With orientations $lx2_i = 1$ and $lx2_j = 1$ and binary variables $a2_ij = 1$ and $b_ij = 0$, equation 22a is reduced to:

$$y_{ik} + w\_ver \leq y_{jl} - w_{jl} \tag{23}$$

Which simply means that the y-coordinate of item i plus one piece of stoppage wood (to allow for loading and unloading) should be smaller than the y-coordinate of item j minus the width of item j, as figure 22 shows:

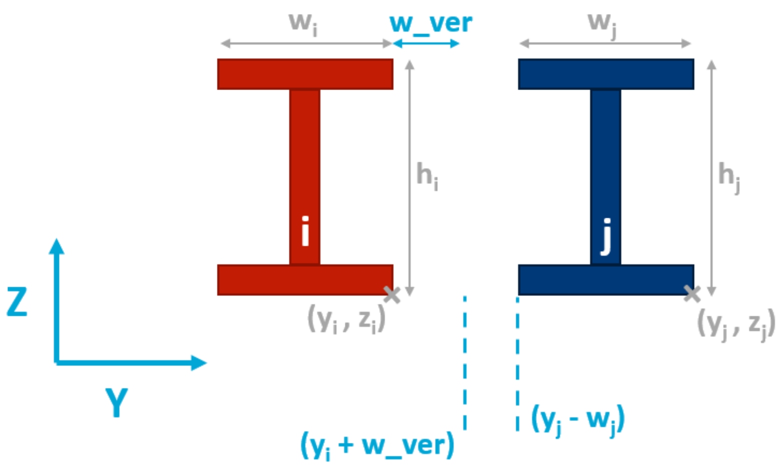


Figure 22: Graphical representation equation 23 [1]

Equations 24a - c connect the sub items ($k > 0$) to the main items ($k = 0$). A graphical representation of these constraints is shown in figure 23.

$$y_{ik} = y_{i0} + \left(lx1_i - lx2_i\right) * yk_{ik} \tag{24a}$$

$$x_{ik} = x_{i0} + \left(lx1_i - lx2_i\right) * xk_{ik} \tag{24b}$$

$$z_{ik} = x_{i0} + zk_{ik} \tag{24c}$$

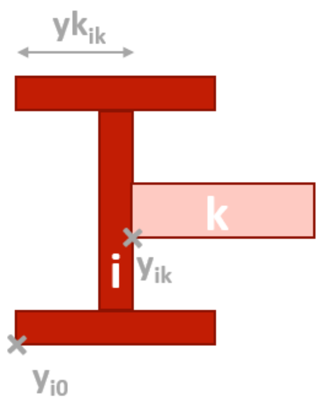


Figure 23: Graphical representation equation 24a [1]

[1]Own work.

Equation pairs 25a - b, 26a - b, and 27a - b make sure that all items are placed completely within trailer bounds:

$$0 \leq \mathrm{x}_{ik} - \mathrm{lx2}_i * \mathrm{l}_{ik} \tag{25a}$$

$$\mathrm{x}_{ik} + \mathrm{lx1}_i * \mathrm{l}_{ik} \leq \mathrm{L}_t \tag{25b}$$

$$0 \leq \mathrm{y}_{ik} - \mathrm{lx2}_i * \mathrm{w}_{ik} \tag{26a}$$

$$\mathrm{y}_{ik} + \mathrm{lx1}_i * \mathrm{w}_{ik} \leq \mathrm{W}_t \tag{26b}$$

$$0 \leq \mathrm{z}_{ik} \tag{27a}$$

$$\mathrm{z}_{ik} + \mathrm{h}_{ik} \leq \mathrm{H}_t \tag{27b}$$

The constraints 28a - c check the legislation regarding the maximum allowable load and the maximum axle loads and fifth wheel loads, related to model characteristics 1 (maximum weight) and 2 (axle loads) from table 3:

$$\sum_{i}^{\mathrm{I}} \mathrm{m}_i \leq \mathrm{m}_{\text{legislation}} \tag{28a}$$

$$\sum \text{mass} * (\text{combined center of mass}) - (\text{x fifth wheel}) * \frac{1}{\text{(x bogies) - (x fifth wheel)}}$$
$$\leq \mathrm{mb}_{\text{legislation}}$$

$$\sum_{i}^{\mathrm{I}} \mathrm{m}_i * \frac{\sum_i^{\mathrm{I}} \mathrm{m}_i * \mathrm{x}_{i0} + \mathrm{lx1}_i - \mathrm{lx2}_i) * \mathrm{cmx}_i)}{\sum_i^{\mathrm{I}} \mathrm{m}_i} * \frac{1}{\mathrm{Bx}_t - \mathrm{Fx}_t} \leq \mathrm{mb}_{\text{legislation}} \tag{28b}$$

$$\sum \text{mass} - \text{bogie load} \leq \mathrm{mf}_{\text{legislation}}$$

$$\sum_{i}^{\mathrm{I}} \mathrm{m}_i - \sum_{i}^{\mathrm{I}} \mathrm{m}_i * \frac{\sum_i^{\mathrm{I}} \mathrm{m}_i * \mathrm{x}_{i0} + \mathrm{lx1}_i - \mathrm{lx2}_i) * \mathrm{cmx}_i)}{\sum_i^{\mathrm{I}} \mathrm{m}_i} * \frac{1}{\mathrm{Bx}_t - \mathrm{Fx}_t} \leq \mathrm{mf}_{\text{legislation}} \tag{28c}$$

Equation 29 adds a trailer to the solution when an item is packed to the trailer:

$$\sum_{i}^{\mathrm{I}} \mathrm{X}_{it} \leq \mathrm{M} * \mathrm{Y}_j \tag{29}$$

[1] Own work.

Equations 30 and 31 state that each trailer only has one conservation system and one building phase, according to characteristics 9 and 10 presented in table 3:

$$\mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{csrv}_i = \mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{csrv}_j \tag{30}$$

$$\mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{bphs}_i = \mathrm{Y}_{it} * \mathrm{Y}_{jt} * \mathrm{bphs}_j \tag{31}$$

Finally, equations 32a - d ensure the duality of the binary variables and that the overlap constraints are defined properly:

$$\mathrm{a}_{ik} + \mathrm{b}_{ik} + \mathrm{c}_{ik} + \mathrm{d}_{ik} + \mathrm{e}_{ik} + \mathrm{f}_{ik} \geq 1 \tag{32a}$$

$$\mathrm{a}_{ik} + \mathrm{b}_{ik} + \mathrm{c}_{ik} + \mathrm{d}_{ik} + \mathrm{e}_{ik} + \mathrm{f}_{ik} \leq 3 \tag{32b}$$

$$\mathrm{x}_{ik}, \mathrm{y}_{ik}, \mathrm{z}_{ik}, \text{integer} \tag{32c}$$

$$\mathrm{lx1}_i, \mathrm{lx2}_i, \mathrm{a}_{ij}, \mathrm{b}_{ij}, \mathrm{c}_{ij}, \mathrm{d}_{ij}, \mathrm{e}_{ij}, \mathrm{f}_{ij}, \mathrm{X}_{it}, \mathrm{Y}_t, \text{binary} \tag{32d}$$

4.2.5 Objective function

Equation 33 is the main objective function to minimize the number of trailers:

$$\min \mathrm{Z} = \sum_{t}^{\mathrm{T}} \mathrm{Y}_t \tag{33}$$

4.2.6 Limitation of mathematical formulation

The presented mathematical model is unable to account for both the item stability and the item movement constraints (characteristics 6 and 7 from table 3). Furthermore, the presented constraint regarding the axle loads (equations 28b - c), cannot be captured by regular MIP solver software such as Gurobi, due to the presence of decision variables in a denominator.
This again stresses the need for a different approach in the form of a heuristic, which will be presented in the following sub section.

4.3 Proposed model

In this paragraph, the proposed model is presented. First, an overview of the model structure is given. Next, the different functions are discussed.

4.3.1 Algorithm structure

Figure 24 shows a schematic of the main functions of the model. The output is shown at the bottom of the figure: a solution for the loading division. There are several steps required to get there.

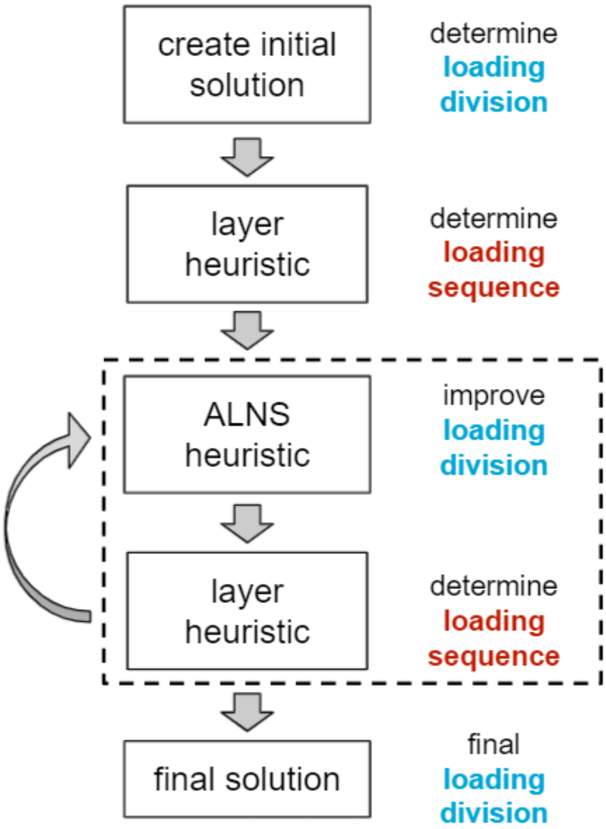


Figure 24: Algorithm overview

Of course, the first step of the model is to create an initial solution. This solution does not necessarily have to be a good loading division, or even a feasible solution, but serves as a starting point. Next, the loading sequence corresponding to the initial loading division is generated. At this step, some issues may occur regarding the loading space. For some trailers, the loading space may be overestimated, for others, the space may be underestimated.
Using the results of the first iterations for the loading division and loading sequence, a new iteration for the loading division is started. The loading division is improved by moving items between trailers and removing or adding trailers. Subsequently, the corresponding loading sequence is generated to see if the new loading division indeed is an improvement. These two steps (within the dotted lines in figure 24) are the main steps of the model and are executed for multiple iterations, after which the model is terminated.

Naturally, the working principle of the model is only a global approach. In appendix A.2, the full structure of the model is thoroughly reviewed, including flow charts containing the in- and output for each function.

In the remainder of this section, the two functions responsible for the loading division and loading sequence as well as the objective function are briefly discussed.

4.3.2 ALNS function (loading division)

Following the results of section 3.3, the ALNS heuristic is chosen as the loading division method.

Working principle

The working principle of the ALNS method is as follows:

1. First, the items are sorted by their conservation system. Next, within a conservation system, the items are sorted by their building phase. A separate ALNS heuristic is executed for each unique conservation-phase combination.
2. An initial solution for the loading division is created by adding items to a trailer until the maximum allowable load is reached. Then, a new trailer is created.
3. At the end of the initial iteration, the loading sequence is executed. The loading sequence function places the items on the trailer assigned during the initial solution.
4. For the next iteration, the results from the loading sequence are evaluated. If for a certain trailer some items could not be loaded, these items are automatically removed from the current solution using a removal heuristic.
5. The first objective is to obtain a feasible solution. That is, a solution that allocates and packs all items on a trailer. To do this, repair heuristics are used to place removed items on new trailers.
6. If, after a certain iteration, a feasible solution has been obtained, the next step is to improve this solution. To do this, removal heuristics again break the existing solution by removing items from trailers. Next, the repair heuristics focus on re-allocating the items.
7. A simulated annealing criterion is used to determine whether a new solution is accepted. If the new solution is a better solution, it is always accepted. If it is a worse solution, it may be accepted depending on the iteration number and the difference in objective value between the best and current solution.

4.3.3 Algorithm objective function

Of course, the ALNS heuristic requires an objective to decide whether a new solution is accepted or rejected. The objective function used in the model is an extended version of the objective function in equation 33 presented in the mathematical model: it is a composite of two objectives, each with their own weight, to help distinguish improvements between similar-looking solutions:

$$\min \mathrm{Z} = \mathrm{W}_1(\text{nr. of trailers}) + \mathrm{W}_2(\text{fill\%}^*) \tag{34}$$

In equation 34, W_1 and W_2 are weights and fill%* is defined as the filling percentage of the most empty trailer in the solution. Naturally, the number of trailers is the primary objective. This is always an integer (there are no half trailers), and every solution with less trailers than the current solution should always be defined as a better solution. The filling percentage is a secondary objective to help identify better solutions. An example of the utility of fill%* is shown in figure 25. The number of trailers is expected to rarely fluctuate throughout the iterations. When only considering the primary objective, this would mean that for a large amount of iterations, the new objective may stay exactly the same, which makes it impossible for the model whether a solution is actually improved. That is where the secondary objective fill%* comes in. It is assumed that when the most empty trailer has become more empty in a new solution, as the second part of figure 25 shows, the new solution is better than before: the lower the filling percentage, the closer the trailer is to being empty, and the higher the probability that the trailer will be empty in a future iteration. An empty trailer means a lower primary objective.

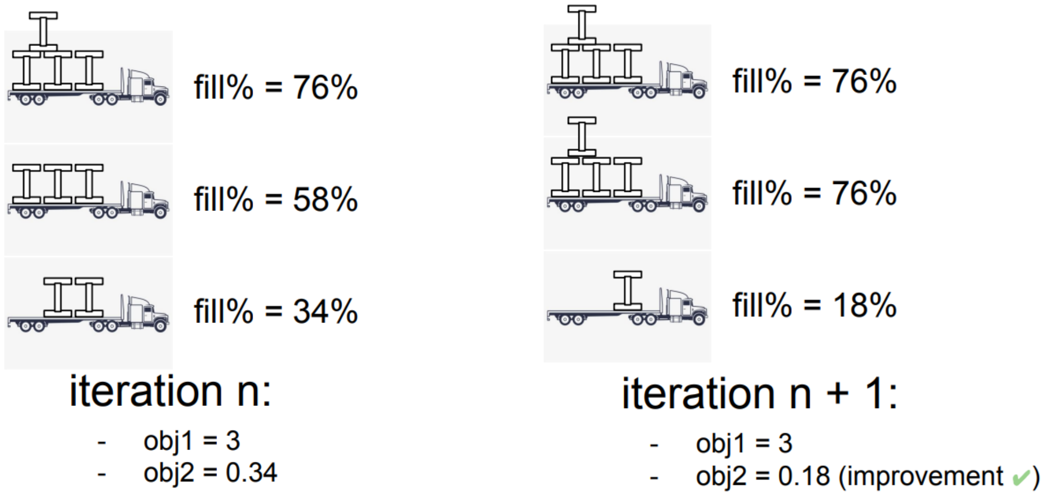


Figure 25: Description of the secondary objective

The next step is to find the correct weights, so that the condition is met that every solution with less trailers than the current solution should always be accepted. To do this, it is best to allow zero interference between the objectives.
Therefore, the weights are $W_1 = 1$ and $W_2 = 0.01$: since the filling percentage is always a value between 0 and 100, with this setup the secondary objective can never exceed a value of 1 (100% times 1), and overrule the primary objective.

4.3.4 Layer heuristic (loading sequence)

For the second function, as discussed in section 3.3, none of the presented existing methods is able to capture all required characteristics of the problem and at the same time have a reasonable computation time. Therefore, a new method is introduced: the layer heuristic.
The layer heuristic is designed to create a loading sequence according to the loading division presented by the ALNS function. Therefore, it receives the items allocated to a certain trailer as input, and returns the loading sequence for that trailer. The heuristic has the following working principle:

1. All items assigned to the trailer are sorted by their height.
2. The first layer is created at the bottom of the trailer. In a layer, all items have the same height, to sustain loading stability. The longest item present in this layer forms the length bound of the layer. Initially, the width bound of the layer is the trailer width. There are two options:
   (a) The width and length bounds of the trailer are reached before all items of the same height are used: in this case, there is only one bottom layer.
   (b) All items of the same height fit within the layer before the trailer bounds have been reached: in this case, a new bottom layer is created positioned left of the current bottom layer. This process is repeated until the trailer bounds are reached.
3. When the bottom of the trailer is full, new layers are created on top of the bottom layers. For the first new layers, the bounds are the length and width of the first bottom layer, until no more new layers can be added on the first bottom layer. For the next new layers, the second bottom layer forms the basis, and so on.

4. The layer building continues until either of the next two conditions has been met:
   (a) There are no items left: the process is terminated and the loading sequence is a success: all items have been placed.
   (b) One of the constraints is violated: the maximum allowed weight, or the maximum allowed bogie or fifth wheel loads are exceeded, or the layers are stacked beyond the maximum trailer height. In this case the loading sequence is not feasible, as not all items could be placed.
5. When the layer heuristic is terminated, and there are no items left, the trailer has obtained a feasible solution. The resulting weight, axle loads and maximum height are stored to serve as input for the loading division heuristics.

As figure 26 shows, comparing the layer heuristic to the methods introduced in section 3.3, the total score of the layer heuristic (21) is far superior to the other methods.

	orientation MIP		3D raster MIP		sub volume heuristic		irregular shape heuristic		layer heuristic	
	score	total	score	total	score	total	score	total	score	total
computational time (3)	2	6	1	3	3	9	2	6	3	9
complexity (1)	3	3	3	3	2	2	1	1	1	1
feasibility (3) - 3D - item support - inter-flange placement	1 yes no no	3	3 yes yes yes	9	2 yes yes no	3	1 no no yes	3	3 yes yes yes	9
financial (1)	1	1	3	3	3	3	3	3	3	3
total	**7**	**13**	**10**	**18**	**10**	**17**	**7**	**13**	**10**	**21**

Figure 26: Proposed layer heuristic methods compared to available methods in literature. Scores range from 1 (bad) to 3 (good), weights range from 1 (less important) to 3 (very important).

The layer heuristic can be seen as related to the sub volume heuristic presented by Jin et al [6]. Therefore, it is expected that computation times are comparable to the sub volume heuristic, thus rewarding the layer heuristic the maximum score.
Because the new heuristic involves many practical constraints related to this research, it is complex to turn into code, resulting in a low score for complexity. Naturally, a full score is awarded for feasibility, as the method is developed specifically for this purpose, and another full score for financial, as the model is built by the author of this report and therefore is free.

The ALNS function and the layer heuristic continuously interact with each other, alternately creating a new loading division and its respective loading sequence. Finally, after the model has run for a certain number of iterations, the resulting best objective is collected.
When the ALNS heuristics for all conservation-phase combinations are finished, the total best objective can be calculated.
This best solution is then complemented by a pallet item function, and the solution is improved once more by a merging function.
The pallet item function assigns all the items that are small enough to fit on a pallet to the loading division. No new loading sequence is created, which is explained in appendix A.2. Next, the merging

function merges trailers with unique conservation-phase combinations, as long as the trailers have an equal conservation system and similar building phase. A similar building phase means that the main phase is equal (201 and 224 belong to main phase 2, for example). Only unique conservation-phase combinations are evaluated during the merging process.

After the merging function is finished, the model presents its results and is terminated.

4.4 Discussion

In this section, the proposed approach has been given. First, the research scope has been defined, after which the mathematical model of this research has been formulated. It showed that this mathematical model is unable to account for all model characteristics, thereby again confirming the need of a heuristic model. In the final part of this section, the proposed model has been presented, in line with the defined sub questions for this section:

3. How can the loading division process be automated, accounting for the loading conditions?
4. How can the presented loading sequence problem be implemented in a 3D optimization model?

The provided model consists of an ALNS heuristic, that is responsible for the loading division, and the new proposed layer heuristic, that generates a loading sequence. In this section, a description of both heuristics has been given, so that sub questions 3 and 4 can be considered answered.

5 Verification and validation

In this section, the model verification and validation are presented. Sargent [8] defines model verification as 'ensuring that the computer program of the computerized model and its implementation are correct'. This translates to: does the model work as intended?
On the other hand, the model validation can be defined as the 'substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model' (Sargent [8]). In other words: is the proposed model the correct model for the presented problem?
An answer to both questions will be given in the next paragraphs.

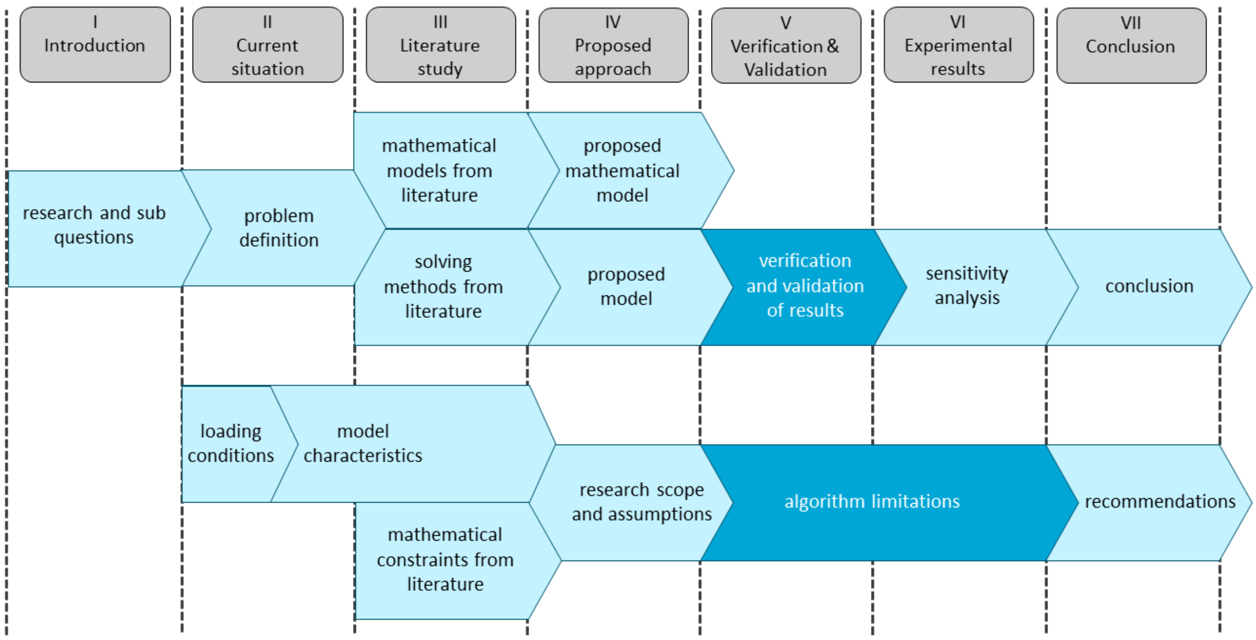


Figure 27: Report structure

5.1 Case scenario

The data set used in this section is a project recently executed at ASK Romein. The project regards a data center consisting of over 3000 steel components. In the traditional loading division created for this project, a total of 121 transport lots or trailers were created.

For the verification, several hypotheses are tested to see if the model works according to its design. Tests are performed on the layer heuristic, the ALNS algorithm and the functions responsible for data collection. The tests require samples of the full data set, consisting of multiple conservation systems and building phases.
The validation is executed using the preliminary results for the full model and several transport lots from the traditional loading division. The preliminary results are used to validate the full model, whereas the traditional transport lots are used to determine the performance of the layer heuristic.

5.2 Verification

The model consists of multiple layers with several important functions that each require a verification. In this section, all executed verification tests are listed for the ALNS model, the layer heuristic and the data collection functions.

Starting with the verification of the ALNS heuristic in table 4, most tests are related to the performance of the removal and repair heuristics of the model. Furthermore, attention is given to the calculation of the objective function, as this is a vital component of the model. As table 4 shows, all verification tests are passed.

Test description	**Hypothesis**	**Result**
ALNS heuristics	All heuristics are considered by the algorithm	TRUE
Pre-analysis:	Pre-analysis is executed before each iteration	TRUE
Objective function	The objective is calculated from the current solution	TRUE
Removal heuristic (I):	For unfeasible solutions, the removal heuristic only removes the unplaced items	TRUE
Removal heuristic (II):	For feasible solutions, the removal heuristic removes a fixed number of items	TRUE
Repair heuristics:	Items are added until the maximum load is reached	TRUE
Height repair heuristic:	Highest score awarded to trailers with most common height	TRUE

Table 4: Verification table ALNS heuristic

In table 5, the layer heuristic is verified. The feasibility of the provided solution is an important part of the heuristic, and therefore it is thoroughly tested. The feasibility depends on the violation of the loading conditions. Again, all tests are passed.

Test description	**Hypothesis**	**Result**
Feasible solutions	A solution is only feasible when all items are placed	TRUE
Item characteristics	All items on the trailer have an equal conservation system and phase	TRUE
Layer width	A layer is closed when the trailer width is exceeded	TRUE
Layer length	A layer length never exceeds the length of the longest item or the trailer length	TRUE
Trailer height:	When a layer exceeds the trailer height, it is not accepted	TRUE
Axle loads	No item is placed when one of the axle loads is exceeded	TRUE

Table 5: Verification table layer heuristic

Table 6 shows the verification results of the functions responsible for the data collection. This concerns the data that is presented as output of the model, and as such should be guaranteed to be correct. Similar to the other verification tests, the data collection functions have passed all tests.

Test description	**Hypothesis**	**Result**
Best objective per conservation system:	Objective equals sum of all phases within conservation system	TRUE
Total objective	Objective equals sum of all objectives	TRUE
Computation time:	Initial + iteration time equals total time	TRUE
Trailer merging:	Objective before merging equals final objective minus merged trailers	TRUE
First best solution	First reported solution corresponds to objective plot	TRUE

Table 6: Verification table data collection

The fact that all verification tests presented in this section are passed strongly suggests that the model works as designed, and the model can be considered verified.

5.3 Validation

In the following paragraphs, the validation of the layer heuristic and the complete proposed model is executed. It is investigated whether the layer heuristic and model fit the presented problem. Of course, it should be mentioned that a 100% proof of validation does not exist: all models are, in one way or another, a simplified representation of reality.

5.3.1 Layer heuristic

It is not easy to validate the working principle of the layer heuristic. For example, the optimal solution is not known.
Therefore, a different approach is used: since the loading division for the case scenario data set has already been created, and therefore the original transport lots are already known, the layer heuristic is tested against these transport lots. For each transport lot, all items are generated in the layer heuristic, to test if the layer heuristic is capable to successfully produce a loading sequence. Two examples of these tests are shown in figures 28 and 29.

In the figures, the left plot is a schematic view from the rear of the trailer, the top right is a (right) side view and the bottom right is a top view. The grey bar represents the trailer bed. The items in figure 28 are all H-shaped profiles, with two of them assemblies with steel plates welded to them (the yellow and red rectangles). The items in figure 29 are more complex-shaped, all containing multiple plates. The center right figure shows the location of the total load and resulting axle loads.
Both presented loading sequences are feasible, meaning all items from the transport lots could be placed concerning all loading conditions presented in this report. In fact, this is not just the case for these specific examples, but for all other traditional transport lots that have been tested. Ultimately, the conclusion that can be drawn from this is that the loading sequence seems capable of correctly simulating the actual loading sequence.

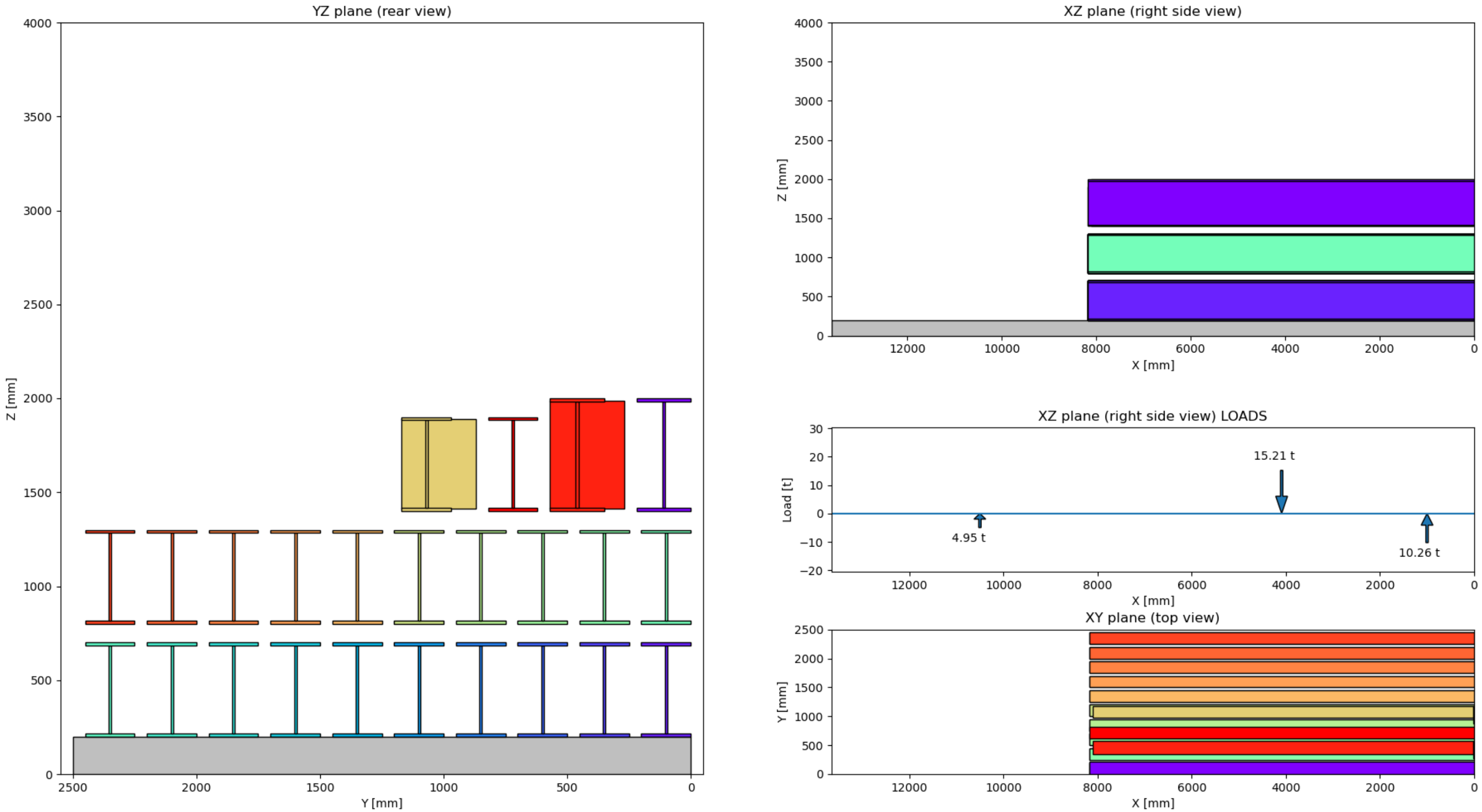


Figure 28: Loading sequence transport lot 331

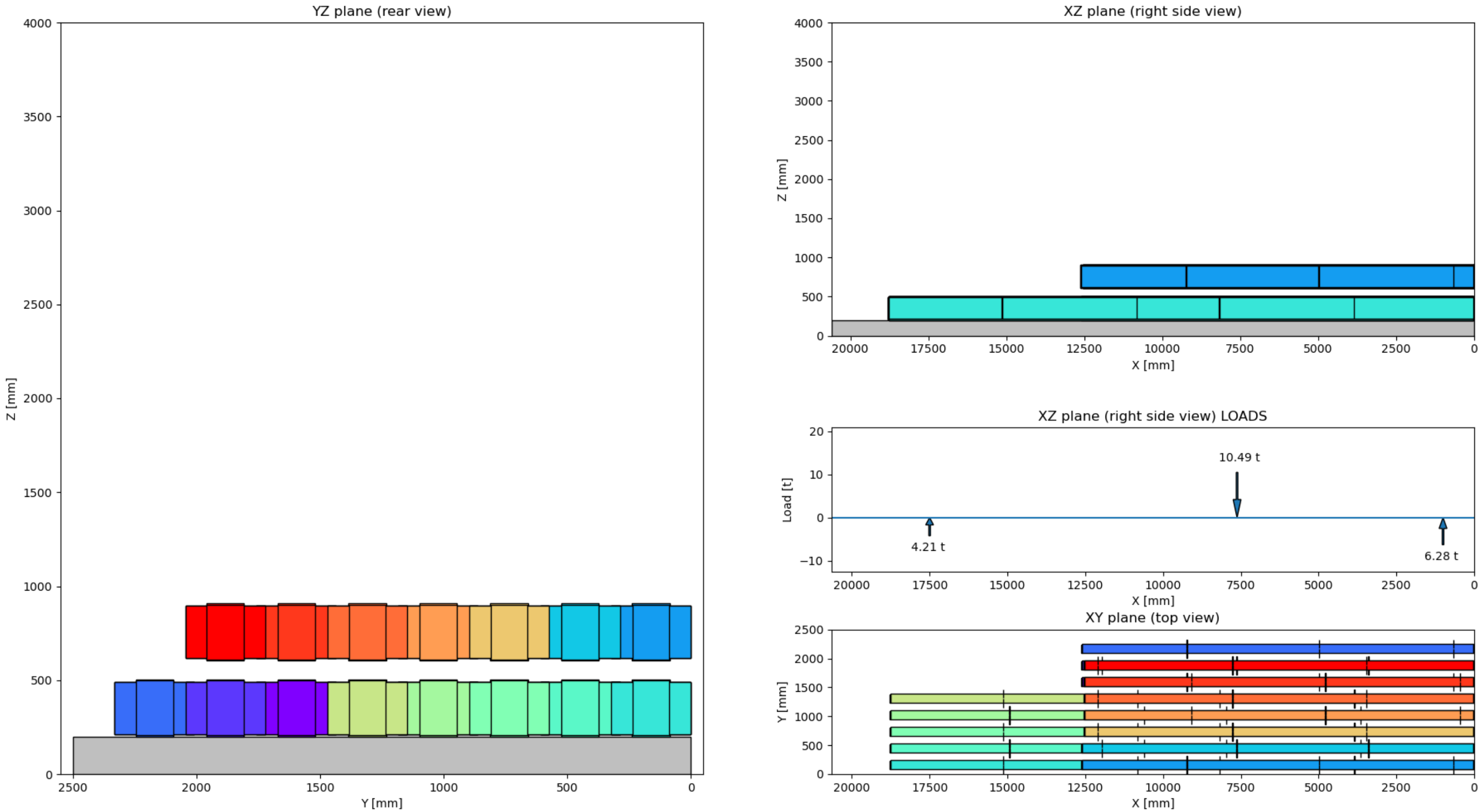


Figure 29: Loading sequence transport lot 357

5.3.2 Proposed model

To provide a validation for the proposed model, as mentioned before, the model is tested using the data set described in the case scenario of section 5.1. The output values for the loading division as well as the resulting loading sequence are evaluated.

Table 7 shows the model output compared to the traditional result for the loading division. The traditional loading division from the case scenario consists of 121 trailers, and the proposed model produces a loading division of only 92 trailers. This means that the loading division can be reduced by at least 29 trailers or 24%, which is an enormous reduction.

	current approach:	proposed approach:
guaranteed solution:	121 trailers	92 trailers **(-24%)**
best solution:	121 trailers	90 trailers **(-26%)**
time needed	approx. multiple hours	<10 min
loading conditions	loading division	loading division and loading sequence

Table 7: Results for 25 tests with 4 ALNS iterations, a deterministic initial solution and 10 removed items per iteration

Unfortunately, it was not possible to test the model against a different data set. As mentioned before, the case scenario data concerns a relatively small project at ASK Romein. Therefore, to test the model scalability, all components in the original case have been multiplied by 5 to see if the model still works and if the results are satisfactory. Of course, since the model with the original data is able to create a loading division of 92 trailers, the same data multiplied by five should result in a loading division of at most 5x92 trailers.
Table 8 shows the results. In the second column the original results are shown, the third column shows the results from the second column multiplied by 5, and the fourth column shows the model results using the 5 times the components of the original data set.

	results 1x:	expected 5x:	results 5x:
guaranteed solution:	92 trailers	<460 trailers	344 trailers **(-25%)**
best solution:	90 trailers	<450 trailers	340 trailers **(-24%)**
time needed	5 min	25 min	45 min **(+80%)**

Table 8: Comparing the model performance with the original data set and 5 times the original data set

As can be seen in table 8, the results are a lot better than expected. This can be explained as follows: in the original results, there were some trailers with only a few items placed on them. This was not caused by a malfunction of the model, but simply because of the fact that there were very little items with that particular conservation system or building phase. If for example, in the entire data set, there are only 6 (small) items with a particular conservation system, the trailer they are assigned to will be almost empty. If then all items are multiplied by 5, there are 30 of these items, and they can probably all be placed on one trailer. As a result, the expected number of 5 trailers for these items can be reduced by 4, and this explains the significant reduction in trailers.
Something else that should be mentioned is the fact that the computation time is much higher than 5 times the original computation time. This can be explained by the fact that a 5 times larger data set requires more than 5 times system memory to execute the model.

Next, examples of the layer heuristic from transport lots created in the new model (figure 31 and 32), will be compared an actual example of a loading sequence (figure 30).
Comparing figure 30 and 31, strong similarities can be seen. Items are divided into different layers.

Within these layers, items are placed tight against each other. H-shaped profiles are mostly packed in standing direction.
The loading sequence of figure 30 has a pyramid-shaped loading to have a good load distribution in the width direction of the trailer. In the top layer of the loading sequence of the proposed model, the items are placed more to the right. This is a result of the way the layer heuristic is designed. However, this is not a critical problem, since the items can easily be re-arranged towards the middle of the trailer.

The loading sequence presented in figure 32 contains a few large steel columns (the larger colored squares). Furthermore, some of the steel plates that are sticking out from an item's profile are placed within the flanges of another item, similar to what can be seen in figure 30.
Looking at the weight distribution in the width direction again, the placement of the heavy columns on the right hand side of the trailer may be a concern, but again it is not a critical issue as it can be resolved by re-arranging some of the items.
Overall, it can be concluded that although the loading sequences generated by the layer heuristic are not perfect, they are sufficiently capable of representing the real loading sequence, and thus can be used to check the loading conditions related to the loading sequence.

Figure 30: Loading sequence ASK Romein Roosendaal

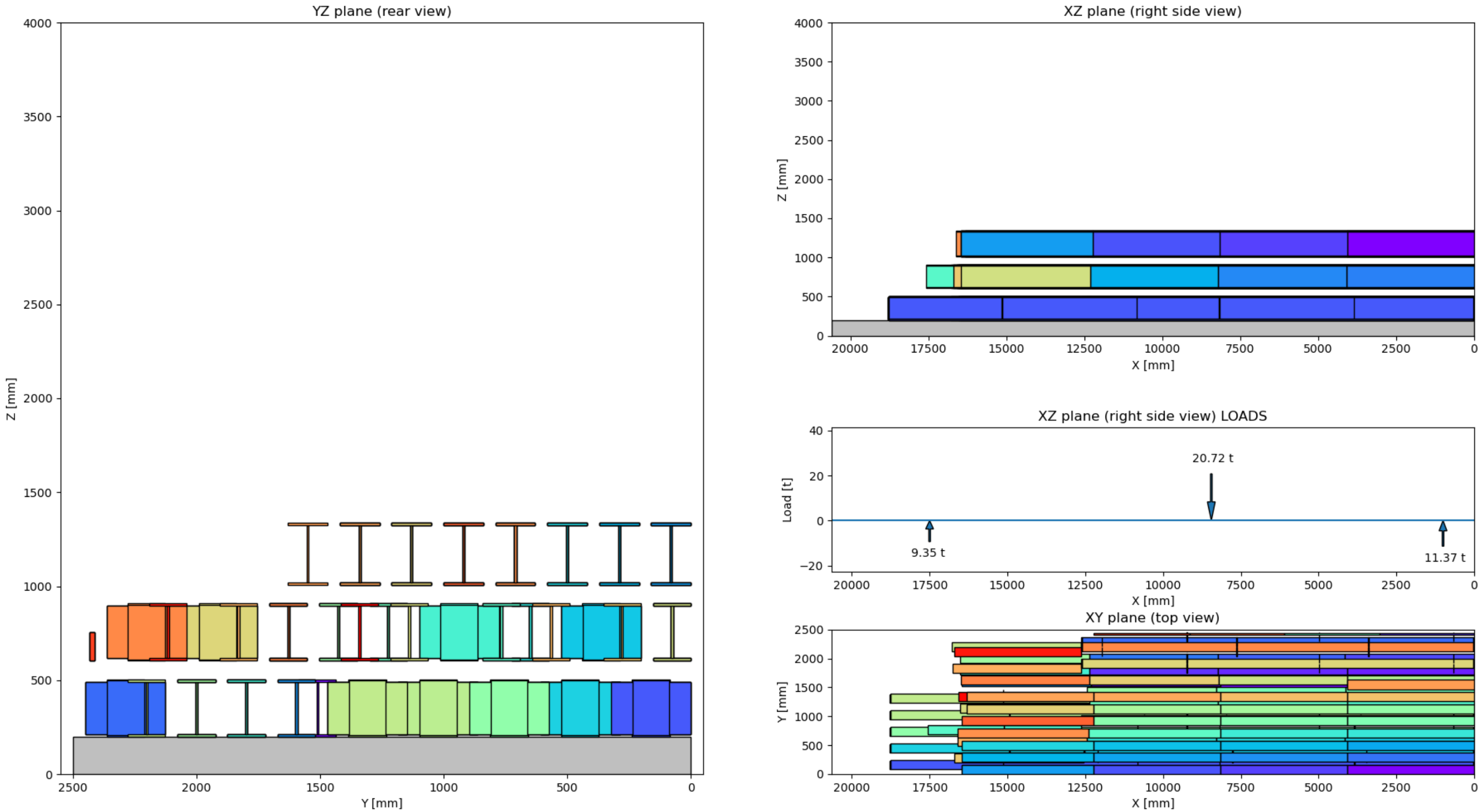


Figure 31: Loading sequence example conservation T−, phase 452

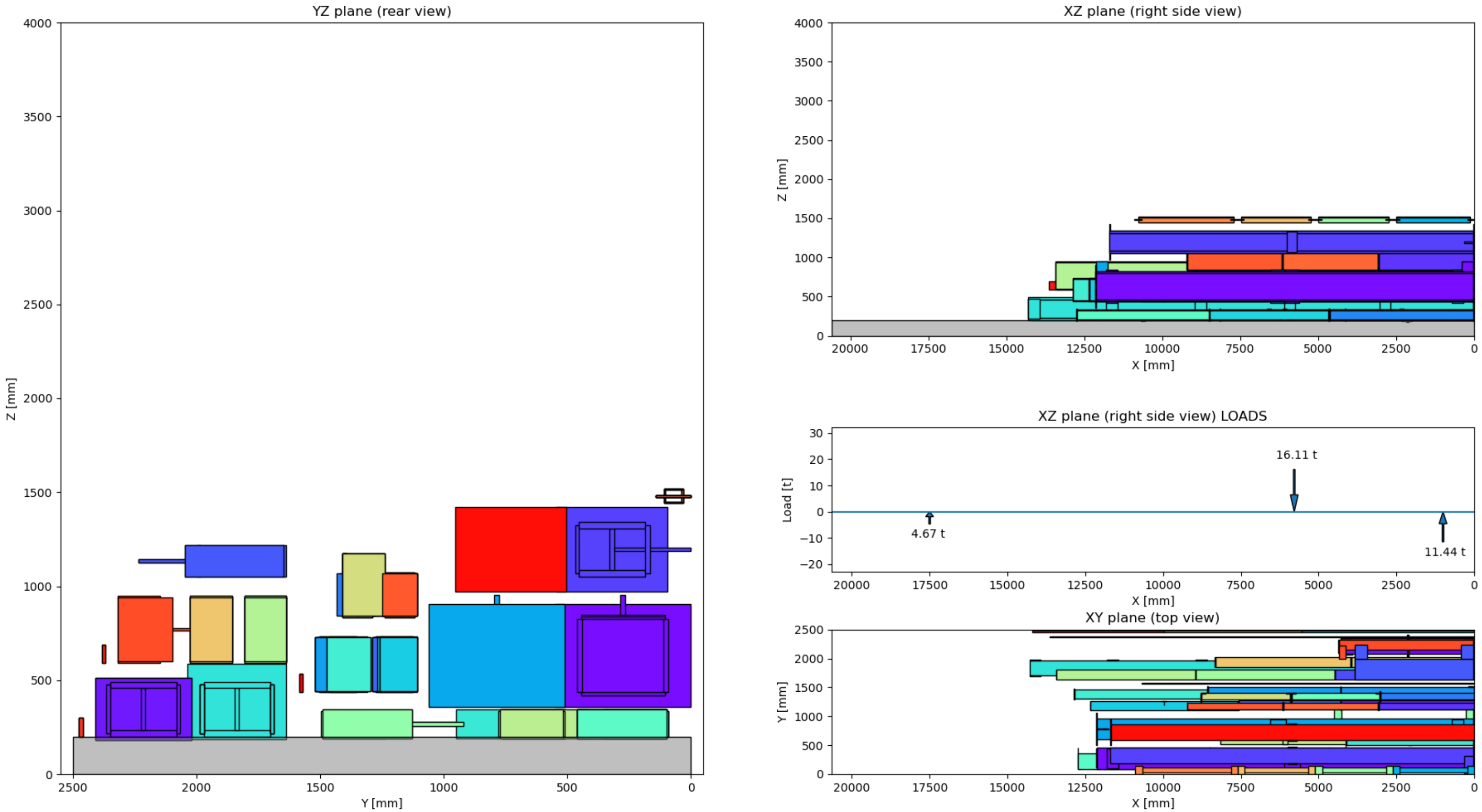


Figure 32: Loading sequence example conservation BW60, phase 202

5.4 Discussion

The sub questions assigned to this section are as follows:

5. How can the loading division and loading sequence models be verified?

6. How can the loading division and loading sequence models be validated?

The verification mentioned in question 5 and the validation mentioned in question 6 have been presented in subsections 5.2 and 5.3, respectively.
The verification tests of sub section 5.2 have all been passed by the proposed model and layer heuristic, which is a clear indication that they work as intended.
As mentioned before, the validation of the layer heuristic is a bit more difficult, but can be done by comparing its output to transport lots created in the traditional loading division. All transport lots tested resulted in feasible solutions for the loading sequence by the layer heuristic.
For the full model, the results for the loading division are a significant improvement compared to the traditional loading division. Furthermore, the model is also capable of handling much larger data sets, albeit at the cost of larger computation times, thereby confirming the scalability of the model.

6 Experimental results

In this section, the experimental results will be presented. First, several experiments will be conducted in the form of sensitivity analyses. Next, the best parameter values will be used to determine the overall solution, which will be compared to the existing solution at ASK Romein.
To secure the validity of the results, each test is conducted 25 times, thereby reducing the effects of random chance. The experiments have been conducted on a notebook laptop with 16 GB RAM and an i7 quadcore processor. The model is implemented in Python version 3.7.6 and executed in the Anaconda Spyder environment, version 4.1.3.
As mentioned in section 4.3.3, the objective function is a composite of two objectives: the primary objective is the number of trailers, which is the key performance indicator for this research. The secondary objective is the filling percentage of the least-filled trailers, which only plays a role in the process of finding better solutions. Therefore, it should be mentioned that for the final result, the final filling percentage is not considered: only the number of trailers used for the loading division is important. Nevertheless, the results for the secondary objectives will sometimes be shown in the upcoming section to give a complete overview of the results.

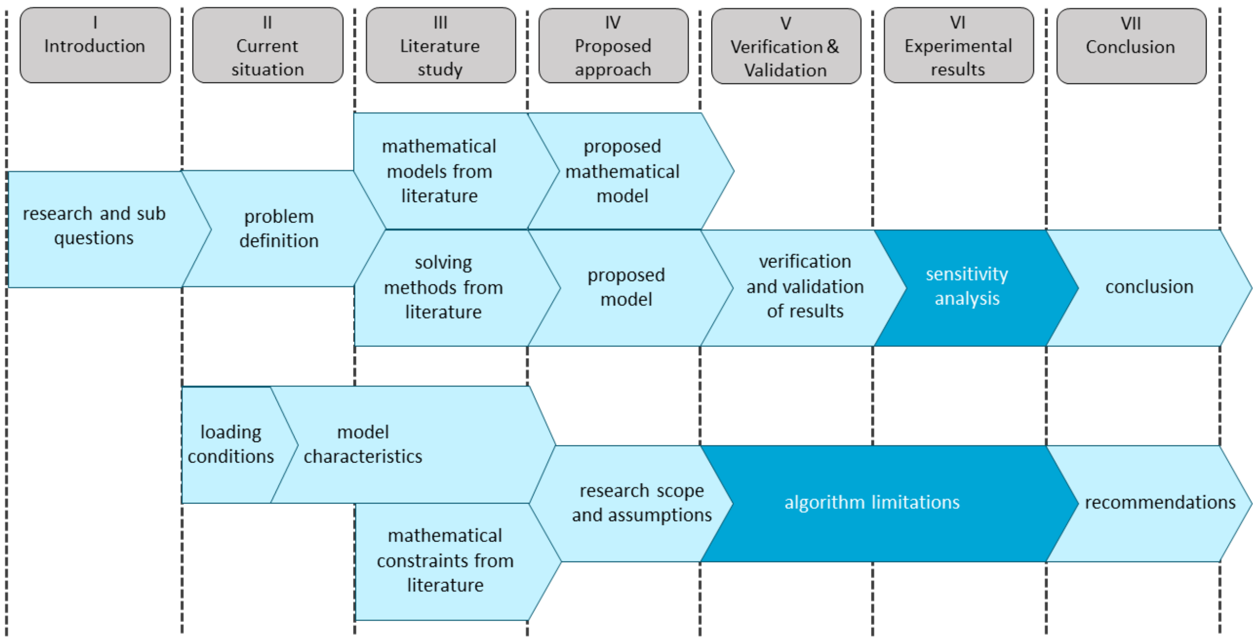


Figure 33: Report structure

6.1 Experiment overview

Regarding the case scenario of the experiments, the same data set is used as presented in section 5.1: a data center consisting of over 3000 components and a total of 121 transport lots.

Three types of tests will be conducted using this data set. First of all, the effect of the number of iterations for each ALNS model on the computation time and solution quality is tested. The parameter to be altered for this experiment is 'iterationsALNS'.
The goal of this test is to find a minimum number of iterations for which the solution is always feasible, and to find a good trade-off between the solution quality and the computation time. Of course, a minimum solution value is desired, but if the number of iterations causes excessive computation times with little improvement, a trade-off can be useful.
In this test, the full data set is used, to ensure that none of the subsets (phase-conservation combinations) results in an unfeasible solution.

Second, the effect of randomizing the initial solution is tested. The original initial solution heuristic does not contain any randomness factor. This means, that for each new experiment, the same initial solution is generated.
As preliminary results showed (presented in section 6.3), using the original, deterministic initial solution heuristic, most of the time leads to exactly the same results, even though the improvement heuristics do contain a randomness factor. This observation induces the possibility that the model may get stuck in a local optimum.
Therefore, in the second experiment, a slightly altered initial solution heuristic, containing a randomness factor, is tested against the original initial solution heuristic. The full data set is used, to ensure the validity of the results.

The third test focuses on another parameter, the number of items to be removed during each ALNS iteration: 'itemsremoved'. This experiment is designed to find the best value for this parameter.
To reduce testing computation times, instead of the full data set, samples of the data set are used. Each sample consists of exactly one phase-conservation combination. The samples are chosen carefully, so that they strongly represent the full data set. They are among the largest combinations, meaning they require at least six trailers each.

Table 9 shows the characteristics of the full data set, CompleetVracht, and the aforementioned samples:

	normal items (weight)	pallet items (weight)	total items (weight)
CompleetVracht	3028 (1084t)	3845 (33t)	6873 (1118t)
Phase 202-BW60	208 (81t)	20 (2t)	228 (82t)
Phase 305-T–	194 (79t)	62 (1t)	256 (81t)
Phase 451-T–	370(93t)	393(3t)	763(96t)

Table 9: Data characteristics

After all experiments have been conducted, finally the results are presented for the model with the optimal parameters, obtained from the tests.

To gain insight in the meaning of the results, several statistical methods are implemented in the presentation. They are listed below.

Mean

The mean or average of a set of data points is defined as the sum of all data points divided by the number of data points:

$$\text{mean } \mu = \frac{\sum_{\mathrm{i}}^{\mathrm{n}} \mathrm{x_i}}{\mathrm{n}} \tag{35}$$

With $\mathrm{x_i}$ a single data point and n is the number of data points.

Mode

While the mean gives a good impression of the given result, it can be heavily influenced by outliers. For example, for data set [2, 2, 2, 2, 8], the mean equals 4, but this number is significantly different from all the numbers present in the data set. Therefore, it is a good idea to also include the mode. The mode is defined as the most common number in the data set.

Population standard deviation

Two standard deviations: one for a full 'population', that is, of a full data set, and one for data samples. In this case, the standard deviation is taken from all available data points, meaning the former is used:

$$\mathrm{std}\ \sigma = \sqrt{\frac{\sum_{\mathrm{i}}^{\mathrm{n}}(\mathrm{x_i} - \mu)^2}{\mathrm{n}}} \tag{36}$$

Student t-test

For some cases, one could argue that the results of two separate tests are different enough to be considered significantly different. This can be useful to state that using a certain value for a parameter is a significant improvement.
The most popular way to prove that there is a significant difference between two tests, is the so-called student t-test. This test provides a value $\mathrm{t_{exp}}$. Using a critical t-value $\mathrm{t_{crit}}$, there are two possible outcomes to this test:

1. $\mathrm{t_{exp}} < \mathrm{t_{crit}}$: there is not enough evidence to reject the null hypothesis, $\mathrm{H_0}$: there is no evidence that A and B are significantly different.
2. $\mathrm{t_{exp}} > \mathrm{t_{crit}}$: reject $\mathrm{H_0}$: A and B are significantly different within the confidence interval.

When comparing data set A with data set B, the formula to obtain $\mathrm{t_{exp}}$ is given in equation 37:

$$\mathrm{t_{exp}} = \frac{|\mu_{\mathrm{A}} - \mu_{\mathrm{B}}|}{\sigma_{\mathrm{AB}}\sqrt{\frac{1}{\mathrm{n_A}} + \frac{1}{\mathrm{n_B}}}} \tag{37}$$

Where σ_{AB} is the pooled standard deviation:

$$\sigma_{\mathrm{AB}} = \sqrt{\frac{(\mathrm{n_A} - 1)\sigma_{\mathrm{A}}^2 + (\mathrm{n_B} - 1)\sigma_{\mathrm{B}}^2}{\mathrm{n_A} + \mathrm{n_B}}} \tag{38}$$

The value for $\mathrm{t_{crit}}$ depends on the degrees of freedom, N, and the desired confidence interval. For this report, a confidence interval of 95% is taken, meaning there is a 95% chance that the resulting outcome of this test is valid. The degrees of freedom are defined as:

$$\mathrm{N} = \mathrm{n_a} + \mathrm{n_b} - 2 \tag{39}$$

Since each data set contains exactly 25 data points, N = 48. The resulting value for $\mathrm{t_{crit}}$ equals 1.677 and will be used throughout this report.

6.2 Experiment 1: sensitivity analysis ALNS iterations

The goal of this experiment is to find the optimal number of iterations for the ALNS heuristic. To do this, two versions of the model are tested: one with a deterministic initial solution heuristic, and one with a randomized initial solution heuristic. In short, the reason for this is the possibility of local optima occurring for the former heuristic. This will be explained more thoroughly in subsection 6.3. Both approaches are executed for 4, 5, 6, 7, 10 and 50 iterations. Tables 10 and 11 show the mean, mode, standard deviation and best result for the primary objective and computation time for the deterministic and randomized approach, respectively. In figure 34, the mean primary objective is plotted against the computation time for various values of the parameter iterationsALNS. To keep

things readable, in this figure, the iterations that contain one or more unfeasible solutions are plotted with objective value 1000, and a broken axis is used. Finally, figure 35 is a bar graph of the computation times for all tested numbers of ALNS iterations.
As mentioned before, each test is carried out 25 times, so every mean in a figure or table represents the mean of 25 tests.

itemsremoved = 10	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
iterationsALNS:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
3	1000	1000	1000	0.000	0.00%	0.00%	98.03	4.100	3	0.000
4	91	92	91.8	0.400	8.80%	0.14%	119.32	6.291	4	0.000
5	90	92	91.56	0.637	8.80%	0.14%	142.75	8.553	4	0.000
6	90	92	91.64	0.557	8.81%	0.16%	168.51	12.445	4	0.463
7	90	92	91.76	0.512	8.78%	0.12%	194.77	16.229	4	0.392
10	91	92	91.8	0.400	8.77%	0.10%	271.41	13.158	4	0.814
50	90	92	91.52	0.574	8.77%	0.12%	1191.24	37.802	4	12.706

Table 10: Experiment 1: results deterministic approach

itemsremoved = 10	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
iterationsALNS:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
3	1000	1000	1000	0.000	0.00%	0.00%	98.28	4.848	3	0.000
4	92	1000	637.92	443.458	3.37%	4.15%	122.25	8.527	4	0.000
5	91	95	167	245.646	7.98%	2.52%	145.12	9.084	5	0.466
6	90	95	95.36	2.095	8.57%	0.71%	175.00	10.305	4	0.722
7	91	94	95.24	1.986	8.38%	0.63%	194.32	12.679	5	0.693
10	92	94	95.88	1.986	8.36%	0.62%	282.35	10.896	5	1.386
50	90	94	94.84	2.810	8.92%	0.96%	1258.64	70.629	50	15.235

Table 11: Experiment 1: results randomized approach

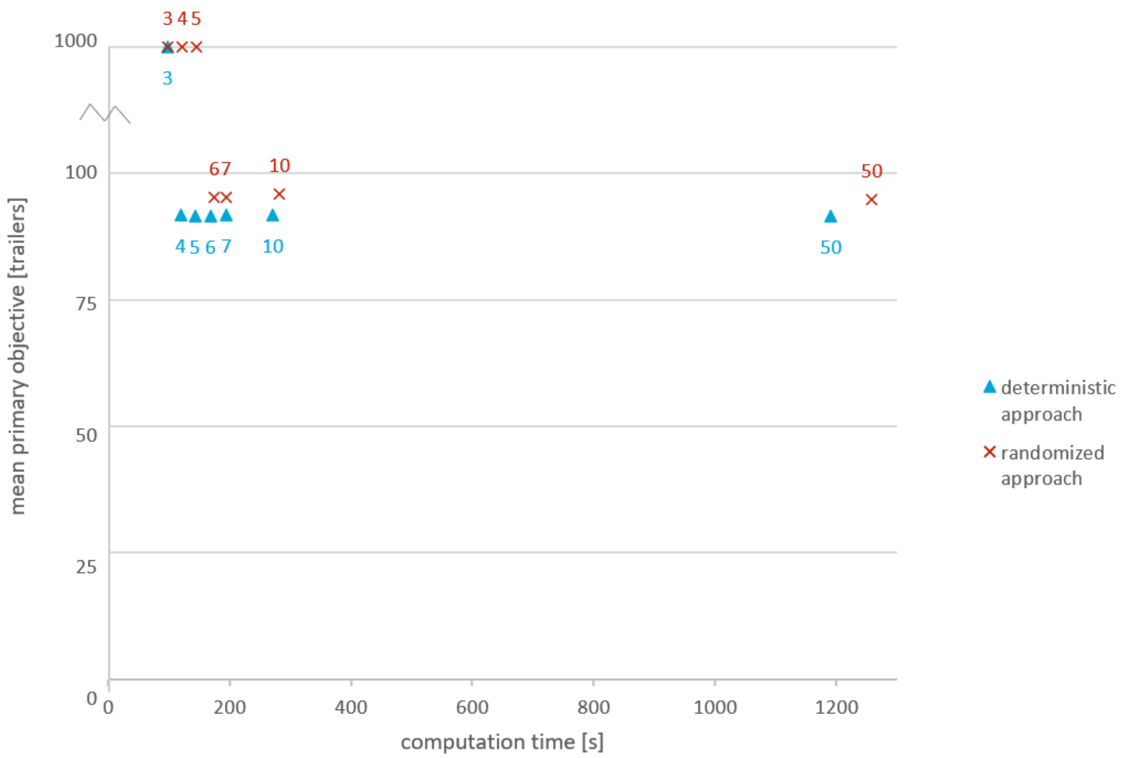


Figure 34: Experiment 1: primary objective versus computation times for various ALNS iterations

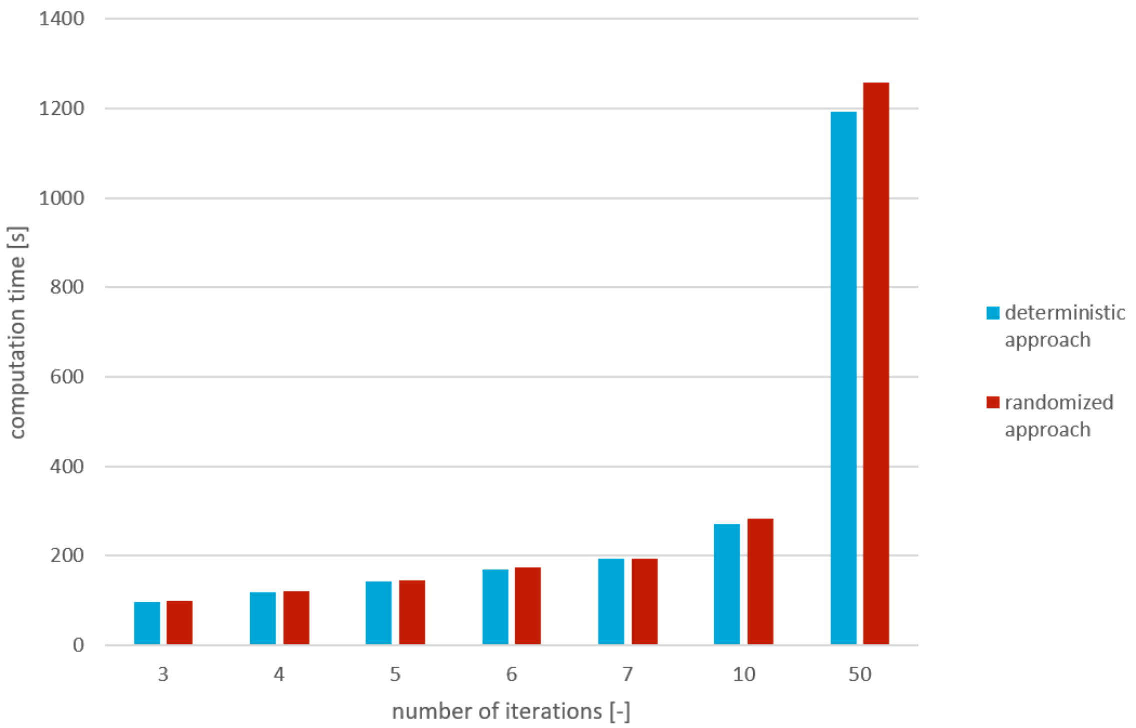


Figure 35: Experiment 1: computation times for various ALNS iterations

When analyzing the results, a few ideas occur. First of all when looking at the means in tables 10 and 11 and figure 34, it appears that the minimum number of iterations to always obtain a feasible solution equals 4 for the deterministic and 6 for the randomized approach. For both the deterministic and randomized approaches, using just three iterations always results in an unfeasible solution. For the randomized approach, using four or five iterations may sometimes lead to a feasible solution, but not always.
Furthermore, the test results show strong differences in computation times, as figure 35 shows. For a larger number of iterations, the computation time significantly increases. There also seems to be a minor difference in computation time between the deterministic and randomized approach, but this will be investigated in the next experiment.
In stark contrast to the computation time, looking at figure 34 again, the primary objective appears to show no improvement at all for an increasing number of iterations. This induces the statement that the best number of ALNS iterations equals the minimum number of iterations that produces a feasible result.
The best reported objective is 90 trailers. Looking at the minimum number of ALNS iterations for the deterministic approach, the tests with 4 iterations did not reach a solution for 90 trailers, whereas the tests with 5 iterations did. However, looking at the means and standard deviations, both results appear to be similar, meaning the tests with 4 iterations might have missed a 90 trailer solution by chance. To prove this, a student t-test is executed between the results for 4 and 5 ALNS iterations. The result in table 12 shows that there is no proof that both data sets are significantly different, so that it can be assumed that the model using 4 iterations reaches similar objectives as the model using 5 iterations, with better computation times, meaning the former is the best option.

t_exp:	t_th:	result:
1.594	1.677	not enough evidence to reject null hypothesis: no evidence that 4 and 5 are significantly different

Table 12: Student t-test for 4 and 5 ALNS iterations for the deterministic approach

For the model with the randomized initial solution, the first number of iterations that guarantees a feasible solution is equal to 6, so this is the proposed value.

6.3 Experiment 2: randomizing the initial solution

This experiment tests the difference between the model with a deterministic initial solution and an initial solution with a randomness factor. For the former, the generated initial solution is always exactly the same. For the latter, the order of items that are arranged to the trailers is varied, so that a different initial solution is obtained for each test.
As mentioned before, the reason for this test is the fact that preliminary tests on samples of the data set showed little variation in results for the model with a deterministic initial solution. Apparently, the initial solution has a large effect on the eventual outcome. Therefore, to prevent the model to be stuck in a local optimum, the initial solution is randomized.

In this experiment, the deterministic approach and the randomized approach are tested for 6 and 50 ALNS iterations. The value of 6 is taken because it is the lowest number of iterations for which both approaches always achieve a feasible solution, and 50 ALNS iterations are taken to investigate the results for a large number of iterations. Again, the full data set is used to ensure the validity of the results.

Figures 36 and 37 show all primary objectives for the tests with 6 and 50 ALNS iterations, respectively. It should be noted that the data presented here is equal to the data of 6 and 50 iterations from experiment 1. However, in experiment 2 all 25 results is plotted, whereas in experiment 1 only the mean value of the 25 tests was shown. For completeness, the results for the mean, mode and standard deviation are again given in tables 13 and 14.

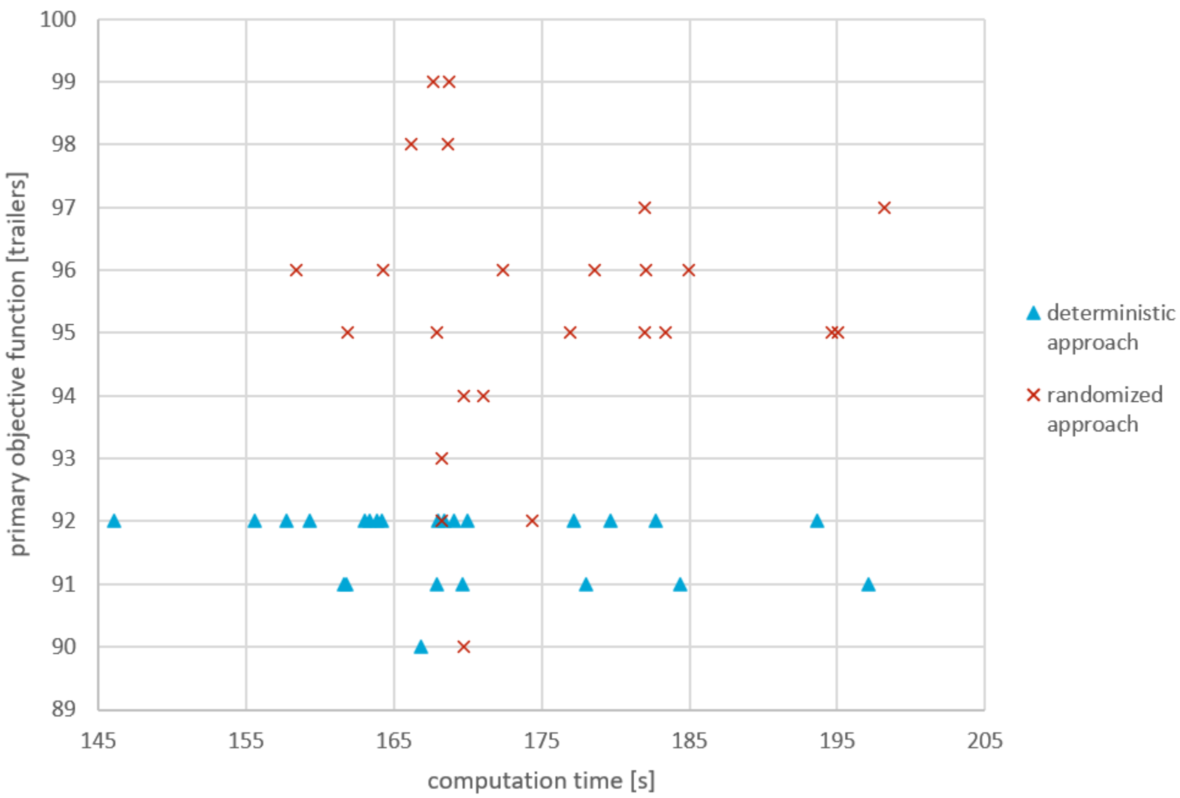


Figure 36: Experiment 2: primary objective function vs computation time for 6 ALNS iterations

itemsremoved = 10	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
iterationsALNS:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
6	90	92	91.64	0.557	8.81%	0.16%	168.51	12.445	4	0.463
50	90	92	91.52	0.574	8.77%	0.12%	1191.24	37.802	4	12.706

Table 13: Experiment 2: results deterministic approach

itemsremoved = 10	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
iterationsALNS:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
6	90	95	95.36	2.095	8.57%	0.71%	175.00	10.305	4	0.722
50	90	94	94.84	2.810	8.92%	0.96%	1258.64	70.629	50	15.235

Table 14: Experiment 2: results randomized approach

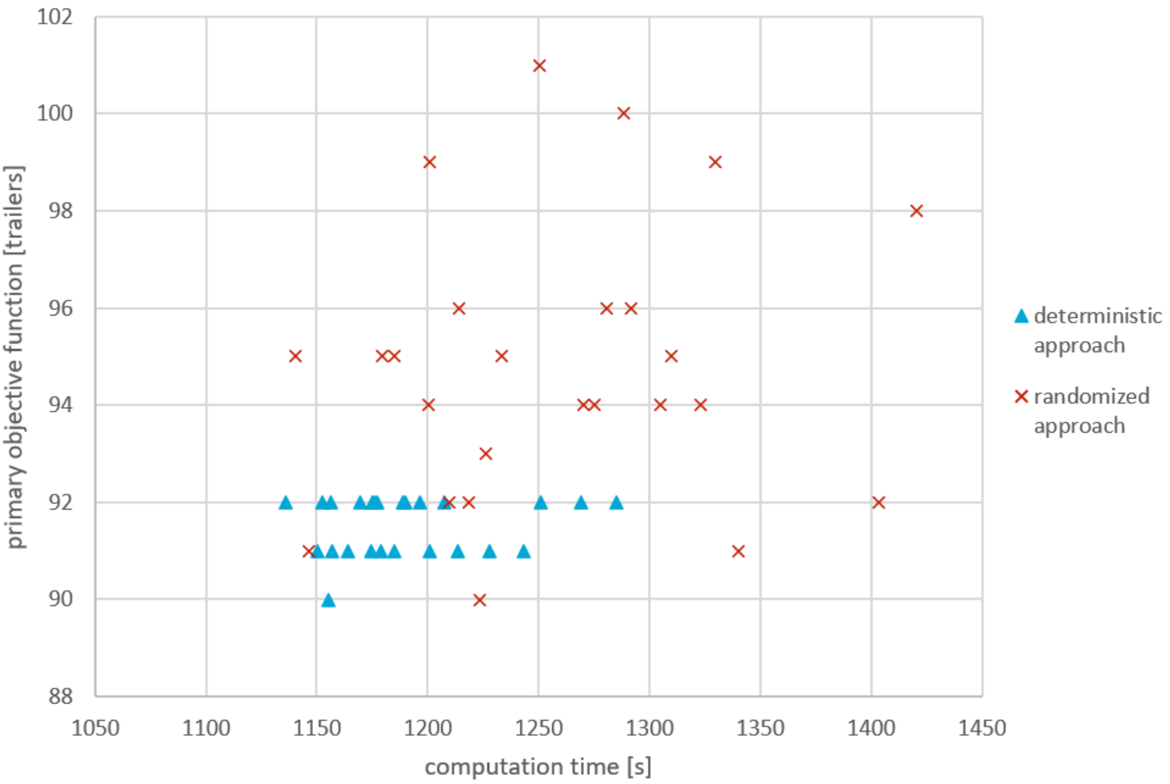


Figure 37: Experiment 2: primary objective function vs computation time for 50 ALNS iterations

The reason for giving the results of all 25 tests is to visualize the spread of the data points. Looking at figures 36 and 37, the deterministic approach on average has a better performance than the randomizing approach, with a smaller spread, which is confirmed by the differences in standard deviation for the deterministic and randomized primary objective in tables 13 and 14: 0.557 versus 2.095 for 6 iterations, and 0.574 vs. 2.810 for 50 iterations.
What is interesting is that each test type presented in this experiment reaches the best solution of 90 trailers exactly once out of 25 times. Even though the deterministic approach is significantly better, both approaches are able to reach the optimal solution of this report, for different numbers of iterations. This strengthens the idea that the model is strongly dependent on the initial solution. It appears that the initial solution of the model already is a relatively good solution, but is very hard to improve upon.

Figures 38 and 39 show an example of the development of the objective function for a single test with 10 ALNS iterations. It can be seen that although the secondary objective function is slightly improved upon throughout the number of iterations (even reaching a new best objective at the penultimate iteration), the best value for the primary objective, 127[1], is reached early in the process.

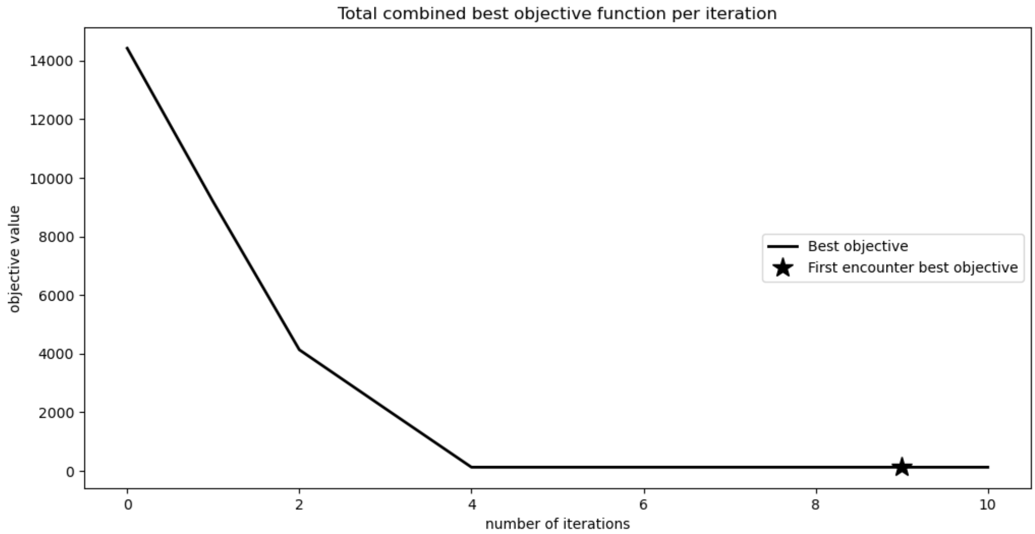


Figure 38: Progress of the objective functions for a single test with 10 ALNS iterations

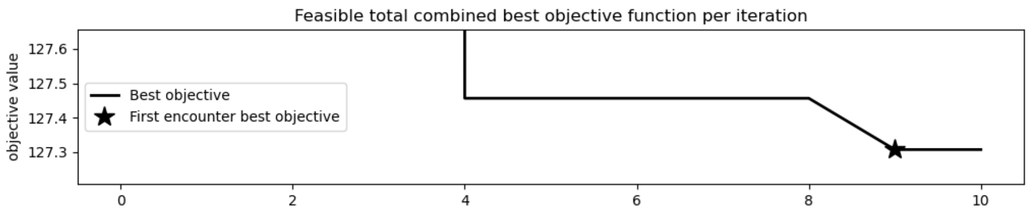


Figure 39: Progress of the feasible objective functions for a single test with 10 ALNS iterations

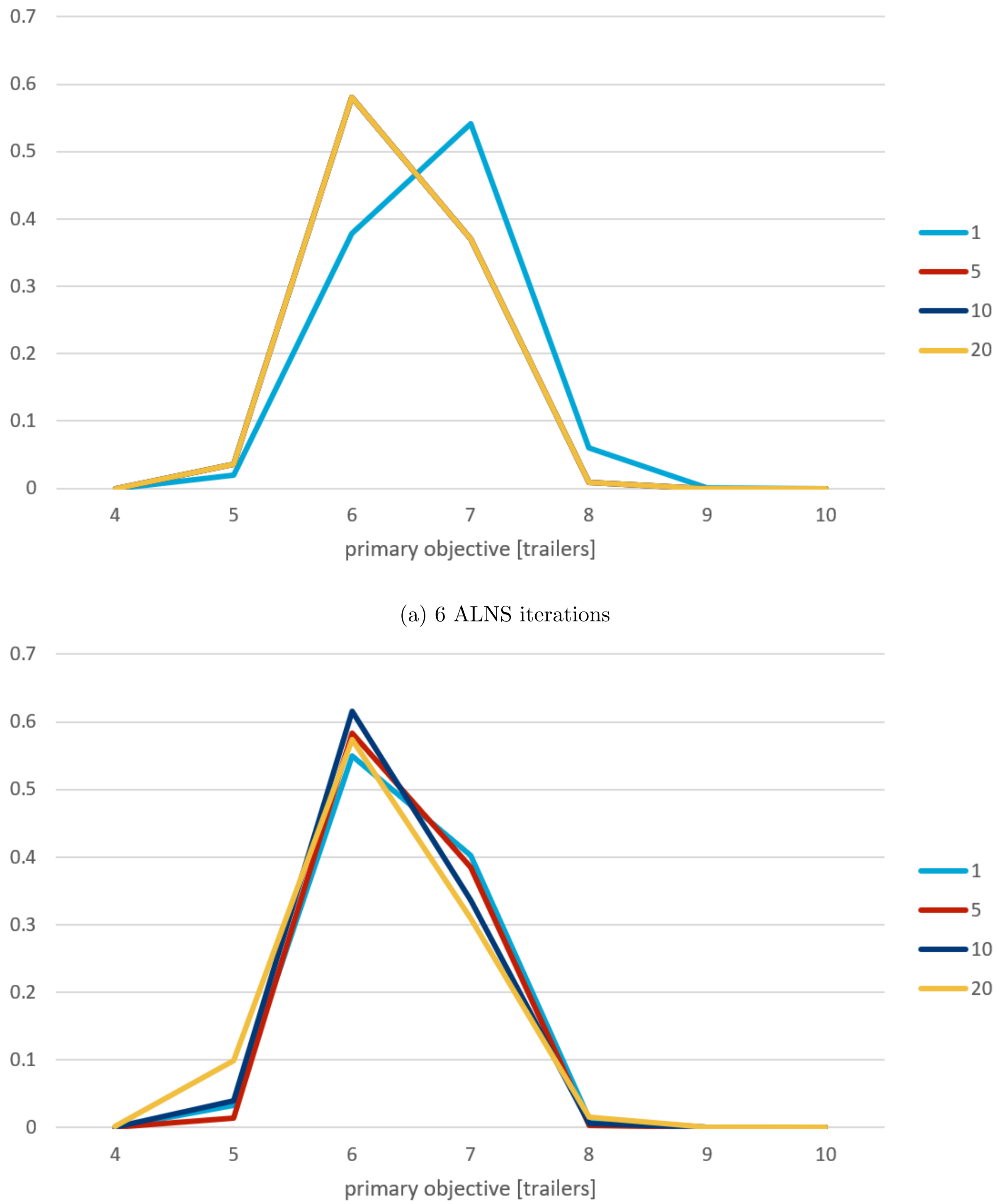
The fact that the initial solution is hard to improve upon may be caused by the number of items that is removed in each iteration. Therefore, the following experiment investigates the effects of varying the number of removed items.

6.4 Experiment 3: sensitivity analysis removed items

In the previous experiment, it is stated that the improvement of the initial solution may be hampered by the number of removed items in each ALNS iteration. This can be explained by the following example: let us say that the number of items to be removed in one iteration is 100. Out of these items, 99 are successfully placed in other trailers, but one item cannot be placed. This means the entire solution is marked as unfeasible, even if the placement of the other 99 items is an improvement. This definitely makes the improvement process more difficult, which is why the next experiment revolves around the number of removed items, described by the parameter 'itemsremoved'.

[1]It should be noted that the objective shown in figure 39 does not include the merging results, which explains why the objective is much higher than the values presented in this section.

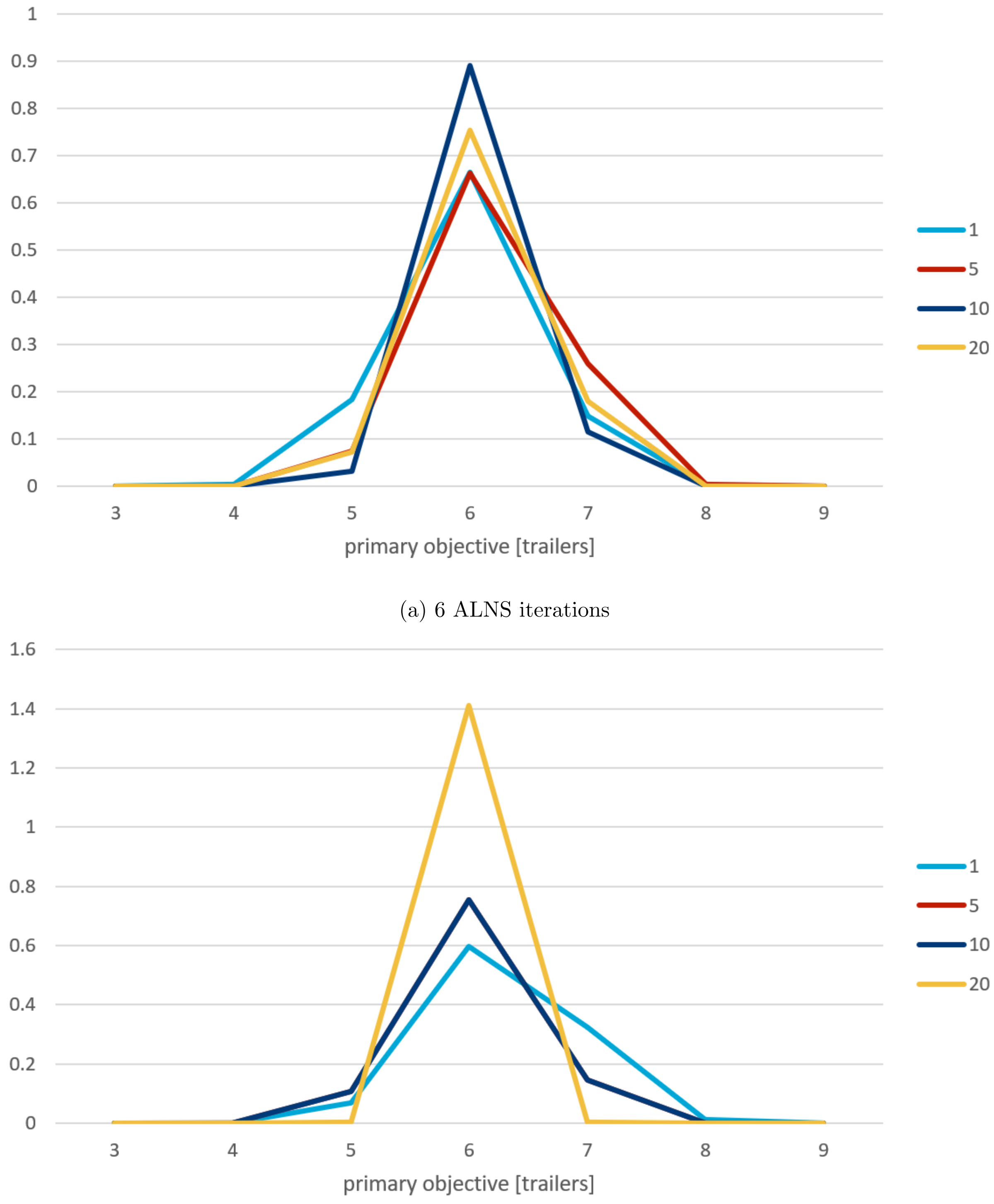
In this experiment, the solution quality and computation times are measured for itemsremoved = 1, 5, 10 and 20, and 6 and 50 ALNS iterations. As mentioned in the introduction of this section, to reduce the testing computation times, data set samples representing the full data set are used. This works, because the model deals with each phase-conservation combination separately.
In figures 40 - 42, the normal distributions for the different data sets are shown. The means, modes and standard deviations are given in tables 15 - 17.



(a) 6 ALNS iterations

(b) 50 ALNS iterations

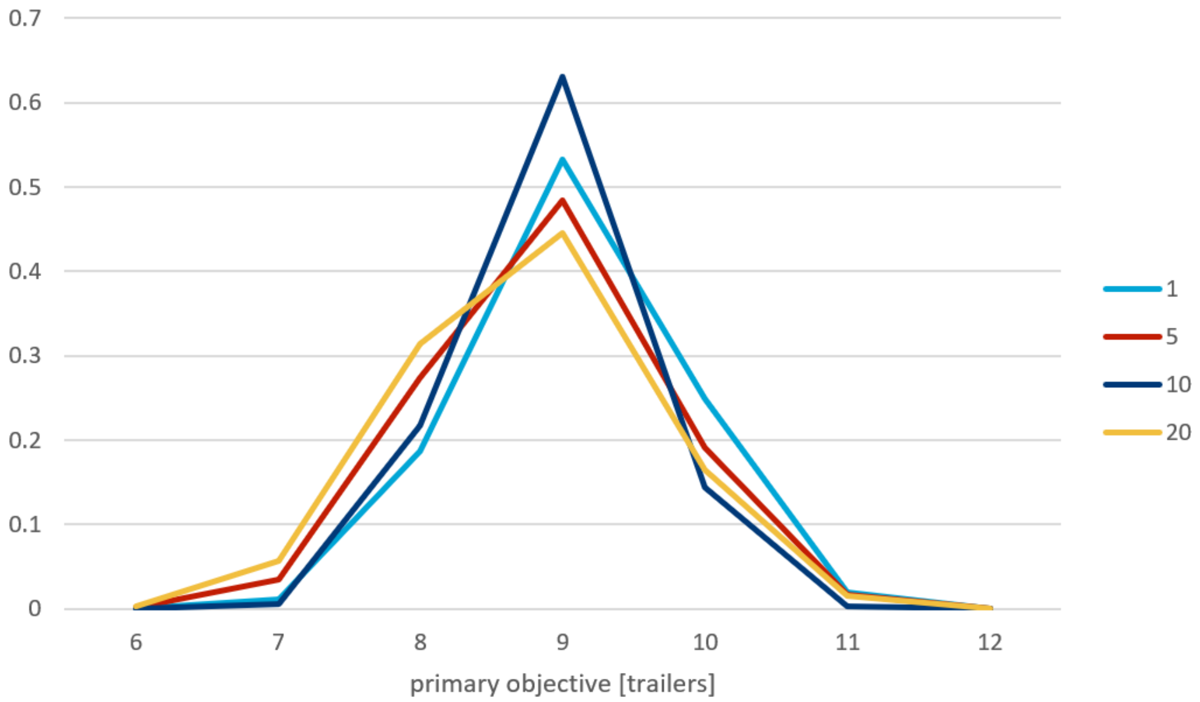
Figure 40: Experiment 3: Normal distributions for data set sample phase 202 - conservation BW60 for various numbers of items removed



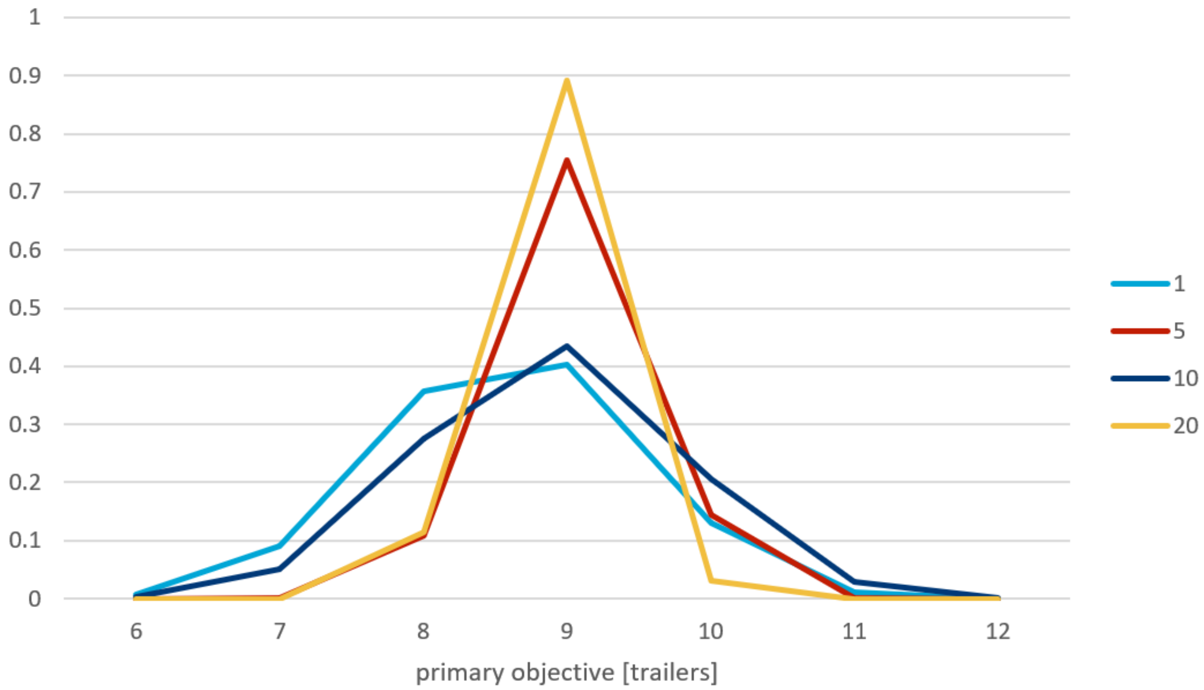
(a) 6 ALNS iterations

(b) 50 ALNS iterations

Figure 41: Experiment 3: Normal distributions for data set sample phase 305 - conservation T– for various numbers of items removed



(a) 6 ALNS iterations



(b) 50 ALNS iterations

Figure 42: Experiment 3: Normal distributions for data set sample phase 451 - conservation T– for various numbers of items removed

iterations ALNS = 6	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	6	7	6.64	0.625	11.65%	10.64%	3.02	0.861	3	0.627
5	6	6	6.36	0.557	13.04%	10.98%	3.02	0.799	3	0.546
10	5	6	6.36	0.557	12.20%	11.87%	3.26	0.712	3	0.449
20	5	6	6.36	0.557	16.29%	13.47%	3.04	0.603	3	0.557

iterations ALNS = 50	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	5	6	6.40	0.566	10.10%	11.82%	20.32	3.356	3	7.899
5	6	6	6.40	0.490	10.89%	9.40%	20.68	3.949	3	9.513
10	5	6	6.32	0.546	12.28%	11.66%	19.04	2.458	3	5.579
20	5	6	6.24	0.650	15.58%	15.37%	20.04	3.429	3	9.463

Table 15: Experiment 3: results sample phase 202 - conservation BW60

iterations ALNS = 6	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	5	6	5.96	0.599	7.23%	5.50%	1.62	0.372	3	0.480
5	5	6	6.20	0.566	6.30%	5.28%	1.65	0.415	3	0.449
10	5	6	6.12	0.431	5.37%	4.24%	1.72	0.447	3	0.496
20	5	6	6.12	0.515	6.05%	5.29%	1.71	0.369	3	0.490

iterations ALNS = 50	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	5	6	6.28	0.601	4.07%	2.73%	12.24	1.547	3	3.418
5	5	6	6.04	0.528	7.20%	5.44%	10.67	1.569	2	0.496
10	5	6	6.04	0.528	5.32%	4.17%	10.34	1.314	3	0.466
20	5	6	6.00	0.283	5.24%	3.52%	11.18	1.109	3	0.480

Table 16: Experiment 3: results sample phase 305 - conservation T–

iterations ALNS = 6	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	7	9	9.08	0.744	1.12%	1.12%	4.03	0.823	5	0.601
5	8	9	8.88	0.816	1.21%	1.20%	4.18	0.848	5	0.614
10	8	9	8.92	0.627	1.06%	0.93%	3.99	1.137	5	0.480
20	7	9	8.76	0.862	1.23%	1.18%	4.18	1.327	5	0.748

iterations ALNS = 50	primary objective [trailers]:				secondary objective [fill%*]:		time:		first best iteration:	
	best	mode	mean	std.p	mean	std.p	mean	std.p	mode	std.p
itemsremoved:	[trailers]	[trailers]	[trailers]	[trailers]	[fill%*]	[fill%*]	[s]	[s]	[-]	[-]
1	7	8	8.60	0.894	1.57%	1.54%	37.60	5.351	4	0.755
5	8	9	9.04	0.528	1.26%	1.32%	37.01	5.785	5	0.531
10	8	8	8.88	0.909	1.14%	0.87%	35.95	4.754	4	7.915
20	8	9	8.88	0.431	1.00%	0.66%	34.97	6.702	5	0.512

Table 17: Experiment 3: results sample phase 451 - conservation T–

When analyzing the mean and standard deviation values in tables 15 - 17, there seems to be little to no difference between the obtained solution data for the objective and computation time, which would mean the number of items removed in each iteration has no significant effect on the solution quality, nor the computation time. The graphs in figures 40 - 42 give the same idea. To investigate whether the obtained results are indeed not significantly different, the student t-test is used again. The different values for removed items for each data set sample and number of ALNS iterations are compared to each other in tables 18 - 23.

test:	t_exp:	t_th:	result:
1 vs. 5	1.6724	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 5 are significantly different
1 vs. 10	1.6724	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	1.6724	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 20 are significantly different
5 vs. 10	0	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	0	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 18: Student t-tests for sample phase 202 - conservation BW60 for itemsremoved = 1, 5, 10 and 20 and 6 ALNS iterations

test:	t_exp:	t_th:	result:
1 vs. 5	0	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 5 are significantly different
1 vs. 10	0.5090	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	0.9285	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 20 are significantly different
5 vs. 10	0.5455	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	0.9829	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0.4714	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 19: Student t-tests for sample phase 202 - conservation BW60 for itemsremoved = 1, 5, 10 and 20 and 50 ALNS iterations

test:	t_exp:	t_th:	result:
1 vs. 5	1.4569	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 5 are significantly different
1 vs. 10	1.0847	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	1.0127	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 20 are significantly different
5 vs. 10	0.5625	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	0.5227	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 20: Student t-tests for sample phase 305 - conservation T− for itemsremoved = 1, 5, 10 and 20 and 6 ALNS iterations

test:	t_exp:	t_th:	result:
1 vs. 5	1.500	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 5 are significantly different
1 vs. 10	1.500	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	2.1068	1.677	reject the null hypothesis: 1 and 20 are significantly different
5 vs. 10	0	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	0.3341	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0.3341	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 21: Student t-tests for sample phase 305 - conservation T− for itemsremoved = 1, 5, 10 and 20 and 50 ALNS iterations

test:	t_exp:	t_th:	result:
1 vs. 5	0.9057	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 5 are significantly different
1 vs. 10	0.8220	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	1.4055	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 20 are significantly different
5 vs. 10	0.1943	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	0.5056	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0.7506	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 22: Student t-tests for sample phase 451 - conservation T− for itemsremoved = 1, 5, 10 and 20 and 6 ALNS iterations

test:	t_exp:	t_th:	result:
1 vs. 5	2.1185	1.677	reject the null hypothesis: 1 and 5 are significantly different
1 vs. 10	1.0980	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 10 are significantly different
1 vs. 20	1.4102	1.677	not enough evidence to reject null hypothesis: no evidence that 1 and 20 are significantly different
5 vs. 10	0.7614	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 10 are significantly different
5 vs. 20	1.1744	1.677	not enough evidence to reject null hypothesis: no evidence that 5 and 20 are significantly different
10 vs. 20	0	1.677	not enough evidence to reject null hypothesis: no evidence that 10 and 20 are significantly different

Table 23: Student t-tests for sample phase 451 - conservation T− for itemsremoved = 1, 5, 10 and 20 and 50 ALNS iterations

Looking at the results for the student t-test, it occurs that for only two cases the null hypothesis could be rejected, meaning at a 95% confidence interval the obtained results are stated to be significantly different. However, the data may be biased, since the primary objective is an integer and the objective means include decimal numbers. This makes it easier to gain differences between tests, and may have lead to the two false negatives in the student t-test. Because the null hypothesis is rarely rejected, despite the bias, it can be concluded from these results that overall, there is no significant difference in the quality of the solution when varying the number of items removed during each iteration.
This conclusion strengthens the idea that the initial solution created by the ALNS algorithm is hard to improve by said algorithm. This could either mean that the initial solution already has a very good quality, or the ALNS heuristic itself is not perfect. The latter could possibly be solved by trying out additional ALNS removal and repair heuristics to allow for more strategies, but this is for future research.

6.5 Current solution versus proposed solution

In this paragraph, the best result from the proposed model of this thesis is compared to the actual loading division created at the company of ASK Romein.

The best number of trailers for the full data set presented in this report equals 90 trailers, obtained by various versions of the model, shown in tables 10 and 11. However, this solution cannot always be reached by the model.
Table 24 shows the results for 25 tests of the best version of the model, which is the model that creates a deterministic initial solution and runs for 4 ALNS iterations, with 10 removed items per iteration. As can be seen in this table, the maximum value for the primary objective is 92 trailers, which means that for any loading division created by this model, the solution will not consist more than 92 trailers. The average execution time for a single test is less than two minutes, of course excluding the time needed to import the items and the execution of the pallet item and merging function.

test nr:	best objective:	primary objective:	secondary objective:	execution time:
	[-]	[trailers]	[fill%*]	[s]
0	91.0872	91	0.0872	120.76
1	91.091	91	0.091	128.96
2	92.0875	92	0.0875	115.96
3	92.0872	92	0.0872	113.43
4	92.0872	92	0.0872	119.49
5	92.0873	92	0.0873	116.18
6	92.0875	92	0.0875	108.89
7	91.0909	91	0.0909	118.32
8	92.0875	92	0.0875	117.77
9	92.0875	92	0.0875	120.72
10	92.0872	92	0.0872	134.57
11	91.0915	91	0.0915	121.32
12	91.0909	91	0.0909	110.6
13	92.0875	92	0.0875	108.21
14	92.0875	92	0.0875	125.65
15	92.0875	92	0.0875	120.49
16	92.0872	92	0.0872	116.23
17	92.0875	92	0.0875	115.05
18	92.0875	92	0.0875	116.41
19	92.0872	92	0.0872	117.47
20	92.0875	92	0.0875	124.44
21	92.0875	92	0.0875	124.75
22	92.0875	92	0.0875	117.36
23	92.0875	92	0.0875	131.44
24	92.0875	92	0.0875	118.45
average:		91.80	0.087992	119.32

Table 24: Results for 25 tests with 4 ALNS iterations, a deterministic initial solution and 10 removed items per iteration

Now it is time to start comparing this value of 92 trailers to the actual loading division created at ASK Romein (table 25). The data set used for the experiments consists of 121 trailers. This means that the loading division can be reduced by at least 29 trailers or 24%, which is an enormous reduction.

	current approach:	proposed approach:
guaranteed solution:	121 trailers	92 trailers **(-24%)**
best solution:	121 trailers	90 trailers **(-26%)**
time needed	approx. multiple hours	<10 min
loading conditions	loading division	loading division and loading sequence

Table 25: Results for 25 tests with 4 ALNS iterations, a deterministic initial solution and 10 removed items per iteration

Currently, it is not exactly known how much time on creating the loading division is spent by everyone responsible. However, it costs at least several hours, as each item has to be assigned to a transport lot manually. Comparing this to the computation time of the proposed model, it is in the range of minutes, which is significantly faster. And even if the number of iterations of the model would be extended to a very large amount of iterations (even more than 50): the model does not require constant supervision. It only has to be initiated once and does all the work.
Furthermore, it should be noted that the proposed model is able to account for all loading conditions, both for the loading division process (such as maximum weight) as for the loading sequence process

(such as axle loads), whereas the current loading division is created with little attention to the loading sequence conditions. As mentioned earlier in this report in the problem definition, this can be the cause for costly last-minute alterations to the loading division, which is avoided with the new approach.

6.6 Discussion

The sub questions assigned to this section are as follows:

7. How do the parameters of the model influence the solution quality and computation time?

Sub question 7 is related to the sensitivity analysis: In the first experiment, it appeared that the number of ALNS iterations has little to no effect on the primary objective, and a large effect on the computation time: naturally, the more iterations, the higher the computation time. As a result, the proposed number of ALNS iterations is 4 for the deterministic approach, as this is the minimum number of iterations to guarantee a feasible solution. Even though the best reported solution of 90 trailers is reported for 5 iterations and not for 4 iterations, the student t-test showed that the results of 4 and 5 iterations cannot be proven to be significantly different in the 95% confidence interval. Therefore, there is a large possibility that the tests with 4 iterations did not reach the best solution by chance. Due to the superior computation time for 4 iterations compared to 5 iterations, the former is the best option. For the randomized approach, the minimum number of iterations that always results in a feasible solution equals 6.
As the second experiment has shown, the model that creates a deterministic initial solution is superior to the model with a randomized initial solution. Both approaches can reach the best reported solution of 90 trailers, but the objectives of the deterministic approach have a lower mean and standard deviation, making it a much more robust model.
The third experiment confirms that the model relies strongly on the initial solution, as no significant differences could be detected between tests with a different number of items removed in each iteration. The results of the third experiment are validated using the student t-test.

When comparing the results of the proposed approach to the current approach at ASK Romein, the following conclusions can be drawn:

1. The current loading division of the data set used for the validation could be reduced from 121 to 92 trailers or less, which is a reduction of at least 24%.

2. The computation time for the model to run is, even in the worst cases (with the highest number of iterations), far superior to the current time spent on creating the loading division. Furthermore, the proposed model can run without human supervision, which means no working hours are wasted during the execution.

3. The proposed model accounts for all loading conditions presented in section 2. For the current approach at ASK Romein, only the loading conditions related to the loading division can be checked. The proposed model, by digitally generating the loading sequence, is also able to check the loading sequence conditions. This can avoid unwanted surprises when the actual loading sequence takes place, such as for example an exceeded axle load.

The experimental phase concludes the research of this report. The next section will provide a conclusion as well as a list of recommendations.

7 Conclusion

In this report, a model is presented that automates and optimizes the loading processes for steel transport. The research revolves around two loading processes: the loading division, e.g. allocating items to different trailers, and the loading sequence, e.g. placing items onto the trailer.
The possibility of automating the process of loading division is investigated. Furthermore, the loading sequence process is generated digitally and implemented in the loading division process. This means that besides the loading division conditions, also the conditions for the loading sequence can be checked as early as the calculation and design phase.

Next, all sub questions presented in section 1 will be discussed.

1. What is the state-of-the-art of the loading processes at ASK Romein?

In section 2, insight has been given to the situation at ASK Romein, and the main problem is defined. Both transport-related processes, the loading division and loading sequence, use many conditions. Because of the many conditions and the fact that the loading sequence is not exactly known when creating the loading division, loading space is over- or underestimated.

The next sub question has been discussed in section 3:

2. What are the existing solutions of 3D space maximization problems?

The problem of this research can be defined to have an input minimization objective, fixed dimensions, weakly heterogeneous trailers and strongly heterogeneous items, making it a Multiple Bin Size Bin Packing Problem (MBSBPP). The accompanying packing strategy is the pre-assignment strategy, where first all items are distributed over the trailers (loading division), and next the items are placed on the trailers (loading sequence).
Several examples of solution methods from the literature have been discussed and evaluated in section 3: the ALNS heuristic appears to be the most promising loading division method, due to its flexibility and fast computation time. Regarding the loading sequence methods, none of the available methods truly fits the problem of this research: a gap exists in the literature, meaning a new heuristic is required.

In section 4, the proposed model was presented. Sub questions 3 and 4 have been answered in this section:

3. How can the loading division process be automated, accounting for the loading conditions?

4. How can the presented loading sequence problem be implemented in a 3D optimization model?

The provided model consists of an ALNS heuristic, that is responsible for the loading division, and the new layer heuristic, introduced in this report, that generates a loading sequence.
The ALNS heuristic uses several repair and removal heuristics to destroy and create new solutions, so that new options can be explored. The loading conditions are converted to model constraints and implemented in the model functions.
The layer heuristic is a revolutionary design in the sense that it provides a loading sequence solution for a real-world case without giving up competitive computation times. Based on the actual way the loading sequence is executed, by building layers of items on the trailer, the layer heuristic is a robust and fast solution method.

In section 5, the verification and validation have been given, so that sub questions 5 and 6 can be answered:

5. How can the loading division and loading sequence models be verified?

6. How can the loading division and loading sequence models be validated?

The verification tests of section 5 have all been passed by the proposed model and layer heuristic, which is a clear indication that they work as intended.
As mentioned before, analyzing the validation performance of the layer heuristic is more difficult, but can be executed by comparing its output to transport lots created in the traditional loading division. All tested transport lots resulted in feasible solutions for the loading sequence by the layer heuristic. For the full model, the results for the loading division are a significant improvement compared to the traditional loading division. Furthermore, the model is also capable of handling much larger data sets, albeit at the cost of larger computation times, thereby confirming the scalability of the model.

Finally, experiments were conducted on the proposed model, to provide an answer for sub question 7:

7. How do the parameters of the model influence the solution quality and computation time?

In section 6, various experiments have been conducted, with interesting results. The number of ALNS iterations has very little influence on the primary objective, and is proportionally related to the computation time. The former induces that the model has difficulty in improving the firstly created initial solution, possibly because the initial solution itself already has a good quality. The proposed number of ALNS iterations is the lowest number that guarantees a feasible solution.
The model that creates a deterministic initial solution has a better performance than the model with a randomized initial solution. Both are capable of achieving the best reported solution of 90 trailers, but the objectives of the deterministic approach have a lower mean and standard deviation, making it a much more robust model.
Finally, when comparing test results for several values of the number of items removed in each iteration, no significant differences are reported. This confirms that the model is very reliant on the first created initial solution, and that this initial solution is already of a very high quality.

Regarding the main research question of this thesis:

> How can the loading processes of trucks for steel structure transport be automated and optimized?

The automation is reached by the proposed model using an ALNS function, which is capable of optimizing the loading division and the layer heuristic, which generates the loading sequence. The layer heuristic fills a gap in the literature for more practical loading optimization problems. It provides a great answer to the complex problem of loading steel components and has a more than respectable computation time to work with.
To fully understand the performance of the proposed model, its results are compared to the current approach at the company of ASK Romein. This leads to three conclusions:

1. The model is capable of reducing the existing loading division by at least 29 trailers on a total of 121, which translates to a reduction of at least 24%.
2. The computation time for the model is in any case lower than the current time spent on creating the loading division. Besides, the proposed model can run without human supervision, which means no working hours are wasted during the execution.
3. The proposed model includes the loading conditions for the loading sequence process, instead of just the loading division conditions that can currently be checked when making the loading division, by digitally generating the loading sequence. Because of this, it can be secured that the loading sequence can be executed.

7.1 Recommendations

Throughout the assignment, several limitations of the current version of the model were discovered most of them could be fixed during the master thesis, but some of them can be investigated at a later point in time. These recommendations for future research are listed below:

- As the validation of section 6 shows, the ALNS algorithm has difficulties with improving the initial solution. This can be explained by the fact that the problem presented in this report contains many different constraints, which sometimes overlap. One idea to cover this is to keep track of feasibility flags for each trailer. For example, if for 5 continuous iterations the rear axle load always is the binding constraint, the algorithm could maybe act upon this by trying a different strategy for a new iteration.

- The previous recommendation is related to the next: it is mentioned that different strategies may be useful to create more possibilities. This can be achieved by designing more ALNS removal and repair heuristics, specified to assign a certain item to a certain trailer, according to which strategy is used. Some ideas for new heuristics:

  - A destroy heuristic that removes small items from trailers where the fifth wheel load constraint is violated (reducing the mass concentration at the front of the trailer) or removes large items from trailers where the rear axle load is exceeded (reducing mass concentration at the rear of the trailer).
  - A removal heuristic that counts the number of different item heights per trailer and removes items from trailers that have a large number of different heights. The item height defines the trailer layer, and more items of the same height easy the loading sequence process.
  - A repair heuristic that places items of similar weight on the same trailer, or a repair heuristic that places items of a similar length or width. A repair heuristic that concentrates on item height is already present in the current model.

- The current layer heuristic function, responsible for the loading sequence process, is already able to quite realistically simulate the actual loading sequence. However, it is still far from perfect. For example, for specific cases the rear end of the trailer may be underused. This could be solved by defining another set of layers placed in front of the existing layers, thereby considering contact between items in adjacent layers.
- It appears from the second experiment from section 6 that adding a randomizing factor increases the deviation in results. Although this was not beneficial for the ALNS heuristic, it may just be useful for the layer heuristic. If for example multiple iterations would be implemented each time the layer heuristic is initiated, and a small randomizing factor would continuously make small alterations, the chances for a feasible solution could increase due to an increasing solution space.

Besides these improvements on the model, for future research it would also be beneficial to conduct experiments on the model using different data sets, to test the robustness of the model.
Finally, to be able for the company of ASK Romein to use the model of this report, a software tool has to be created.

References

[1] Ana de Almeida and Marisa B Figueiredo. “A particular approach for the three-dimensional packing problem with additional constraints”. In: *Computers & Operations Research* 37.11 (2010), pp. 1968–1976.

[2] Sara Ceschia and Andrea Schaerf. “Local search for a multi-drop multi-container loading problem”. In: *Journal of Heuristics* 19.2 (2013), pp. 275–294.

[3] Teodor Gabriel Crainic, Guido Perboli, and Roberto Tadei. “TS2PACK: A two-level tabu search for the three-dimensional bin packing problem”. In: *European Journal of Operational Research* 195.3 (2009), pp. 744–760.

[4] Michael Eley. “A bottleneck assignment approach to the multiple container loading problem”. In: *OR Spectrum* 25.1 (2003), pp. 45–60.

[5] Michael Eley. “Solving container loading problems by block arrangement”. In: *European Journal of Operational Research* 141.2 (2002), pp. 393–409.

[6] Zhihong Jin, Takahiro Ito, and Katsuhisa Ohno. “The three-dimensional bin packing problem and its practical algorithm”. In: *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* 46.1 (2003), pp. 60–66.

[7] Aubrey L Mendelow. “Environmental Scanning–The Impact of the Stakeholder Concept”. In: (1981).

[8] Robert G Sargent. “Verification and validation of simulation models”. In: *Journal of simulation* 7.1 (2013), pp. 12–24.

[9] Gerhard Wäscher, Heike Haußner, and Holger Schumann. “An improved typology of cutting and packing problems”. In: *European journal of operational research* 183.3 (2007), pp. 1109–1130.

[10] Xiaozhou Zhao et al. “A comparative review of 3D container loading algorithms”. In: *International Transactions in Operational Research* 23.1-2 (2016), pp. 287–320.

[11] Ivan Žulj, Sergej Kramer, and Michael Schneider. “A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem”. In: *European Journal of Operational Research* 264.2 (2018), pp. 653–664.

Report number: 2021.MME.8540

An optimization model for 3D loading space for the transport of large steel structures

Master Thesis - Jasper Krombeen (4464745)

Delft University of Technology

Appendix

A Appendix

A.1 Research paper

The research paper for this report can be found on the next pages.

An optimization model for 3D loading space for the transport of large steel structures

Jasper Krombeen [1]

1. Delft University of Technology

Abstract

In this paper, a model is presented that automates and optimizes the loading processes for the transport of steel structures at the company of ASK Romein. This can be captured in two main loading processes: the loading division, which consists of allocating items to different trailers, and the loading sequence, which concerns the placement of items onto their respective trailer.

The possibility of automating the process of the loading division is investigated. This includes the digital generation of the loading sequence process. Achieving a form of automation would both mean a reduction in time required to create a loading division as well as allow for optimizing the number of required trailers.

Using an extensive literature review as well as an in-depth investigation of the current situation at the company, a model is developed, which consists of an ALNS heuristic responsible for the loading division, and a new function, the layer heuristic, proposed in this report, which digitally generates the loading sequence.

The proposed model is validated using real-life data. For the used data set, the model is able to reduce the existing loading division by at least 24%, and the computation time is far superior to the current time required to create a loading division. Because the loading sequence process is digitally generated, the model is capable of checking all loading conditions, such as axle loads, even before the actual loading sequence has taken place.

Keywords: 3D space optimization; Multiple Bin Size Bin Packing Problem; Adaptive Large Neighborhood Search

Introduction

The construction of large steel structures, such as football stadiums, distribution centers, and the increasingly popular data centers, requires many different types of steel assemblies, in all kinds of shapes. This makes the transport to site a complex procedure: loading a truck with steel assemblies is not as straightforward as loading a set of (rectangular-shaped) boxes.

When it comes to loading, there are two different processes distinguishable. First, the loading division process, executed during the calculation phase of a project, when all components and assemblies are assigned to a transport lot. Next is the loading sequence process, concerning the exact placement of the assemblies and components when placed onto a trailer. The loading sequence takes place after the production of the assemblies.

In the remainder of this section, a brief description of the problem will be given, after which the literature findings are described.

Characteristic:	Description:	Included in traditional loading division:	Included in proposed loading division:
		trailers	
1. Maximum weight	limit on overall truck weight	✓	✓
2. Axle and fifth wheel loads	limit on axle loads		✓
3. Length	multiple trailer sizes	✓	✓
4. Width, height	fixed trailer dimensions	✓	✓
		items	
5. Flatrack and trailer gap	smaller items can fall through trailer gap		✓
6. Item securing	restrict item movement		✓
7. Item stability	stacking items on larger other items		✓
8. Item orientations	fixed orientation for some items		✓
		processes	
9. Conservation finish	different finishes to different coaters	✓	✓
10. Erection sequence	limited storage on construction site	✓	✓
11. Loading and unloading	(un)loading can be horizontal or vertical		✓

Table 1: Model characteristics

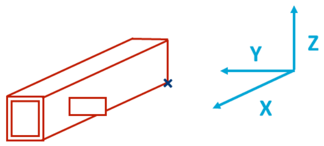
Currently, the loading division is a non-automated process. The design of a project is created in a building information model (BIM) environment. The different construction items present in the BIM model are manually assigned to a transport lot, which is a time-costly process. The assignment takes place conform certain loading conditions related to the loading division, such as for example the maximum weight.
Additionally, there are loading conditions related to the loading sequence, such as the maximum allowed front axle load. These requirements can only be checked when the loading sequence has taken place, and the orientation of the items on the trailer is known. Because the loading sequence is currently not known when creating the loading division, it has to be estimated or guessed. This however may sometimes lead to the over- or underestimation of the available loading space on certain trailers.

In this paper a solution in the form of a model will be presented that automates the loading division process, reducing the time needed, and digitally generates the loading sequence, guaranteeing successful loading and removing errors regarding over- and underestimation of loading space.
In table 1, the applicable loading conditions are converted to model characteristics. The table also shows which characteristics are currently included in the loading division process, and which characteristics are included in the proposed loading division.
Wässcher et al. [3] have described a large number of cutting and packing problems. They are grouped by their desired objective, dimensionality, characteristics of the items to be loaded, and characteristics of the loading objects (trailers). For the problem of this research, they are defined as input minimization, fixed dimensions, strongly heterogeneous items and weakly heterogeneous trailers, respectively. The related literature case is the Multiple Bin Size Bin Packing Problem (MBSBPP).

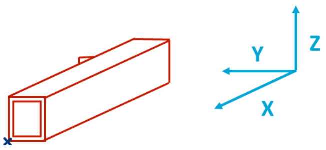
To construct the mathematical model, the decision variables presented by De Almeida and Figueiredo [1] and Jin et al. [2] are used. They are shown below.

$\mathrm{x}_{ik}, \mathrm{y}_{ik}, \mathrm{z}_{ik}$	x, y and z-coordinate of origin of sub item k of of item i
$\mathrm{lx1}_i, \mathrm{lx2}_i$	absolute orientation variables of item i
$\mathrm{a}_{ij}, \mathrm{b}_{ij}, \mathrm{c}_{ij}, \mathrm{d}_{ij}, \mathrm{e}_{ij}, \mathrm{f}_{ij}$	relative orientation variables of items i and j
X_{it}	check if item i is packed in trailer t
Y_t	check if trailer t is in solution

Variables $\mathrm{lx1}_i$ and $\mathrm{lx2}_i$ are an extension of lx_i for non-symmetrical items (figure 2). A schematic overview of the relative orientations is shown in figure 3.



(a) lx1 = 1



(b) lx2 = 1

Figure 2: Decision variables orientations

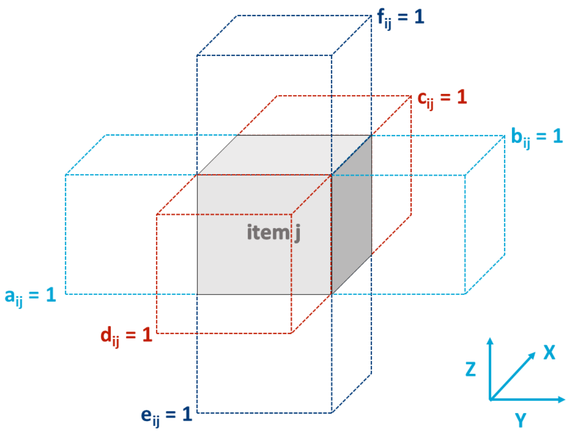


Figure 3: Orientation of item j with respect to item i, and corresponding decision variables

Proposed approach

The mathematical model is given in equations 1 - 11a.

$$\textbf{min } Z = \sum_{t}^{T} Y_t \tag{1}$$

subject to:

$$x_{ik} + l_{ik} + w\_ver \leq x_{jl} \tag{2a}$$

$$y_{ik} + w\_ver \leq y_{jl} - w_{jl} \tag{2b}$$

$$z_{ik} + h_{ik} + w\_hor \leq z_{jl} \tag{2c}$$

$$y_{ik} = y_{i0} + \left(lx1_i - lx2_i\right) * yk_{ik} \tag{3a}$$

$$x_{ik} = x_{i0} + \left(lx1_i - lx2_i\right) * xk_{ik} \tag{3b}$$

$$z_{ik} = x_{i0} + zk_{ik} \tag{3c}$$

$$0 \leq x_{ik} - lx2_i * l_{ik} \tag{4a}$$

$$x_{ik} + lx1_i * l_{ik} \leq L_t \tag{4b}$$

$$0 \leq y_{ik} - lx2_i * w_{ik} \tag{5a}$$

$$y_{ik} + lx1_i * w_{ik} \leq W_t \tag{5b}$$

$$0 \leq z_{ik} \tag{6a}$$

$$z_{ik} + h_{ik} \leq H_t \tag{6b}$$

$$\sum_{i}^{I} m_i \leq m_{\text{legislation}} \tag{7a}$$

$$\sum_{i}^{I} m_{i\_\text{bogie}} \leq m_{\text{legislation\_bogie}} \tag{7b}$$

$$\sum_{i}^{I} m_{i\_\text{fifthwheel}} \leq m_{\text{legislation\_fifthwheel}} \tag{7c}$$

$$\sum_{i}^{I} X_{it} \leq M * Y_j \tag{8}$$

$$Y_{it} * Y_{jt} * csrv_i = Y_{it} * Y_{jt} * csrv_j \tag{9}$$

$$Y_{it} * Y_{jt} * bphs_i = Y_{it} * Y_{jt} * bphs_j \tag{10}$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \geq 1 \tag{11a}$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \leq 3 \tag{11b}$$

$$x_{ik}, y_{ik}, z_{ik}, \text{integer} \tag{11c}$$

$$lx1_i, lx2_i, a_{ij} - f_{ij}, X_{it}, Y_t, \text{binary} \tag{11d}$$

To reduce complexity, overlap equations 2a - c, used to obtain the origin coordinates of the sub items, are a simplified version, depicting a specific orientation of two items. The complete equations applicable to all orientations can be found in the main report of this research. Equations 3a - c connect the sub items ($k > 0$) to their respective main items ($k = 0$). Equation pairs 4a - b, 5a - b, and 6a - b make sure that all items are placed completely within trailer bounds. The constraints 7a - c check the legislation regarding the maximum allowable load and the maximum axle loads and fifth wheel loads. Equation 8 adds a trailer to the solution when an item is packed to the trailer. Equations 9 and 10 state that each trailer only has one conservation system and one building phase.

Finally, equations 11a - d ensure the duality of the binary variables and that the overlap constraints are defined properly.

The presented mathematical model is unable to account for both the item stability and the item movement constraints. Besides, the presented constraint regarding the axle loads (equations 7b - c), cannot be captured by regular MIP solver software such as Gurobi, due to the presence of decision variables in the denominator. Furthermore, the computation time for a model of the scale of this research is expected to be unacceptably high. This stresses the need for a different approach, in the form of a heuristic model, which will be presented next.

Figure 4 shows a schematic of the main functions of the model. The output is shown at the bottom of the figure: a solution for the loading division. There are several steps required to get there.

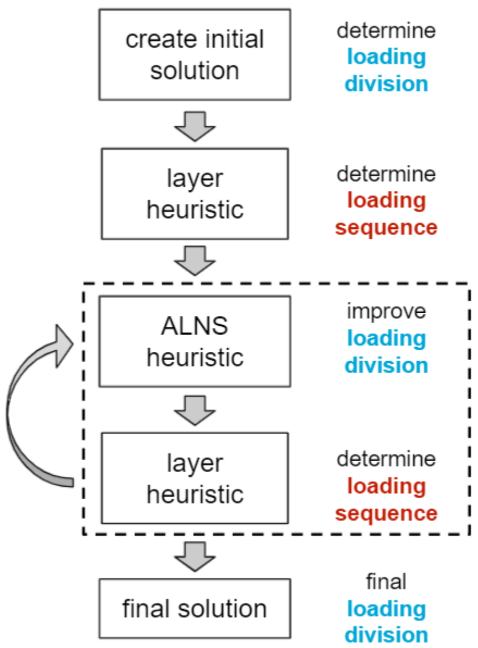


Figure 4: Model overview

As can be seen in the figure, the model consists of two main functions: the ALNS heuristic and the layer heuristic. Using the results of the first iteration for the loading division and loading sequence, a new iteration for the loading division is started. The loading division is improved by moving items between trailers and removing or adding trailers. Subsequently, the corresponding loading sequence is generated to see if the new loading division indeed is an improvement.

Next is the working principle of the model. When the model is initiated, first all the items are sorted by their conservation system and building phase. a separate ALNS heuristic function is executed for each unique conservation-phase combination. Throughout the iterations, the results for the loading division are evaluated against the current best solution. The solution is expressed in the form of the objective in equation 12:

$$\min Z = W_1(\text{nr. of trailers}) + W_2(\text{fill\%}^*) \quad (12)$$

In equation 12, W_1 and W_2 are weights ($W_1 = 1$ and $W_2 = 0.01$) and fill%* is defined as the filling percentage of the most empty trailer in the solution. Naturally, the number of trailers is the primary objective. This is always an integer (there are no half trailers), and every solution with less trailers than the current solution should always be defined as a better solution. The filling percentage is a secondary objective to help identify better solutions. The choice of weights means the primary objective is always prioritized over the secondary objective.
A new solution is only considered when it is feasible, that is, if all items could correctly be placed during the loading sequence. The ALNS heuristic uses its multiple removal and repair heuristics to destroy and create new solutions by swapping around items between trailers. The different heuristics have different traits that enhance the search for better solutions. At the end of each iteration, the loading sequence is generated using the layer heuristic.
The layer heuristic is designed to create a loading sequence according to the loading division presented by the ALNS function. Therefore, it receives the items allocated to a certain trailer as input, and returns the loading sequence for that trailer, similar to the example of figure 5.

The ALNS function and the layer heuristic continuously interact with each other, alternately creating a new loading division and its respective loading sequence. Finally, after the model has run for a certain number of iterations, the resulting best objective is collected.
When the ALNS heuristics for all conservation-phase combinations are finished, the total best

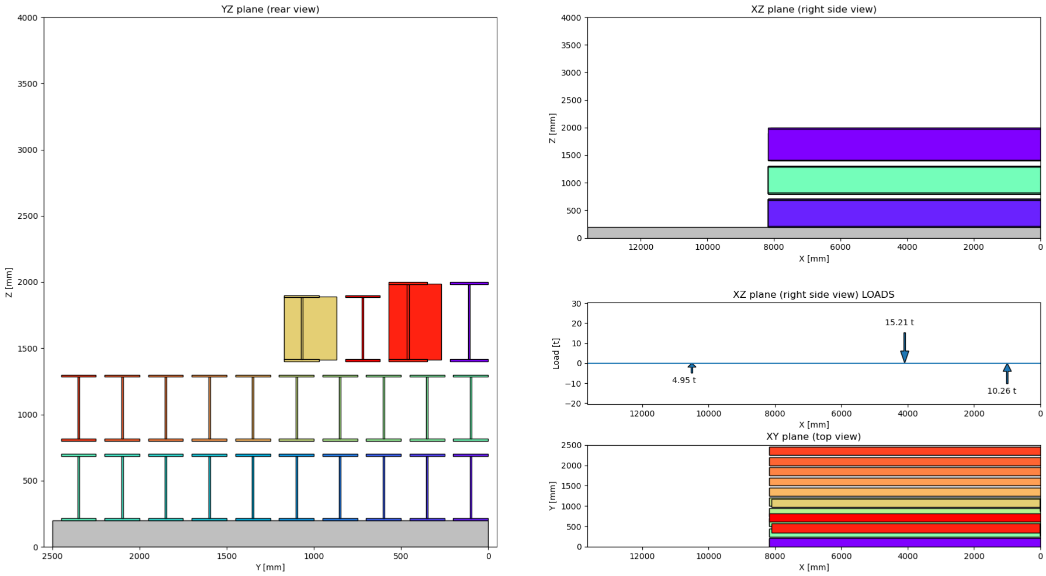


Figure 5: Loading sequence transport lot 331

objective can be calculated.
This best solution is then complemented by a pallet item function, that assigns all the items that are small enough to fit on a pallet to the loading division. The solution is improved once more by a merging function, that merges trailers with unique conservation-phase combinations and low filling rates, to further reduce the number of trailers.

Experimental results

In this paragraph, several experiments will be conducted in the form of sensitivity analyses. First, the best parameter values will be used to determine the performance of the model, which will be compared to the existing solution at ASK Romein. To secure the validity of the results, each test is conducted 25 times, thereby reducing the effects of random chance.

The case scenario used for the experiments is a project recently executed at ASK Romein. The project regards a data center consisting of over 7500 steel components. In the original loading division created for this project, a total of 121 transport lots were created, corresponding to 121 trailers.

The model performance is shown in (table 2). Since the model guarantees a solution consisting of only 92 trailers, this means that the traditional loading division can be reduced by at least 29 trailers or 24%, which is an enormous reduction. Furthermore, the computation time is far superior, and the model is capable to check all eleven loading conditions.

Three types of experiments are conducted using this data set. There are two versions of the model to be tested: one that produces a deterministic initial solution, and one with a slightly randomized initial solution.
For the first experiment, the effect of the number of iterations for each ALNS heuristic on the computation time and solution quality is tested.
The goal of this test is to find a trade-off between the solution quality and the computation time. Figures 6 and 7 show the experimental results.

In figure 6, the primary objective is shown for tests with various iterations. As can be seen in the figure, for the deterministic approach, using less than 4 iterations results in unfeasible solutions. For all feasible results, there is no significant difference in the solution quality. For the randomized initial solution approach, tests with 6 or more iterations result in a feasible solution. Again, there is no significant difference in solution quality between feasible solutions.
Figure 7 shows there is a significant difference in

	current approach:	proposed approach:
guaranteed solution:	121 trailers	92 trailers **(-24%)**
best solution:	121 trailers	90 trailers **(-26%)**
time needed	approx. multiple hours	<10 min
loading conditions	loading division	loading division and loading sequence

Table 2: Results for 25 tests with 4 ALNS iterations, a deterministic initial solution and 10 removed items per iteration

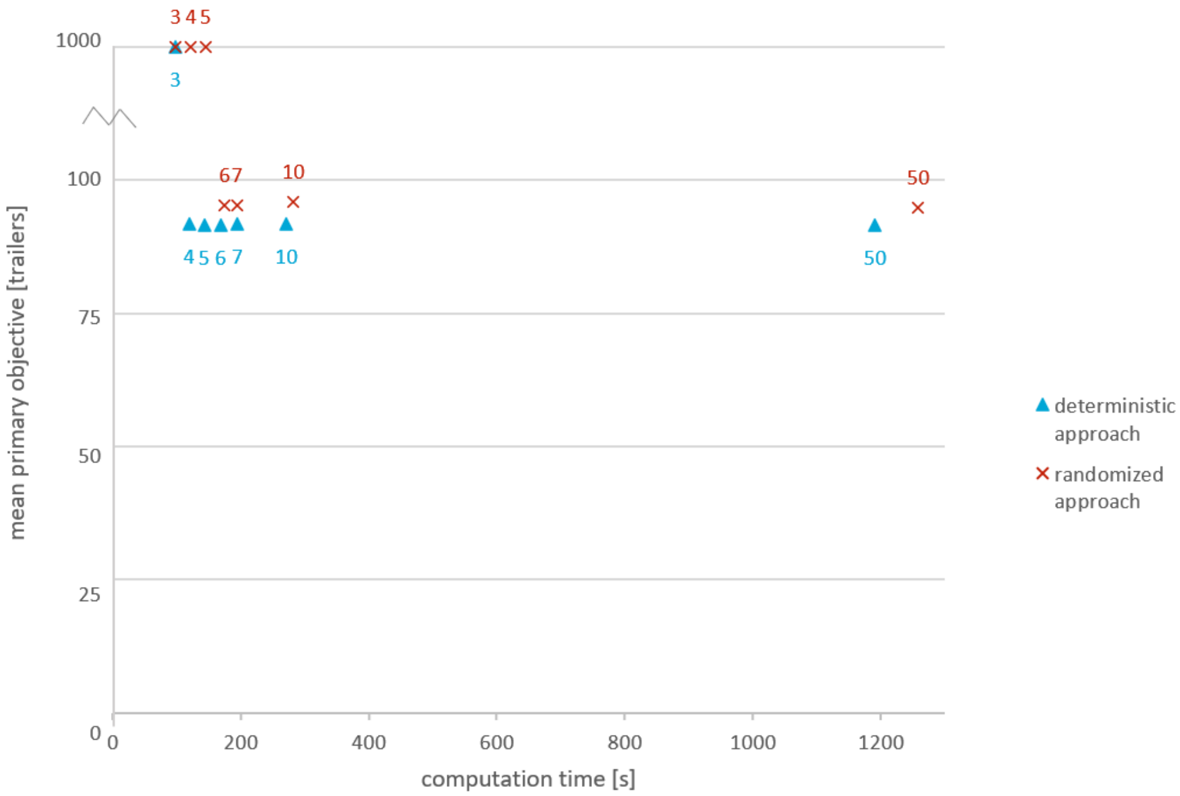


Figure 6: Experiment 1: primary objective versus computation times for various ALNS iterations

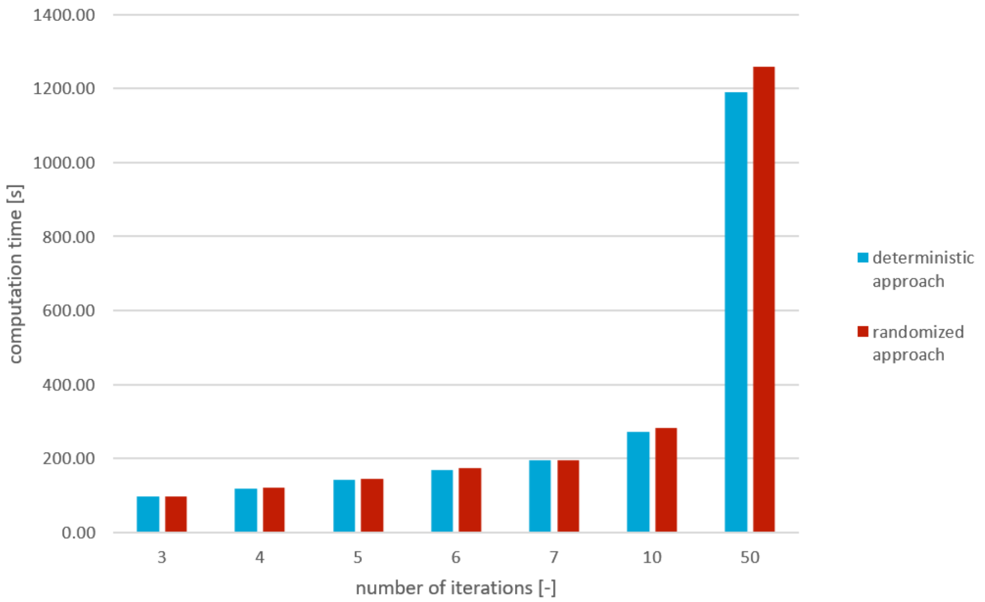
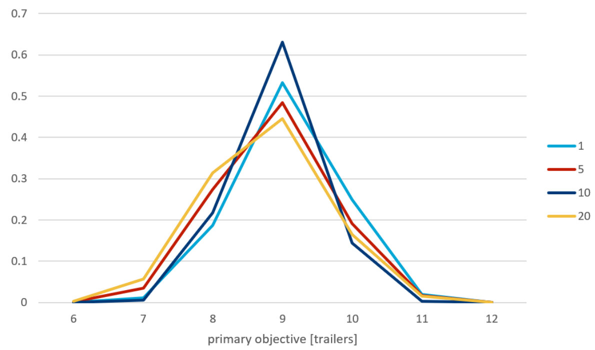


Figure 7: Experiment 1: computation times for various ALNS iterations

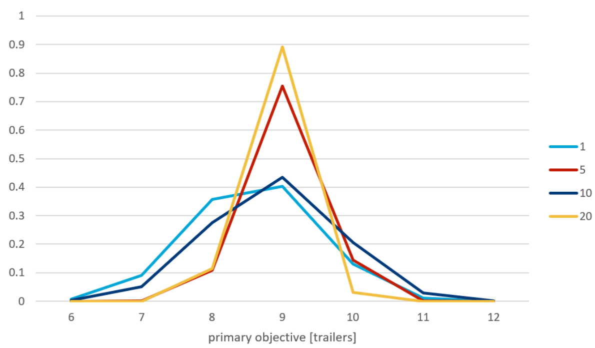
the computation time, which seems to be proportional to the number of iterations, as can be expected.

The second experiment tests the effect of randomizing the initial solution. The original initial solution heuristic does not contain any randomness factor. This means, that for each new experiment, the same initial solution is generated. Using the original, deterministic, initial solution heuristic, most of the time leads to exactly the same results, even though the improvement heuristics do contain a randomness factor. This observation induces the possibility that the model may get stuck in a local optimum.
Therefore, a slightly altered initial solution heuristic, containing a randomness factor, is tested against the original initial solution heuristic.
Figure 8 shows that both approaches are able to report the best solution of 90 trailers, but the model with the deterministic approach guarantees a solution of 92 trailers and has a lower standard deviation and mean, which makes it superior.

The third test focuses on another parameter, the number of items to be removed during each ALNS iteration. This experiment is designed to find the best value for this parameter.
To reduce testing computation times, instead of the full data set, samples of the data set are used. The samples are chosen carefully, so that they strongly represent the full data set. Figure 9 shows the normal distribution results for one of the samples. It can be observed that varying the number of items removed has no effect.



(a) 6 ALNS iterations



(b) 50 ALNS iterations

Figure 9: Experiment 3: Normal distributions for data set sample phase 451 - conservation T– for various numbers of items removed

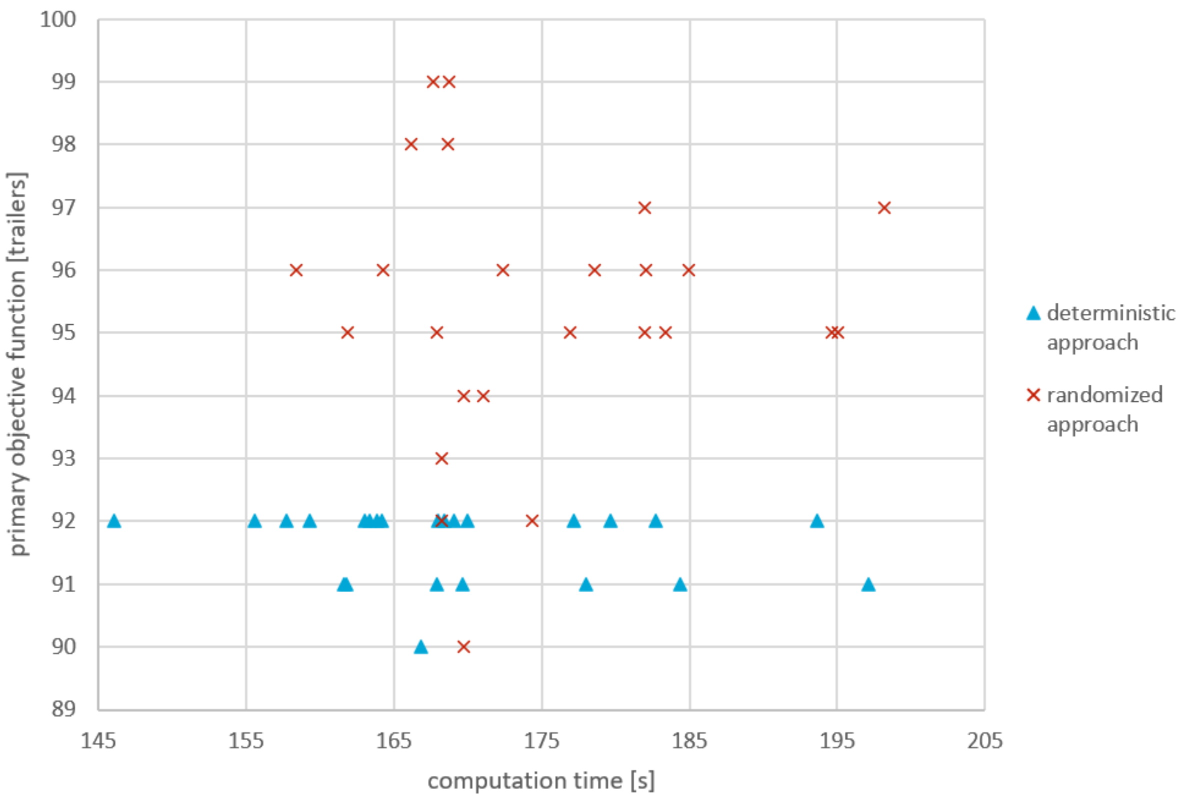


Figure 8: Experiment 2: primary objective function vs computation time for 6 ALNS iterations

Discussion

In this report, a model is presented that automates and optimizes the loading processes for steel transport. The research revolves around two loading processes: the loading division, e.g. allocating items to different trailers, and the loading sequence, e.g. placing items onto the trailer.
The possibility of automating the process of loading division is investigated. Furthermore, the loading sequence process is generated digitally and implemented in the loading division process. This means that besides the loading division conditions, also the conditions for the loading sequence can be checked as early as the calculation and design phase.
The presented model consists of an ALNS heuristic, that is responsible for the loading division, and the new layer heuristic, introduced in this report, that generates a loading sequence.

Looking at the experiments, the number of ALNS iterations has very little influence on the primary objective, and is proportionally related to the computation time. The former induces that the model has difficulty in improving the firstly created initial solution, possibly because the initial solution itself already has a good quality. The proposed number of ALNS iterations (four) is the lowest number that guarantees a feasible solution. The model that creates a deterministic initial solution has a better performance than the model with a randomized initial solution. Both are capable of achieving the best reported solution of 90 trailers, but the objectives of the deterministic approach have a lower mean and standard deviation, making it a much more robust model.
Finally, when comparing test results for several values of the number of items removed in each iteration, no significant differences are reported. This confirms that the model is very reliant on the first created initial solution, and that this initial solution is already of a very high quality.

The layer heuristic fills a gap in the literature for more practical loading optimization problems. It provides a revolutionary answer to the complex problem of loading steel components and has a more than respectable computation time.

To give a good insight in the performance of the proposed model, its results are compared to the current approach at the company of ASK Romein. This leads to three conclusions:

1. The model is capable of reducing the existing loading division by at least 29 trailers on a total of 121, which translates to a reduction of at least 24%.

2. The computation time for the model is in any case lower than the current time spent on creating the loading division. Besides, the proposed model can run without human supervision, which means no working hours are wasted during the execution.

3. The proposed model includes the loading conditions for the loading sequence process, instead of just the loading division conditions that can currently be checked when making the loading division, by digitally generating the loading sequence. Because of this, it can be secured that the loading sequence can be executed.

This master thesis is executed at the company ASK Romein.

References

[1] Ana de Almeida and Marisa B Figueiredo. "A particular approach for the three-dimensional packing problem with additional constraints". In: *Computers & Operations Research* 37.11 (2010), pp. 1968–1976.

[2] Zhihong Jin, Takahiro Ito, and Katsuhisa Ohno. "The three-dimensional bin packing problem and its practical algorithm". In: *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* 46.1 (2003), pp. 60–66.

[3] Gerhard Wäscher, Heike Haußner, and Holger Schumann. "An improved typology of cutting and packing problems". In: *European journal of operational research* 183.3 (2007), pp. 1109–1130.

List of parameters

l_{ik}	length of sub item k of item i $0 \leq i \leq$ I, $0 \leq k \leq$ K
w_{ik}	width of sub item k of item i
h_{ik}	height of sub item k of item i
xk_{ik}	relative distance between the x-coordinates of the origins of main item i and its sub item k
yk_{ik}	relative distance between the y-coordinates of the origins of main item i and its sub item k
zk_{ik}	relative distance between the z-coordinates of the origins of main item i and its sub item k
cmx_i	x-position center of mass of item i (including sub items)
m_i	mass of item i (including sub items)
L_t	trailer length
W_t	trailer width
H_t	trailer height
w_ver	vertical stoppage wood thickness
w_hor	horizontal stoppage wood thickness
Bx_t	distance between trailer front and bogies of trailer t
Fx_t	distance between trailer front and fifth wheel of trailer t
M_t	empty mass of trailer t
Mflatrack	flatrack mass
$mb_{\text{legislation}}$	allowable rear bogie axle load
$mf_{\text{legislation}}$	allowable fifth wheel load
$m_{\text{legislation}}$	allowable item load
$csrv_i$	conservation system of item i
$bphs_i$	building phase of item i
M	large number

A.2 Model structure

In this part of the appendix the documentation for the model can be found. The two most important functions of the model, the ALNS and layer heuristic, are more thoroughly explained. To fully understand the way the model works, figure 43 shows the structure of the model, which consists of multiple layers that are all interacting with each other.

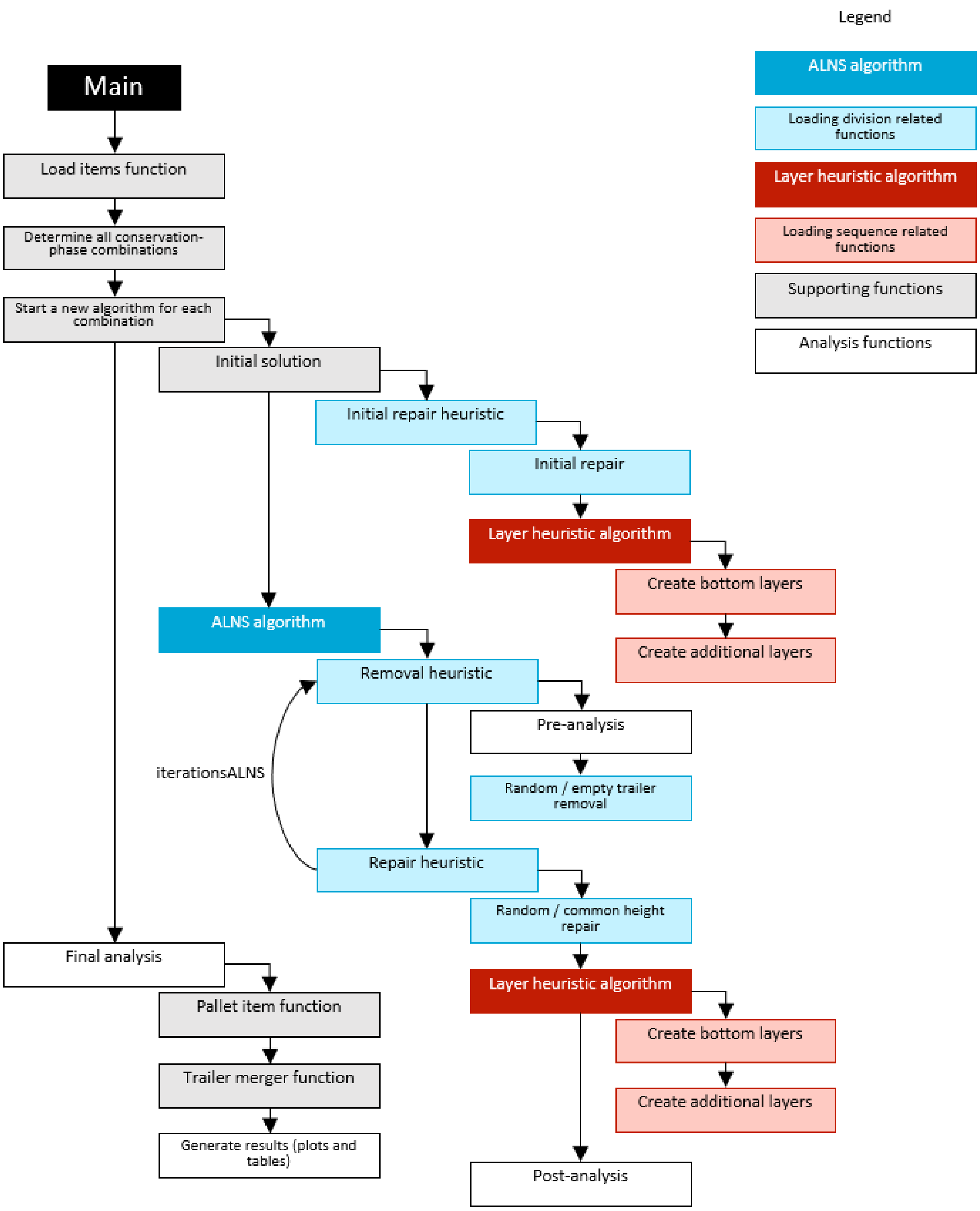


Figure 43: Algorithm structure

In the remainder of this subsection, the most important functions present in figure 43 will be discussed.

The first layer of the model is the main file. Here all input parameters are generated. Furthermore, all other functions and heuristics are initiated from this file.
The main file contains a function to load in all the items of the data set into the python environment. This function has the capability to load only samples of the full data set, such as one specific building conservation-phase combination, or one main building phase. The output of this function is a list

containing all normal items and a list containing all pallet items, the items that are small enough to be transported on pallets. These pallet items will not be considered during the main process of the model, but will be assigned to trailers in the closing stages, as will be described later in this subsection.

The next function determines all conservation-phase combinations that are present in the given data set. It uses the previously mentioned item lists as input and returns a list containing conservation systems and dictionaries containing the respective lists of phases ordered by conservation system.

Next, a new function is started for each phase-conservation combination. This function creates an initial solution as well as a new ALNS heuristic to improve the initial solution for a fixed number of iterations.
The initial solution is created by a special repair heuristic designed specifically for this task. This heuristic, 'initial repair heuristic', is a first greedy heuristic in the sense that it takes the first item from the list and places it in the first trailer, until the trailer is full, after which a new trailer is added. When the repair heuristic is finished, the layer heuristic is executed for the first time, to determine the loading sequence of each created trailer, and see whether space is over- or underestimated.

Next, the ALNS heuristic is initiated. this heuristic destroys and repairs the current solution for a fixed number of iterations, depicted in figure 43 as 'iterationsALNS'.
For each new iteration, a removal heuristic is picked to destroy the current solution. Part of this removal heuristic is an analysis function that checks whether the current solution is feasible or not. There are two options:

1. The current solution is not feasible: the removal heuristic will only remove the items that could not be placed in the previous iteration.
2. The current solution is feasible: the removal heuristic will remove a number of items dependent on the type of heuristic.

There are two types of removal heuristics. The first is called 'random removal heuristic', and, as the name suggests, randomly removes a number of items from random trailers. The second type is called 'empty trailer removal heuristic', and removes items from the most empty trailer, so that maybe this trailer can be fully emptied in future iterations.

When the removal heuristic is finished, the repair heuristic takes over to re-arrange the removed items. There are two types of repair heuristics as well. Again, the first is random repair, which randomly finds a feasible trailer to place an item in. The second is called 'common height repair heuristic', and tries to place an item in a trailer that has multiple items of similar height. This can be beneficial, because on the trailer, layers are created consisting of items with identical heights.
At the end of each repair heuristic, the layer heuristic is called again to determine the loading sequence corresponding to the new solution. Next, an analysis function is initiated to determine the next steps and to save intermediate results.

When all building phase - conservation combinations have run the required number of iterations, the model is close to the end. Only the final analysis has to be executed.
The first step of the final analysis is to combine the results from all different combinations into one solution, so that the final number of trailers can be known. The next step is to add all items that are small enough to be placed on a pallet with other small items. These items are added to trailers that still have not met their maximum weight and axle loads.
The next step is to merge some trailers that only have a few items. These are the trailers that have a similar conservation system and main building phase (main building phase of phase 204 is 2, for example). If all items from two of these trailers can be placed on one trailer, the merger is successful.

Finally, the final solution is generated and plots and tables are exported.