Master's degree thesis

LOG950 Logistics

A heuristic approach to the three-dimensional bin packing problem with weight constraints

Knut Olav Brathaug Sørset

Andres Haugen Vikhagen

Number of pages including this page: 89

Molde, 05/24/2019

Molde University College
Specialized University in Logistics

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. Failure to complete the statement does not excuse students from their responsibility.

*Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.*		
1.	**I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than mentioned in the paper/assignment.**	☒
2.	**I/we hereby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☒ 2. ☒ 3. ☒ 4. ☒ 5. ☒
3.	I am/we are aware **that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations** section 14 and 15.	☒
4.	**I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am/we are aware that Molde University College will handle all cases of suspected cheating according to prevailing guidelines.**	☒
6.	**I/we are aware of the University College's rules and regulation for using sources**	☒

Personal protection

Personal Data Act

Research projects that processes personal data according to Personal Data Act, should be notified to Data Protection Services (NSD) for consideration.

Have the research project been considered by NSD? ☐yes ☒no

- If yes:

Reference number:

- If no:

I/we hereby declare that the thesis does not contain personal data according to Personal Data Act.: ☒

Act on Medical and Health Research

If the research project is effected by the regulations decided in Act on Medical and Health Research (the Health Research Act), it must be approved in advance by the Regional Committee for Medical and Health Research Ethic (REK) in your region.

Has the research project been considered by REK? ☐yes ☒no

- If yes:

Reference number:

Publication agreement

ECTS credits: 30

Supervisor: Lars Magnus Hvattum

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).

All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).

Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒yes ☐no

Is there an agreement of confidentiality? ☐yes ☒no

(A supplementary confidentiality agreement must be filled in)

- If yes:

Can the thesis be online published when the period of confidentiality is expired? ☐yes ☐no

Date: 05/24/2019

Høgskolen i Molde

AGREEMENT

This agreement is between

Andres Haugen Vikhagen and Knut Olav Brathaug Sørset (students),

Lars Magnus Hvattum (Faculty Advisor at Molde University College),

Driw AS v Pål Haugen (Company/Institution), and Molde

University College/MSc Logistics Program Coordinator,

concerning the use of specifications and results reported in the Master's degree thesis in accordance with the study plan for the Master's degree program in Logistics at Molde University College.

I . The student will complete the work assigned for the Master's degree thesis in cooperation with the company/institution•.................Driw AS...

The working title of the thesis is: Plasticity in the constraints of a four-dimensional bin pack algorithm

2. The student has copyrights to the thesis. Those copies of the thesis submitted for evaluation along with descriptions and models, such as computer software that is included as part of or as an attachment to the thesis, belongs to Molde University College. The thesis and its attachments can be used by the College for teaching and research purposes without charge. The thesis and its attachments must not be used for other purposes.

3. The student has the right to publish the thesis, or parts of it, as an independent study or as part of a larger work, or in popularized form in any publication.

4. The company/institution has the right to receive a copy of the thesis with attachments, and the College's evaluation of it. The company/institution will have three (3) months from the time the thesis is submitted to the College for grading to determine whether a patent is possible and to apply for a patent for all or part of the results in the thesis. The specifications and results in the thesis can be used by the company/institution in its own activities.

5. Data related to the project shall be stored on secure locations and should not be made available to others than the participants in the project. The data used in the project are not to be considered as a part of the thesis. The data are not allowed to accompany the submission and cannot be a part of any copyrights. After the thesis is approved all data must be securely deleted from the computer where they are stored. All data must be stored on a password protected unit.
The evaluation committee can be granted access to the data if this is needed for the evaluation of the thesis. The committee must request data access from Driw AS.

6. An additional confidentiality agreement may be entered into between the parties, if the company/institution sees this as necessary.

7. Each part of the agreement should have one copy of the agreement. The final signature should be from the Program Coordinator/Dean validating the agreement.

Place Molde / Ålesund Date of final signature 10.01.19

Anders H Vestgarden

Knut Sandvik

Student(s)

Faculty Advisor

Company/Institution

Program Coordinator / Dean

Preface

This thesis marks the end of our time at Molde University College. LOG950 is a continuation of the course LOG904-100 and concludes our Master of Science in Logistics degree.

The goal with the thesis was to make a heuristic approach for the three-dimensional bin packing problem with weight constraints and to see if it could compete with experienced workers when it comes to efficiency.

Many thanks to Pål Haugen at Driw, who came up with the topic and provided us with the data used in the experiments conducted.

We also want to thank Ann Lisbet Brathaug for proofreading.

We send gratitude to our supervisor Professor of Quantitative Logistics at Molde University College Lars Magnus Hvattum, for feedback and guidance throughout the research and writing process.

Summary

This thesis has focused on the implementation of heuristics to the three-dimensional packing problem with weight constraints to see how the usage of box weight as a decision variable influence the packing of pallets. The problem considered has implemented heuristic selection procedures to check if the packing based on weight and height can compete with the packing conducted by experienced workers. This thesis describes how a mixed integer mathematical model, with and without modifications, can be used to find possible packing possibilities. Two different heuristic approaches are created and presented; the first is an approach that locks each of the coordinates for already packed boxes, while, the second is a construction heuristic that uses weight, and later the height, as the box selection criteria. The models and heuristics are applied to a real-life packing problem and compared to real-life solutions, as well as to each other. The results show that the approaches tested do not manage to compete with the experienced workers and are far from managing to outperform them.

Table of Contents

List of Figures and Tables

1. Introduction

This idea behind the research conducted in this thesis and the data used is provided by a company situated in Ålesund, Driw. Driw provides companies with systems and solutions to get full control and tracking of the flow of goods within the supply chain. In their search to improve their product and knowledge of packing, a three-dimensional packing problem with weight restriction created by Yang (2016) was discovered. This problem led to the idea of a four-dimensional packing problem with weight as the fourth dimension. The question Driw wants to figure out is if implementing the weight of the boxes not only regarding the total weight of the packed boxes but also independent from each other can create better packing solutions than what the workers are capable of at the moment. They will also want to know if the introduction of plasticity in the constraints can create better packing solutions.

Combinatorial optimisation has since it achieved foothold, almost 70 years ago, become one of the driving forces in the field of discrete mathematics. A lot of the problems considered in combinatorial optimisation is concerned with the organisation and use of scarce resources to create better efficiency and improve upon productivity (Nemhauser and Wolsey 1988). The problem is often formulated using graphs and integer or mixed-integer linear models. The field spans such problems as machine scheduling, vehicle routing, facility location and bin packing, expressing the field's relevance towards real-life problems and issues. (Korte and Vygen 2006)

One of the most extensive and researched fields within combinatorial optimisation is the field concerning bin packing. Not only is bin packing one of the most frequently researched fields in combinatorial optimisation, but it is also one of the problems with the most practical relevance, with Korte and Vygen (2006) stating, *"There are not many combinatorial optimization problems whose practical relevance is more obvious"* (2006, pp.425). Bin packing problems can be seen used in problems across a vast number of different industries, some of which being shipping, air cargo and vehicle routing — helping bin packing become one of the most complex and challenging problems in combinatorial optimisation.

Bin packing can be described as a problem where a given number of items, mainly boxes, are going to be packed into a finite number of bins. The objective is to pack these boxes in the best possible way so that as few as possible bins are used, either by maximising the volume utilised or by minimising the empty space in the bin. Most packing situations concerns the packing of items given by their volume, three-dimensional packing, i.e. by the dimensions given by the box's length, height and width. None of the boxes can overlap, so not to occupy the same volume. The weight of the boxes can also be considered when packing. Most bins and containers have a specific restriction on how much weight that can be packed onto it. The restriction makes sure that even though there might be more room for another box, this will not be utilised if the weight restriction is met. Another bin must then be used to pack the box.

In its raw sense, bin packing can come across as a simple enough concept; pack as many boxes as possible into as few as possible bins. Although bin packing may come across as straight forward, the repercussions of inadequate packing can have severe consequences, much in the same way that good packing can have an opposite effect. First and foremost, proper packing is both necessary and essential when concerned with the environment, health and safety (EHS). Unstable packing can cause items to fall, and as a result, there might occur injuries to the workers taking part in the packing, loading and unloading.

Secondly, good packing can prevent damages done to the goods that are to be packed, as well as equipment and the means of transportation. Deformation and damages to the goods caused by inadequate packing can cost a company not only money in terms of paying for damages and multiple deliveries of the same order, but it can also cost the company the entire client. Damage can also be inflicted to the equipment used in the packing and transportation of the goods, as well as to the bin or container itself. This is not only regarding the boxes packed being too heavy, but also where these boxes are placed inside the bin or container, i.e. the weight distribution. Vehicles, mainly larger trucks, used for transporting goods can be damaged during the transportation if too much weight is put on top of the front- and the rear axle, causing the axles to bend or break (Alonso et al. 2017). Pallets that are made of either wood or plastic can break when too much pressure is placed on specific areas. Also, cranes and other equipment can be damaged by the bins or the containers being too heavy, or by a skew weight distribution.

Thirdly, packing can be time-consuming. Proper packing is essential to better the efficiency of both the packing and the unloading procedure (Haessler and Brian Talbot 1990). As cliché as it may seem, Benjamin Franklin's famous quote "*Remember that Time is Money*"

(Founders online, *Benjamin Franklin 1748*, pp.1) is something that is well documented in bin packing- and container loading problems (Monaco, Sammarra, and Sorrentino 2014). Time becomes a factor both regarding faster packing as well as regarding stable and non-damaging packing, excluding the need for several deliveries for the same order. Also, the packing patterns and the order the items are loaded play an essential part when discussing efficiency, mainly concerning the loading and unloading procedures. These problems are often faced in routing problems with loading and unloading constraints, seen in seaborne-, rail-, and road transportation. The packing order is key for efficiency in these problems so that unnecessary items do not have to be unloaded and then loaded back in.

Finally, stable and well-balanced packing can lower fuel consumption, which further leads to less emission. The pursuit of more environmentally friendly ways of transportation is more important today than it has ever been before. It is no longer enough for companies to complete deliveries; these deliveries should also be done in such a way that emissions are as low as possible. Better weight-distribution and loading leads to better cargo stability, which, according to Amiouny et al. (1992) lead to fuel savings.

All these distinctive attributes contribute to the vastness and complexity that is bin packing. Even though many aspects are taken into consideration in bin packing problems and packing problems nowadays are more frequently modelled towards representing real-life problems, many of the aspects are tackled separately. Some problems are concerned with stability, others with weight distribution and fragility, and so forth. This thesis will consider all the different aspects that arise, including total weight, weight distribution, stability and load balancing

The main objective of this thesis is to pack a finite number of boxes onto as few as possible pallets, by implementing the volume and the weight of the boxes both regarding the weight restriction of the pallet as well as independent from each other. The idea stems from the problem tackled by Yang (2016). Just as with many packing problems, a finite number of boxes where to be packed into a finite number of bins with different dimensions. Instead of only focusing on the volume of the boxes, Yang also considered the weights, creating something he called a four-dimensional packing problem. Since the weight of the boxes was only considered in terms of bin weight and not independent from each other, it did not have any effect on the packing patterns or the order the boxes were packed. Causing the problem to be more similar to a three-dimensional packing problem with weight constraint, then a four-dimensional problem.

This thesis considers a model created based on Paquay, Schyns, and Limbourg (2016), with added modifications. The model considers a packing-problem where a given set of boxes, an order of items, are to be packed onto a finite number of same size pallets. The packing is based on the length, height, width and weight of the boxes, and constraints concerning stability, load balancing, overlapping and size restriction is implemented. This model can only solve small problems. A test problem will be implemented to check the validity of the model. This problem will be solvable for the model and show that the restrictions and constraints implemented have the desired effect.

Another packing possibility that will be considered is to pack the pallet bit by bit, locking the coordinates of boxes that have been packed. When the coordinates are locked, the model packs a new set of boxes, but the coordinates of the boxes already packed are unavailable for the new boxes — creating fewer packing possibilities, the more packs that have been packed. If there are no suitable places for a box to be placed, a new pallet is made available.

A construction heuristic is implemented to solve larger problems. The heuristic divides the pallets into sections, making for a smaller problem size. A modified model based on Paquay, Schyns, and Limbourg (2016) is used with the heuristic. When all the boxes in an order are packed, based on the weight of the boxes, each of the packed sections is defined as a box, and the model is implemented to pack these sections, ending up with packed pallets. Plasticity will be introduced to the solution created by the heuristic to see if minor changes to the constraints used can have an impact on the packing.

Changing the heuristic to choose the set of boxes based on height instead of weight will also be considered, although in a smaller example than the weight-based heuristic. This approach is implemented to see what disregarding weight can do to the packing and the volume utilisation.

The thesis has the following structure: Section 2 is devoted to the literature surrounding bin-packing, describing different applications of models in the vast research field, mostly concerned with three-dimensional packing problems and the implementations of stability, weight distribution and load balancing in container loading problems. Section 3 consists of a description of the problem itself. Section 4 describes the models used and the constraints and parameters used in the modelling procedure. Section 5 is dedicated to the description of the construction heuristic created to solve larger instances of the problem. Section 6 creates a detailed overview of the computational results that have been found. Packing from both the modified and the non-modified models by Paquay, Schyns, and Limbourg (2016) will be presented, both regarding the construction heuristic and the maximum fill approach. Section 7 includes the concluding remarks to the thesis, the limitations and some ideas of what can be done in future research.

2. Literature Review

Since bin packing is one of the largest research fields in combinatorial optimisation and includes a vast array of different problems, all having different parameters and dimensions, massive amounts of literature are available on the subject. Extending from one- dimensional problems to the even more complex three-dimensional problems with weight-, stability- and loading constraints, and spanning everything from sheer bin packing problems to vehicle routing problems with loading constraints: The literature and research available are extensive and complex.

There are primarily four different problems the field of bin packing focuses on one-dimensional, two-dimensional, three-dimensional and multi-dimensional problems. The dimensionality describes the geometry of the items that are to be packed and therefore, has a significant impact on the packing patterns of the items. This thesis considers a three-dimensional bin packing problem with weight constraints.

Multi-dimensional problems also occur frequently in the literature and research regarding combinatorial optimisation. These problems differ from the four-dimensional problems by packing items of d-dimensions into a bin or container of d-dimensions and concerns the problem of maximising the number of vectors possible to pack into a bin or container of a specific fixed size (Chekuri and Khanna 2004). Multi-dimensional problems are often used in the field of scheduling and resource allocation (Bansal, Caprara, and Sviridenk, 2006). One particular form of the multi-dimensional problem is the vector bin packing problem. This problem is described by Chekuri and Khanna (2004) as *"The vector bin packing problem, on the other hand, seeks to minimize the number of bins needed to schedule all n-tasks such that the maximum load on any dimension across all bins is bounded by a fixed quantity, say,1."* (2004, pp. 1). The research and literature in this thesis explore three-dimensional problems across the research field, including bin packing, container loading and vehicle routing, with the inclusion of load balancing-, stability- and fragility constraints.

2.1 Bin Packing

In literature, bin packing problems are known under a vast number of different names, including knapsack problem, cutting problem, strip packing, vector packing. All of these problems are very similar, with subtle differences in figure shape, quantity measurement, packing patterns, bin size and more. (Delorme, Iori, and Martello 2018).

The bin packing problem is categorised as an NP-hard problem in combinatorial optimisation. This means that the problem is at least as hard as an NP problem and that the existence of an algorithm that can solve it to optimality in a reasonable amount of time is unlikely to exist (Tsai, Malstrom, and Kuo 1991). Since methods of optimally solving significant bin packing problems in a reasonable amount of time may not exist, a lot of the research on solving bin packing problems have been centred on heuristics and approximation algorithms. In other words, the research focuses on finding near-optimal solutions. Even though approximation algorithms and heuristics may not find the optimal solution, it comes up with solutions that guarantee to be in a specific percentage range of the optimal, in the fraction of the time.

Given the extensive and complex research and literature, different reviews and typologies of bin packing problems have been published. This to better understand and categorise the problem itself, as well as creating an overview of the literature on the subject. One of the first typologies of cutting and packing problems was issued by Dyckhoff (1990). The paper created one of the first expansive overviews of the research field, by both diving into the characteristics that are important to look at in packing and creating small reviews of the literature that exists on the different subjects and dimensionalities. Dyckhoff stated that *"The purpose is to unify the different use of notions in the literature and to concentrate further research on special types of problems".* (1990, pp.145).

Wäscher, Haußner, and Schumann (2007) later updated and improved upon Dyckhoff's typology. They stated that the ever-growing literature on the subject, mainly the significant increase in the last two decades, had made the previous typology a bit outdated. They specified that a new typology would give a fresh and new look on the research field, while also keeping the notion that the new typology would help concentrate future research on specific varieties of the bin packing problem. Their main task consisted of modifying the definitions and characteristics created by Dyckhoff, while also defining basic, intermediate and refined problem types, further developing and updating the vast problem library. The new typology also showed that problems could be solved across different problem categories, stating that not all cutting and packing problems could be designated to a specific problem type, presenting a vaster perspective on the packing and cutting problems.

In later research and since its inception, researchers have emphasised on three-dimensional packing problems, mainly given its greater relatability to real-life problems and complex nature. In three-dimensional packing problems, an assumption that the boxes are rectangular is made. The goal is to pack the items orthogonally into as few as possible identical bins. Even though the boxes are assumed to be rectangular, the packing problem can also include cuboids and cylinders, although these are less researched and less frequent.

Hifi et al. (2010) introduced a mixed integer linear programming model (MILP1) to tackle the three-dimensional packing problem. The objective was to minimise the number of identical bins. They gave each of the possible bins a label, number, and minimised the highest label of the used bin. The model considered overlapping constraints, guaranteeing that no items occupy the same volume in the bin. The model managed to obtain feasible solutions in a reasonable amount of time for instances with up to 60 items. They introduced lower bounds that were centred on the solving of LP-relaxation of the model, where the trivial lower bound was the optimal value of the LP-relaxation of the model. Given the fact that the trivial lower bound is weak, the conducting of a branch and bound approach to the solving of the model would have included a large number of additional tree nodes — further leading to non-feasible solutions in a reasonable time. To obtain feasibly, and time relative, solutions to larger problems, they introduced lower bound constraints that were based on valid inequalities found in parallel-machine scheduling problems. With the inclusion of these inequality constraints, the model managed to yield satisfactory results to the packing problem.

2.1.1 Pallet Packing Problem

Pallet packing problems are less researched than its more known brethren bin- and container packing. The pallet packing problem aims to pack as many items as possible into the dimensions provided by a pallet. Contrary to bins and containers, pallets do not have solid walls. This means that there is less lateral support for the boxes packed then what is seen in container loading problems. The lack of solid walls may, however, come with some advantages, one of which is the possibility of introducing plasticity to the constraints.

Plasticity is a term often used in mathematics and medicine. In mathematics, plasticity is the research of how plastically deformed solids, such as metals, can withstand stress and strain (Hill 1998). In medicine, plasticity is commonly used in the terminology of neural plasticity. Neural plasticity is described by Kaas (2001) as *"neural plasticity refers to the ability of neurons to change in form and function in response to alterations in their environment"* (2001, pp.10542). The principals behind these descriptions of plasticity can be used in pallet packing. Where containers have solid walls, pallets only have natural base areas. By, ever so slightly, increasing the pallet length and/or width, new packing possibilities occur letting the edges of the boxes to be put a little outside the pallet walls.

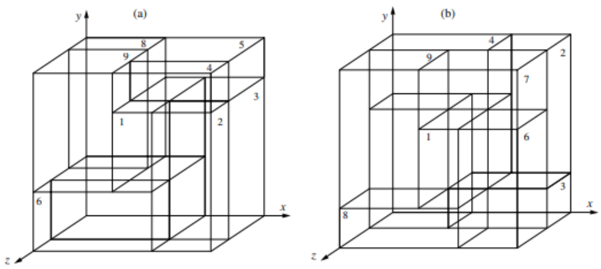
Bischoff, Janetz, and Ratcliff (1995) focused on internal stability in their research of packing. They concentrated on what is called the "Distributor's Pallet Packing Problem", referring it to as *"where items of varying sizes – representing an order for a customer – have to be loaded onto a pallet"* (1995, p.681). The objective is to maximise the volume of the boxes packed onto the pallet. In contrast, "Manufacturer's Pallet Packing Problem" packs identical boxes in layers, each box having the same vertical orientation, creating a possibility of decreasing the problem from a three-dimensional problem into a two-dimensional. Even though the pallet problem, especially the distribution pallet packing problem, have not been prioritised in research or literature, a lot of what can be found in other bin packing problems and container loading can be implemented and vice versa. With the problems having almost all the same variables, parameters and constraints, differing only in aspects such as support, size and heterogeneous vs homogenous boxes.

2.1.2 Corner point and robot packing

Martello, Pisinger, and Vigo (2000) modelled a problem where a given number of three-dimensional rectangular items was to be packed into an unlimited number of identical three-dimensional rectangular bins. The problem did not allow for rotation, and the packing was conducted orthogonally, meaning that each of the edges of the package is parallel to the bin edge it corresponds to, I.e. that the packages are perpendicular to each other. By incorporating an original approximation algorithm, an algorithm for filling a single bin was developed. This algorithm led to the definition of an exact branch-and-bound algorithm for the three-dimensional bin packing problem. The paper was to the knowledge of the authors the first publication of a test instance for the three-dimensional problem.

Den Boef et al. (2005) later showed that the branch-and-bound approach by Martello, Pisinger, and Vigo (2000) was not able to obtain all the possible orthogonal packing possibilities. Instead of Martello, Pisinger and Vigo’s use of orthogonal packing, den Boef et al. (2005)refer to the packing from the branch-and-bound approach as robot packing. Den Boef et al. describes robot packing. as “*A robot packing is a packing that can be achieved by successively placing items starting from the bottom-left-behind corner, and such that each item is in front of, right of, or above each of the previously placed items.”* (2005, pp.735)

Figure 1 shows the differences between the two packing methods. The bins can house the same number of packages; the only difference here is that the two bins are packed differently. The packing solution computed in the orthogonal packing (b) is not possible for a robot to pack. Because this orthogonal packing possibility was not discovered in the initial report, Martello et al. (2007) extended the algorithm developed Martello, Pisinger and Vigo (2000) so that it can solve general problems as well as robotic. The new algorithm got the best results when the boxes were sorted according to non-increasing volume, and later branched on the different pair of boxes. This implementation made infeasible solutions to be detected quickly, given the fact that the relative positioning of the largest boxes could be established early on.



i	w_i	h_i	d_i	x_i	y_i	z_i
1	2	4	9	4	3	0
2	3	7	3	6	0	6
3	3	7	6	6	0	0
4	5	2	3	4	7	6
5	5	2	6	4	7	0
6	6	3	3	0	0	6
7	6	3	6	0	0	0
8	4	6	3	0	3	0
9	4	6	6	0	3	3

i	w_i	h_i	d_i	x_i	y_i	z_i
1	2	4	9	4	2	0
2	3	7	4	6	2	0
3	5	2	4	4	0	0
4	6	3	4	0	6	0
5	4	6	4	0	0	0
6	3	6	5	6	0	4
7	5	3	5	4	6	4
8	6	2	5	0	0	4
9	4	7	5	0	2	4

Note. The tables give the dimensions (w_j, h_j, d_j) and coordinates (x_j, y_j, z_j) of the items. The bin size is $9 \times 9 \times 9$.

Figure 1. Robot packing and orthogonal packing (den Boef et al., 2005, 736)

2.1.3 Extreme point-based heuristic

The solver by Yang (2016) is based on the extreme point-based heuristic. This heuristic was developed by Crainic, Perboli, and Tadei (2008) and could be implemented on three-dimensional as well as two-dimensional bin packing problems. The heuristic is constructed in a way that will yield a more efficient and accurate procedure for placing packages. Extreme point heuristic extends Martello, Pisinger and Vigo (2000) research on corner points described above. The idea of extreme point-based heuristic is that the placement of a three-dimensional package *k* with its left-back-down corner in a particular position in a three-dimensional bin will then create a lot of new points where other items can be placed. This is showed in Figure 2(a) being three-dimensional, and 2(b) being two-dimensional. While Figure 3 depicts the same items only with the corner points highlighted.

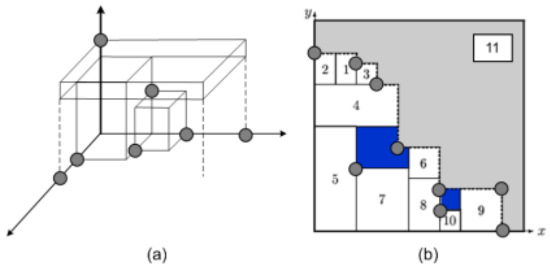


Figure 2. Extreme point (Crainic, Perboli and Tadei, 2007, 3)

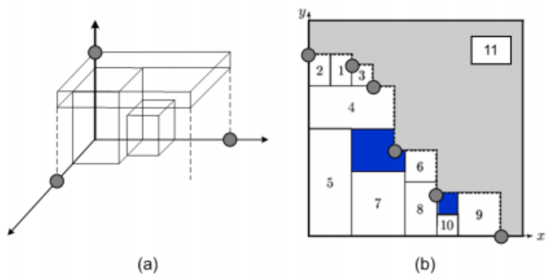


Figure 3. Corner Point (Crainic, Perboli and Tadei, 2007, 6)

Based on Crainic, Perboli, and Tadei (2008), this is the best constructive heuristic for the three-dimensional bin packing problem. What is also quite interesting is that it is claimed that the model is easily transferable to a problem, including package rotation. By merely duplicating each item once for each of the possible rotations, while also adding a constraint on what is called the mutual exclusion of the duplicates, the model would work without any computational effort, by only applying minor alterations.

2.2 Container Loading

Together with bin packing, much time and effort have been devoted to the research of container loading problems. While there are many similarities between the two fields of packing, container loading problems are more concerned with the load stability and the balance of the particular stacking patterns. Given the larger quantity and weight of the packages, imbalance and inadequate stability in container loading can have severe consequences on material and equipment, as well as life. Container loading problems are not limited only to the pure packing of containers to be put either on ships or trucks. The problem description also includes the packing of smaller vehicles and plane cargo, further adding complexity to the constraints and parameters that occur within each problem. Making the models for the container loading problems applicable to a vast arrange of different packing situations.

Even though much research in combinatorial optimisation focuses on container loading problems, and have done so for many years, Bischoff and Ratcliff (1995) indicated early on that a lot of the methodologies, and existing approaches lacked the applicability to real-life scenarios. They stated that some scenarios lacked methodologies for solving, thus having valid approaches for only a limited number of problems. Their paper was created to highlight the limitations existing in the research. By addressing the issues faced, two approaches and a broader typology of container loading problems were created.

As with both Dyckhoff (1990) and Wäscher, Haußner, and Schumann (2007), Bischoff and Ratcliff created a description of the criteria of the different problem types, i.e. the functional requirements they thought to be the most important. These requirements included the well-known orientation constraint seen in the two bin-packing typologies, but this is also where the similarities end. Given the sheer size and weight difference between containers and pallets/bins, bad or inadequate packing can have catastrophic consequences. Meaning that the importance of stable packing is of a more pressing concern in container loading research compared to other bin packing problems.

Based on Bischoff and Ratcliff (1995), Bortfeldt and Wäscher (2013) created a review of the constraints that exist and should be included in container loading. They stated that one of the problems with the research on container loading problems was its lack of compatibility towards problems faced in real-life scenarios, mainly because the constraints did not mimic the real-life constraints faced in practice. The creation of an overview of container loading constraints that were more targeted toward real-life problems would solve a lot of the issues faced in container loading.

Although Bischoff and Ratcliff (1995) put a focus on how narrow the researched problem types in container loading was and how it should be a goal to expand the research field, Bortfeldt and Wäscher (2013), almost two decades later, found that the same issues still existed. In their review, they stated that only about 4 out of 14 problems mentioned by Bischoff and Ratcliff (1995) are focused on in the research and that the only notable extensions in the literature are the joined loading and routing problems. Also, when looking at the issues with the lack of applicable real-life constraints, the research is scarce. Orientation and stability constraints being the only constraints frequently brought up in the research. They further stated that the modelling approaches to the numerous container problems are limited, having issues with solving problems with multiple constraints that need to be satisfied at the same time.

Models that could solve small container loading problems have existed for a long time. These models have mostly disregarded practically-relevant constraints, and have sought out to only pack as much as possible. Chen, Lee, and Shen (1995) considered a standard container loading problem where the boxes are to be packed orthogonally into a bin, minimising the empty volume inside. They focused on a mathematical formulation, creating a zero-one mixed integer programming model that came up with a good solution to small scale problems with few items. Along with the solution, they considered other applications for the model, applications that involved the solving of different sub-problems that exist in container loading, with or without modifications to the model. One application being the solving of a container loading problem with weight distribution. The modification would include the inclusion of a weight parameter on the boxes as well as a weight imbalance limit that is tolerable along the x-axis. Two constraints would then be added to regulate this imbalance.

A particular case of the container loading problem was considered by Sheng et al. (2016). In their paper, they focused on a container loading problem for pallets with infill boxes (CLPIB). The main focus of this problem was to load a set of pallets, packed with identical rectangular boxes, into a container. When no more pallets could be loaded into the container, boxes were taken from other pallets and put in between the gaps that the pallets had created, better utilising the volume of the container. Later, when the unloading procedure sets in, these boxes need to be repacked onto their initial pallets. While each of the boxes could be placed in any of the six orthogonal orientations, the pallets needed to be packed with their bottoms parallel to the container floor while also being fully supported by either the floor of the container or another pallet. They applied a heuristic approach in the form of a tree search- and greedy sub-algorithms, tree search sub-algorithm conducting the pallet packing and the greedy sub-algorithm conducting the filling of the residual space with the individual boxes. These heuristic approaches, combined with the guarantee that the infill boxes all can adequately pack a pallet, created a solution that is competitive given the conditions suggested.

2.2.1 Load balancing and stability constraints

Bin packing problems have evolved immensely since the start of the 1970s. Since the focus on heuristics emerged in this period, many different types of heuristics have been implemented in the research of bin packing. Liu et al. (2008) mention next-fit, first-fit, best-fit and bottom-left-fill as heuristics that have been often used in the solving of bin packing problems.

> Heuristic approaches are particularly useful for problems with a high complexity, for which deterministic methods like the branch and bound approach are often unable to find the solution within a reasonable amount of time. Although heuristics are fast in generating a solution packing plan, the quality of the solution is highly dependent on the input sequence of items (Liu et al. 2008, p.358).

Further, Liu et al. (2008) discuss one big issue of bin packing problems and how such problems are solved, the almost total focus on minimising unused space and number of bins. The objective of the bin packing problem is to pack the items in as few as possible bins or containers. This means that it almost disregards everything else, including load balancing and the stability of the packed bins, pallets or containers. The importance of a stable and well balanced packed bin has been previously stated by Amiouny et al. (1992) as both a safety measure and an implementation to save fuel and cost. They considered a problem concerning the loading of either a truck or an aircraft, where the objective was to create a heuristic that managed to pack a given set of boxes in such a way that the centre of gravity (CG) met a given target point, subsequently allowing for a packed bin that was well-balanced. The heuristic either came up with a different packing pattern creating better solutions or created no new solutions.

Supplementing Amiouny et al. (1992) research of air cargo packing concerning stability, Paquay, Schyns, and Limbourg (2016) developed a mixed integer mathematical model to solve the three-dimensional bin packing problem in which strongly heterogeneous boxes are to be packed into air cargo containers of different sizes and shapes, minimizing the empty volume of the container. Their approach included constraints applicable in real-life scenarios, such as orientation/rotation, fragility, load balancing, vertical stability and centre of gravity. Contrary to a standard container loading problem, air cargo container loading sometimes includes a particular type of container called a ULD. These containers are truncated parallelepipeds, i.e. one or two walls of the container are not perpendicular to one another. Such containers add to the complexity of the model by having slanted walls, requiring the boxes to satisfy certain constraints related to cuts in the bin. No conclusions were made based on the computational results the model produced. Paquay, Schyns, and Limbourg suggest that further research should concentrate on appropriate techniques and methods that combine exact algorithms and heuristics.

Additional research of load-balancing in bin packing has been conducted by Trivella and Pisinger (2016) Their paper considered a multi-dimensional version of bin packing with an extension of load balancing. The model made it possible to find optimal solutions to smaller cases of the problem. The inclusion of the load balancing constraints, the barycentre (geometric centre) of the boxes and the whole bin, and the mass/weight of the boxes, further extended and complicated the model, making it even more complex. Whereas the models by Chen, Lee, and Shen (1995) and Hifi et al. (2010) managed to pack up to 60 items in a reasonable amount of time, the extended load balancing multi-dimensional model suggested by Trivella and Pisinger (2016) had hardships with problems where more than 20 items were to be packed. Because of the complex nature of the extended packing problem, heuristic approaches were conducted to solve larger problems. The heuristic approaches included a multi-level local search, which then included a graph characteristic of the packing. Some straightforward approaches to balancing and local searches on transitive orientation was also implemented. Their experiment was able to find an effective load balancing. The heuristic ran for about 3-5 minutes for cases with 200 items, which makes the algorithm useful in real life logistical instances.

Ramos, Silva, and Oliveira (2018) developed a new load balancing methodology for the container loading problems concerned with road transportation. The new methodology was created based on, what they described as, simplified and soft constraints regarding load balancing. As seen in the research by Paquay, Schyns, and Limbourg (2016) and Trivella and Pisinger (2016) the barycentre, or centre of gravity, has been stated to be the geographical centre of the container or the geographical centre of the boxes. Ramos, Silva, and Oliveira (2018) state that this characteristic of the centre of gravity does not meet any of the regulations and legislations met in real-life scenarios. To meet these legislations and regulations, the load stability constraints are treated as hard constraints, found by defining the feasible area of the centre of gravity derived from the specifics of the vehicles. Another aspect of the paper is concerned with the weight the front- and rear axle of the vehicle can withstand. To tackle both the aspect of centre of gravity, regarding its distance from both the sidewalls and the front/back of the container, and axel strength, a load distribution diagram (LDD) is created. The LDD is described as;

> The LDD is a two-dimensional plot that shows the maximum admissible load of a vehicle as a function of the longitudinal or transversal position of its centre of gravity, thus defining the area where the location of the cargo CG is acceptable (Ramos, Silva, and Oliveira, 2018, p.1141).

Very good results are obtained with the two algorithms proposed by Ramos, Silva, and Oliveira (2018).

2.2.2 Fragility and load bearing constraints

A close relative to the load balancing and stability constraints, fragility and load bearing constraints are implemented so that items intended to be packed are not getting damaged or deformed, mainly in regards of not putting too much weight on top of certain items. Bischoff (2006) created a new heuristic approach for tackling container loading problems where several items had limited load bearing strength. The problem considered was a three-dimensional packing problem with weakly heterogeneous boxes intended to be packed into a single container of known dimensions. The focus when dealing with fragility and limited load-bearing strength is the rules regarding the placement of the items. To solve the problem, they created a new construction heuristic. The implementation of this construction heuristic created better volume utilisation solutions than what could be found in earlier work by Bischoff and Ratcliff (1995).

Bortfeldt (2012) considered a capacitated vehicle routing problem with three-dimensional loading constraints. The problem description included the dimensions (length, width and height) of the items as well as for the vehicle. The loading space in the vehicle and the items were all rectangular, with the vehicle having a given weight limit it can carry. Five vital packing constraints were added to the model, unloading-, weight-, orientation-, support- and stacking constraints. The unloading constraint stated that it must be possible to unload all the boxes at a customer i, as well as stating that a box demanded by a customer that is to be visited later than customer i cannot be placed above a box demanded by said customer i. 90 degree orientations were allowed, and the weight of all the packed boxes could not exceed the weight limit for the vehicle. Stability was insured by a constraint stating that if a box is not located on the floor, a certain percent of the bottom of the box needed to be supported by another box. As for the stacking constraint, which considered the fragility of the boxes, a specific fragility flag was given to all of the boxes labelling them either fragile or not. This constraint stated that only fragile boxes could be packed on top of other fragile boxes, while both fragile and non-fragile boxes could be packed on top of non-fragile boxes. To solve the problem, a hybrid algorithm was implemented. A tabu search algorithm created the routes for the vehicles, while a tree search algorithm, using the extreme point-based heuristic for its placement, was implemented for the packing of the boxes. The hybrid algorithm turned out to be superior to other methods used in similar problems, both in solution quality and computation time.

Closely related to the work done by Paquay, Schyns, and Limbourg (2016), Junqueira, Morabito, and Sato Yamashita (2012) also considered a three-dimensional packing problem where practical constraints such as stability and fragility were implemented. Distinguishing the two papers was Junqueira, Morabito, and Sato Yamashita (2012) inclusion of horizontal stability, which means that not only the bottom face of the boxes needed to be supported but the lateral sides as well. To tackle the question concerning fragility, constraints that stated that box i can only have a certain number of boxes on top of itself were implemented to avoid deformation and damages to the boxes. Even though this constraint is stated in the beginning of the paper, it is later reviled that the constraint states that no box can be put on top of a box that is categorized as fragile. 0-1 integer linear programming models were created and managed to solve moderate sized problems. The model considered four different types of boxes, one type having the ability to rotate around any of the axis and the other three having fixed rotations. Cubic containers were used to simplify the problem. They stated that the models managed to reflect the practical situations considered, but that further research should focus on methods such as heuristics, decomposition methods and relaxation methods.

2.2.3 Different applications

As with the container mentioned above, different applications of the container loading problem can be found in the literature. Two of these problems are the container stowing and stacking problem, and the vehicle routing problem.

Container stowing and stacking problems

While the container loading problems mostly consider the packing of items inside a container, the container stowage planning problem considers the stacking of the containers onto and off cargo ships. Monaco, Sammarra, and Sorrentino (2014) considered a Terminal-Oriented Ship Stowage Planning Problem (SSPP) to minimise the port costs, in terms of the yard and transport operations. Constraints concerning load balancing and weight distribution were considered by stating that containers put on top of each other cannot increase in weight. The loading of the vessels considered the dimensions of the boxes and the dimensions of the cargo hold, as well as the order the boxes needed to be loaded. A problem that the SSPP need to consider, mainly in terms of stacking the containers, is that not all the containers essentially are offloaded at the same port, in most cases not. Therefore, the order that the containers are loaded into the cargo hold is critical in the minimisation of the costs associated with the offloading. To solve the SSPP problem, they created a binary integer model. Given the NP-hardness of the problem, the proposed model is supplemented by a two-step heuristic. The two-step heuristic was based on tabu search and iterative local search algorithm, where the first step found a feasible solution and the second step later tried to obtain a better feasible solution. They concluded that the heuristic approach gave satisfactory results, although they suggested that simulation-based optimisation tools could obtain better estimations.

Vehicle Routing with loading constraints

As mentioned by Bortfeldt and Wäscher (2013), vehicle routing with loading constraints is one of only a few extensions of the bin packing and container loading problem. Bartók and Imreh (2011) considered one such problem. By combining two of the most researched problems in combinatorial optimisation, they state that, even though more complex and challenging models occur, problems more closely related to reality are created. Their paper focused on a pickup and delivery vehicle routing problem with the extension of weight limits and packing- and loading constraints of three-dimensional items. The main goal was to transport all items, given each of the vehicle's demands, while satisfying all of the given constraints. They considered a three-dimensional packing problem deriving from container loading, where several three-dimensional items with a given length, height and width are to be packed into a vehicle of known length, width and height. Weight constraints were also taken into effect in regards to the vehicles weight limit. Before the packing of the vehicles, heuristics regarding route building was implemented. Simple local search heuristics was later applied to improve upon these initial routes. After all the routes were created, heuristics such as block algorithm and interval preparation methods were used to pack the given vehicles. All yielding satisfactory results, with the local search algorithms improving the initial solutions by an average of 15-20%.

2.3 Distinguishing between literature and the problem

Research on load balancing, stability and fragility in container loading is the closest the literature has been to represent real-life scenarios. These problems are amongst the few packing problems that include the weight of the boxes not only in the sense of pallet or bin weight but also in the packing of the given boxes independent from each other. Papers such as Junqueira, Morabito, and Sato Yamashita (2012) and Paquay, Schyns, and Limbourg (2016) have considered packing problems with both stability- and fragility constraints. While these two papers are some of the literature that mostly resembles practically relevant problems, their models only consider small to moderate problems, both stating that further research should be conducted in terms of heuristics and approximation algorithms. Even though these two papers considered both the stability and the fragility of the boxes, the fragility constraints can be seen as too strict. Stating that no fragile boxes can have a box on top off itself is a harsh constraint, bearing in mind that a box put on top off a fragile box does not necessarily equal deformation or damaged goods. A better approach could the constraint used by Bortfeldt (2012), stating that only fragile boxes can be packed upon other fragile boxes. The concept of no box upon a fragile box is implemented to avoid any form of damage and deformation totally but can in the same process disregard a lot of viable packing patterns. Bin packing problems concerning the volume and the weight of the boxes, stability and non-deformation of the boxes as well as the packing of multiple identical bins in a reasonable amount of time does not exists, to the knowledge of the authors. All these practical, relevant constraints are forming the basis of this thesis and is what separates it from the research that has already been conducted in the field of bin packing. The thesis will build upon the model suggested by Paquay, Schyns, and Limbourg (2016), together with our own added constraints. It will solve a problem taken directly from the packing industry and use heuristic approaches to show different and viable packing procedures.

3. Problem Description

The problem considered in this thesis derives from an idea by Driw to check if packing based on weight instead of volume can compete with the packing done by experienced workers. A client of Driw has allowed Driw to provide data on the orders the different branches within the company have placed. These data include the different orders placed by the different customers as well as the dimensions of the boxes, i.e. the length, width, height and weight. The main objective of this thesis is to minimise the number of pallets that are needed to pack a given set of boxes given by their predefined dimensions.

This thesis describes what the inclusion of independent box weight does to the packing of a pallet, concerning both minimising the number of pallets needed to pack all the goods and concerning packing stable and well-balanced pallets. The thesis considers a bin packing problem where a given number of heterogeneous rectangular boxes are to be packed onto a finite number of rectangular pallets. The set of boxes are packed based on their length, width, height and weight.

The boxes are packed based on which order they belong. Different customers place orders which contain a set of items/boxes. Each of the boxes belongs to a specific and distinct order. Boxes belonging to different orders cannot be packed on the same pallet, meaning that boxes can only be packed on pallets along with boxes from the same order. Within each order, the boxes are divided into three different categories or "box-types", them being (A), (B) and (C). These "box-types" are given to the different boxes so that boxes from the same categories are packed on the same pallets. A box belonging to category (A) can only be packed together with other boxes from the said category, with the same logic applying to category (B) and (C) as well. Customers can place more than one order, but even though the pallets are going to the same location, boxes from different orders, regardless of customer, cannot be packed on the same pallet.

All the different boxes are given by their volume and weight. They need to be packed in such a manner that they do not overlap with each other, and all the boxes need to be packed within the dimensions of the pallet. The pallets used are homogeneous, all having the same dimensions. Its base area gives the length and width of the pallet. There are also restrictions on how high the pallet can be when packed. Along with the volume restrictions of the pallet, a weight restriction is also set. Al the packed goods must be placed inside the dimensions of the pallet, while not exceeding the height and weight restriction.

The pallets used are standard Euro-pallets with dimensions 80cm x 120cm x 148 cm. The pallets cannot be loaded with more than 750kg of goods. They are packed orthogonally, meaning that the boxes need to be packed parallel to the pallet "walls" and perpendicular to one another. This sort of packing allows for six different orientations, visualised in Figure 4. In this thesis, only two orientations are allowed, the two being ninety-degree rotations along the X and Y axis, so that top and bottom are always the same, as shown in Figure 5. By allowing the boxes to rotate along the X- and Y-axis only, the notion of a distinct top and bottom of a box is implemented, creating packing solutions that abide by the concept of "this side up".

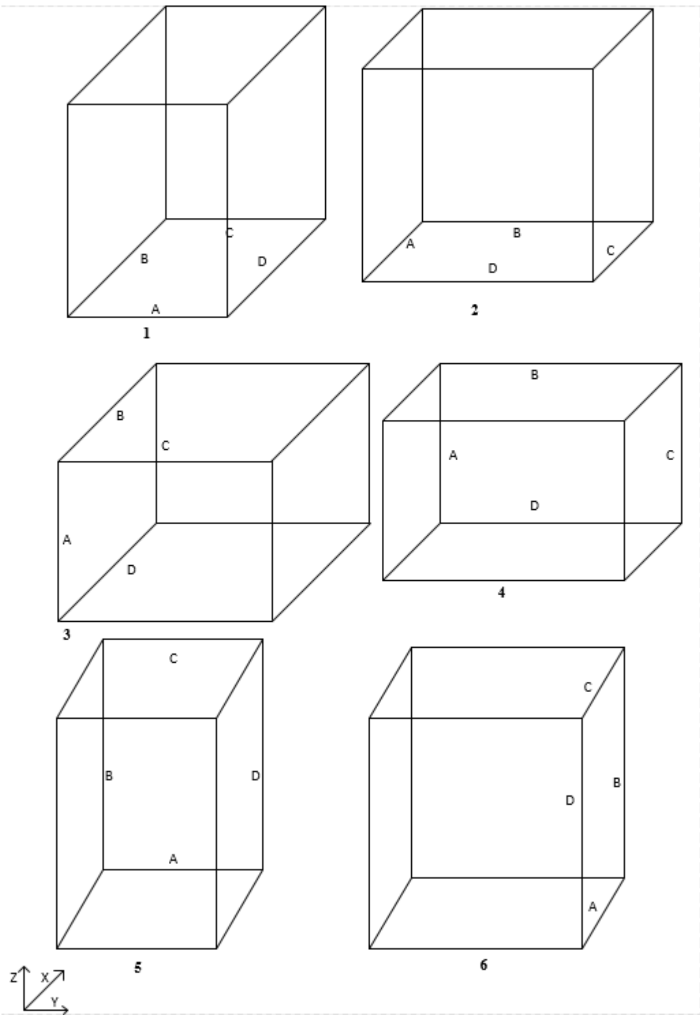


Figure 4. The six rotations allowed in orthogonal packing (made by the authors)

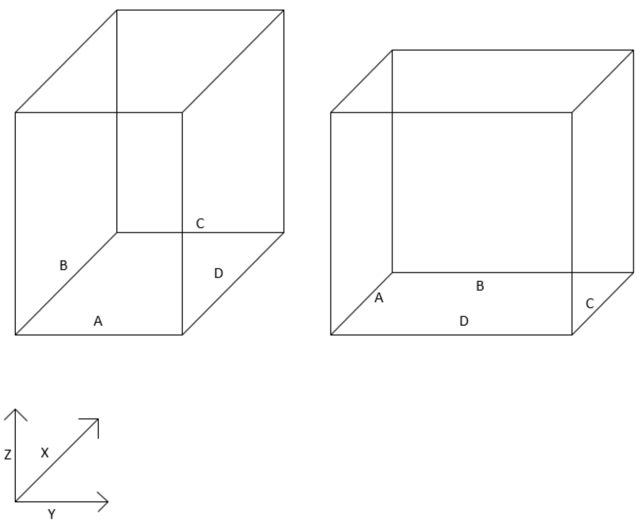


Figure 5. The two orientations allowed in this problem (made by the authors).

Cargo stability is implemented to obtain packing that is safe and stable. The cargo stability in this thesis is concerned with constraints regarding vertical stability, commonly known as static stability. The vertical stability states that the bottom of the boxes needs to be supported either by the floor of the pallet or by the top face of another box. As well as creating good support for the boxes, vertical stability also excludes the possibility of floating boxes. The pallets are also packed in such a manner that the centre of gravity (CG), the middle, of the boxes are supported by the top face of another box or by the pallet floor.

4. Mathematical Model

The mathematical model is made by Paquay, Schyns, and Limbourg (2016), and the goal for the model is to minimise the unused volume packed in a container. It includes real-life applications of bin packing such as stability constraints, the orientation of boxes and weight distribution in the container. We have made some alterations to the original model by adding three new constraints. We have disregarded the possibility for cuts and not implemented the fragility constraints, seeing as we do not know if a box is too fragile to support another box. An explanation of all parameters, sets, indexes and variables can be found in the appendix, including the code used in AMPL.

4.1 Non-modified model

Indexation, sets and parameters

n represents the number of boxes to be packed, and i and j are arbitrary boxes in the set of boxes. The number of pallets available is denoted by m, where k is an arbitrary pallet. l represents an arbitrary vertex where v is the total number of vertices for a box. a is an arbitrary axis in the set O^A. The set O^B represents the different box sides, where b is an arbitrary box side.

Objective function:

$$\min \sum_{k=1,\dots,m} l'w'h'U_k - \sum_{i=1,\dots,n,} l_i w_i h_i \qquad (1)$$

U_k is a binary variable which tells if pallet k is used or not. The parameters l', w' and $h'$are the maximum length, width and height of the pallet and multiplied together they give the maximum volume of a pallet. Parameters l_i, w_i and h_i are the length, width and height for box i. The objective function minimises the total unused volume of the pallets.

S.t.:

$$\sum_{k=1,\dots,m} P_{ik} = 1\,, i = 1, \dots, n \qquad (2)$$

Variable P_{ik} is a binary variable, which tells if box i is on pallet k or not. Constraints (2) tells us that box i is assigned to precisely one pallet.

$$\sum_{i=1,\ldots,n} \gamma_i P_{ik} \le \gamma' U_k, k = 1, \ldots, m \quad (3)$$

Parameters γ_i and γ' represents weight for box i and maximum weight, which can be packed on a pallet, respectively. The weight of all items on pallet k cannot exceed pallet k's weight capacity. It is also defined that U_k must be one if there is a box on pallet k.

Cannot exceed pallet size:

$$X'_i \le \sum_{k=1,\ldots,m} l' P_{ik}, i = 1, \ldots, n \quad (4a)$$

$$Y'_i \le \sum_{k=1,\ldots,m} w' P_{ik}, i = 1, \ldots, n \quad (4b)$$

$$Z'_i \le \sum_{k=1,\ldots,m} h' P_{ik}, i = 1, \ldots, n \quad (4c)$$

The continuous variables X'_i, Y'_i and Z'_i represents the "right back top corner" coordinate for box i for X-, Y- and Z-axis, respectively. Constraints (4a) says that the "right back top" X coordinate for box i cannot exceed the length of pallet k if box i is on pallet k. It is the same for two next constraints except that for constraints (4b) it is the Y-coordinate and width and for constraints (4c) it is the Z-coordinate and height.

Orientation:

$$\sum_{a \in O^A} R_{iab} = 1, i = 1, \ldots, n, b \in O^B \quad (5a)$$

$$\sum_{b \in O^B} R_{iab} = 1, i = 1, \ldots, n, a \in O^A \quad (5b)$$

Introducing a new binary variable R_{iab}, which tells if side b for box i is parallel to axis a or not. Side b for box i can only be parallel to one axis (constraints (5a)), and axis a can only be parallel to one side of box i (constraints (5b)).

$$X'_i - X_i = l_i R_{i,\mathrm{X},\mathrm{L}} + w_i R_{i,\mathrm{X},\mathrm{W}} + h_i R_{i,\mathrm{X},\mathrm{H}}, i = 1, \ldots, n \quad (5c)$$

$$Y'_i - Y_i = l_i R_{i,\mathrm{Y},\mathrm{L}} + w_i R_{i,\mathrm{Y},\mathrm{W}} + h_i R_{i,\mathrm{Y},\mathrm{H}}, i = 1, \ldots, n \quad (5d)$$

$$Z'_i - Z_i = l_i R_{i,\mathrm{Z},\mathrm{L}} + w_i R_{i,\mathrm{Z},\mathrm{W}} + h_i R_{i,\mathrm{Z},\mathrm{H}}, i = 1, \ldots, n \quad (5e)$$

In constraints (5c), (5d) and (5e) we have three new continuous variables X_i, Y_i and Z_i. They are the "left front bottom corner" for X, Y and Z coordinate for box i. The left-hand side of the three constraints is the difference between "right back top corner" and the "left front bottom corner", implying that $X'_i > X_i$, $Y'_i > Y_i$ and $Z'_i > Z_i$. The right-hand side is the length, width and height of box i if side b for box i is parallel to the X-, Y- and Z-axis. As said in constraint (5b), we know that one and only one side b for box i can be parallel to the different axis. Then, only one of the terms on the right-hand side can have a value greater than zero. The three constraints then give the length, width or height of box i if side b is parallel to axis a. This is the difference between the "right back top corner" and the "left front bottom corner". These possible orientations are visualised in Figure 4.

$$R_{i,\mathrm{Z},\mathrm{H}} = 1, i = 1, \dots, n \tag{5f}$$

For all boxes, the height of the box must be parallel to the Z-axis, visualised in Figure 5.

<u>Overlapping:</u>

$$A_{ij} + A_{ji} + B_{ij} + B_{ji} + C_{ij} + C_{ji} \geq P_{ik} + P_{jk} - 1, i, j = 1, \dots, n, k = 1, \dots, m\colon i \neq j \tag{6a}$$

Variables A_{ij}, B_{ij} and C_{ij} are binary variables telling if box i is on the right side, behind or above box j or not. If A_{ij}, B_{ij} and/or C_{ij} is one, it implies that box j is on the left side/in front of/under box i. Overlapping can only happen if box i and box j are on the same pallet. If box i and box j are not right, left, behind, in front, above or under each other (LHS=0), then box i and box j are not on the same pallet ($0 \geq P_{ik} + P_{jk} - 1$).

$$X'_j \leq X_i + (1 - A_{ij})l', i, j = 1, \dots, n\colon i \neq j \tag{6b}$$

$$X_i + 1 \leq X'_j + A_{ij}l', i, j = 1, \dots, n\colon i \neq j \tag{6c}$$

Constraints (6b) and (6c) defines the variable A_{ij}. Note that if the length, width or height of box i are non-integer, change one to a small number in constraints (6c), (6e) and (6g). Figure 6 shows an example where $A_{ij} = 1$.

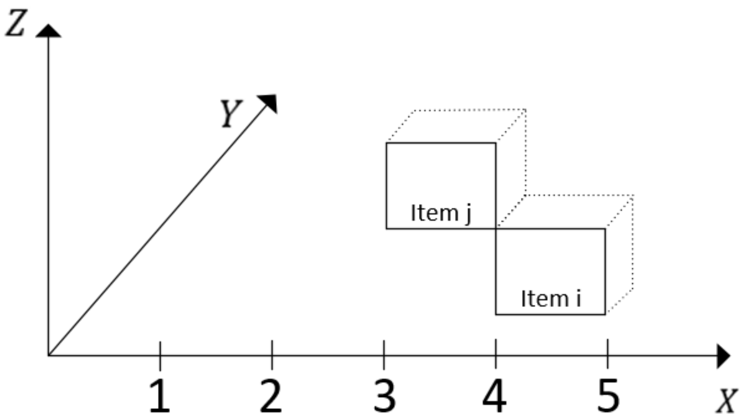


Figure 6. Example of A_ij=1. (made by the authors)

$$Y_j' \leq Y_i + \left(1 - B_{ij}\right)w'i, j = 1, \dots, n\colon i \neq j \tag{6d}$$

$$Y_i + 1 \leq Y_j' + B_{ij}w', i, j = 1, \dots, n\colon i \neq j \tag{6e}$$

Constraints (6d) and (6e) defines the variable B_{ij}. It works the same way as constraints (6b) and (6c).

$$Z_j' \leq Z_i + \left(1 - C_{ij}\right)h', i, j = 1, \dots, n\colon i \neq j \tag{6}$$

$$Z_i \leq Z_j' + C_{ij}h', i, j = 1, \dots, n\colon i \neq j \tag{6g}$$

Constraints (6f) defines the variable C_{ij}. It works the same way as constraints (6b). Constraints (6g) is not in the original model. We added it because C_{ij} did not work correctly. Now it works the same way as constraints (6c) and (6e).

Stability:

$$\sum_{j=1,\dots,n\colon i\neq j} \sum_{l=1,\dots,v} V_{ijl} \geq 3\,(1 - G_i), i = 1, \dots, n \tag{7}$$

Introducing variables G_i and V_{ijl}. G_i is a binary variable saying if box i is on the ground or not. V_{ijl} is a binary variable which tells if vertex l for box i is supported by box j or not. An item has eight vertices, but only the four bottom vertices shown in Figure 7 are used. The constraint says that if box i is not on the ground, box i must have at least three vertices that are supported and removes the possibility for i and j to be equal.

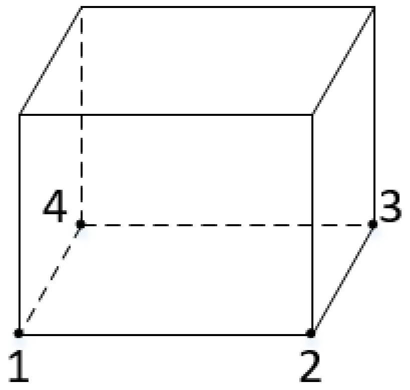


Figure 7. The four bottom vertices used (made by the authors)

$$Z_i \leq (1 - G_i)h', i = 1, \dots, n \quad (8)$$

If box i is on the ground, the Z-coordinate for box i must be zero.

Defining suitable height:

$$Z_j' - Z_i \leq T_{ij}, i, j = 1, \dots, n: i \neq j \quad (9a)$$

$$Z_i - Z_j' \leq T_{ij}, i, j = 1, \dots, n: i \neq j \quad (9b)$$

$$T_{ij} \leq Z_j' - Z_i + 2\left(1 - \delta_{ij}\right)h', i, j = 1, \dots, n: i \neq j \quad (9c)$$

$$T_{ij} \leq Z_i - Z_j' + 2\delta_{ij}h', i, j = 1, \dots, n: i \neq j \quad (9d)$$

The variables T_{ij} and δ_{ij} define Q_{ij} in constraints (10a-b). T_{ij} represents the absolute value of $\left|Z_j' - Z_i\right|$ and δ_{ij} is a binary variable which takes value one if $Z_j' \geq Z_i$, zero otherwise. Constraints (9a), (9b), (9c) and (9d) define the absolute value.

$$Q_{ij} \leq T_{ij}, i, j = 1, \dots, n: i \neq j \quad (10a)$$

$$T_{ij} \leq Q_{ij}h', i, j = 1, \dots, n: i \neq j \quad (10b)$$

Variable Q_{ij} is a binary variable taking value zero if box j has a suitable height to support box i ($Z_j' = Z_i$), one otherwise. Constraints (10a-b) says that if $T_{ij} > 0$, then box j does not have a suitable height to support box i ($Q_{ij} = 1$). If $T_{ij} = 0$, then box j has a suitable height to support box i ($Q_{ij} = 0$). Note that if the height of a box is non-integer, a small number must be multiplied with Q_{ij} in constraints (10a). This is because T_{ij} can have a value greater than zero and less than one. For example, if $T_{ij} = 0{,}5$, then $0 < \epsilon Q_{ij} \leq 0{,}5$.

Shared projection:

$$I_{ij} \leq A_{ij} + A_{ji} + B_{ij} + B_{ji}, i, j = 1, \ldots, n: i \neq j \tag{11a}$$

$$A_{ij} + A_{ji} + B_{ij} + B_{ji} \leq 2I_{ij}, i, j = 1, \ldots, n: i \neq j \tag{11b}$$

Variable I_{ij} is a binary variable taking value zero if the projection on the XY plane of box i and box j have a non-empty intersection. If there is a non-empty projection on the X plane, it means that box i does not stand on the right or left side of box j. It is the same for the Y plane, but instead of right/left, box i does not stand behind or in front of box j. Constraints (11a) and (11b) says that if box i and box j have a non-zero projection on the XY plane ($I_{ij} = 0$), box i and j cannot stand on the right/left side and behind/in front of each other.

Support:

$$\left(1 - S_{ij}\right) \leq Q_{ij} + I_{ij}, i, j = 1, \ldots, n: i \neq j \tag{12a}$$

$$Q_{ij} + I_{ij} \leq 2\left(1 - S_{ij}\right), i, j = 1, \ldots, n: i \neq j \tag{12a}$$

Introducing variable S_{ij}, which is a binary variable saying if box i is supported by box j or not. Constraints (12a) and (12b) says that if box i is supported box j, box j has the suitable height to support box i and it is a projection on the XY plane between box i and box j.

Support if they are on the same pallet:

$$P_{ik} - P_{ki} \leq \left(1 - S_{ij}\right), i, j = 1, \ldots, n,\ k = 1, \ldots, m: i \neq j \tag{13a}$$

$$P_{ki} - P_{ik} \leq \left(1 - S_{ij}\right), i, j = 1, \ldots, n,\ k = 1, \ldots, m: i \neq j \tag{13b}$$

These two constraints say that if box i is supported by box j, box i and j is on the same pallet.

Certified support:

$$V_{ijl} \leq S_{ij}, i, j = 1, \ldots, n,\ l = 1, \ldots, v: i \neq j \tag{14}$$

Constraints (14) makes sure that if vertex l for box i is supported by box j ($V_{ijl} = 1$), box i is supported by box j ($S_{ij} = 1$).

Box vertex support:

$$X_j \leq X_i + \beta_{ij}^X l', i, j = 1, \dots, n\colon i \neq j \quad (15a)$$

$$Y_j \leq Y_i + \beta_{ij}^Y w', i, j = 1, \dots, n\colon i \neq j \quad (15b)$$

$$X_i' \leq X_j' + \beta_{ij}^{X'} l', i, j = 1, \dots, n\colon i \neq j \quad (15c)$$

$$Y_i' \leq Y_j' + \beta_{ij}^{Y'} w', i, j = 1, \dots, n\colon i \neq j \quad (15d)$$

Constraints (15a-d) defines the new binary variables β_{ij}^X, β_{ij}^Y, $\beta_{ij}^{X'}$ and $\beta_{ij}^{Y'}$. The variables represent if it is possible for vertex l for box i to be supported by box j. The variable β_{ij}^X takes the value one if the "left front bottom" X coordinate for box j is larger than the "front left bottom" X coordinate for box i. It is the same for β_{ij}^Y except that it is for the Y coordinate instead of the X coordinate. The variable $\beta_{ij}^{X'}$ takes value one if the "back right top" X coordinate for box i is larger than the "left front bottom" X coordinate for box j. It is the same for $\beta_{ij}^{Y'}$, however, instead of the X coordinate, it is the Y coordinate. Note that all the new variables can take value zero or one if the left-hand side is less than or equal to the right-hand side, but if $V_{ijl} = 1$, then the corresponding binary variables in constraints (16a-d) must take value zero.

$$\beta_{ij}^X + \beta_{ij}^Y \leq 2(1 - V_{ij1}), i, j = 1, \dots, n, \ l = 1, \dots, v\colon i \neq j \quad (16a)$$

$$\beta_{ij}^Y + \beta_{ij}^{X'} \leq 2(1 - V_{ij2}), i, j = 1, \dots, n, \ l = 1, \dots, v\colon i \neq j \quad (16b)$$

$$\beta_{ij}^{X'} + \beta_{ij}^{Y'} \leq 2(1 - V_{ij3}), i, j = 1, \dots, n, \ l = 1, \dots, v\colon i \neq j \quad (16c)$$

$$\beta_{ij}^X + \beta_{ij}^{Y'} \leq 2(1 - V_{ij4}), i, j = 1, \dots, n, \ l = 1, \dots, v\colon i \neq j \quad (16d)$$

Constraints (16a) says that if vertex number one for box i is supported by box j, then X_j is less than or equal to X_i and Y_j is less than or equal to Y_i. The left-hand side represents the "left front bottom vertex". This is the same for constraints (16b), (16c) and (16d) except that the left-hand side represents vertex 2 ("right front bottom vertex"), vertex 3 ("right back bottom vertex") and vertex 4 ("left back bottom vertex"), respectively. In Figure 8, vertex 1 for box i is supported by box j. That means that $X_j \leq X_i$ and $Y_j \leq Y_i$.

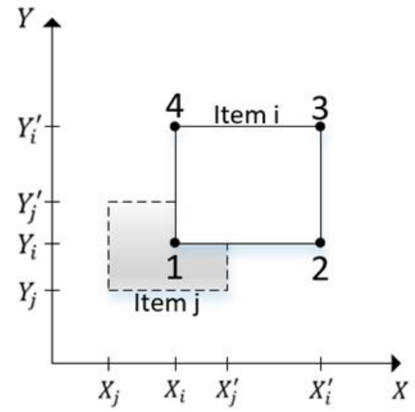


Figure 8. Figure showing vertex support (Paquay, Schyns, and Limbourg (2016), pp.198)

Weight distribution:

Coordinate for the mass of box i.

$$\gamma_i^X = \frac{X_i + X_i'}{2}, i = 1, \dots, n \tag{17a}$$

$$\gamma_i^Y = \frac{Y_i + Y_i'}{2}, i = 1, \dots, n \tag{17a}$$

$$\gamma_i^Z = \frac{Z_i + Z_i'}{2}, i = 1, \dots, n \tag{17a}$$

γ_i^X, γ_i^Y and γ_i^Z are continuous variables for the mass of box i. They represent the X, Y and Z coordinate for the middle point of box i, assuming that the weight is uniformly distributed.

$$\sigma_{ik}^X \leq l' P_{ik}, i = 1, \dots, n, k = 1, \dots, m \tag{18a}$$

$$\sigma_{ik}^X \leq \gamma_i^X, i = 1, \dots, n, k = 1, \dots, m \tag{18b}$$

$$\sigma_{ik}^X \geq \gamma_i^X - l'(1 - P_{ik}), i = 1, \dots, n, k = 1, \dots, m \tag{18c}$$

Variable σ_{ik}^X is a new variable representing $\sigma_{ik}^X \equiv P_{ik}\gamma_i^X$. It gives the X coordinate for the centre of gravity for box i if box i is on pallet k. This is not linear since there are two variables multiplied together. To make it linear, constraints (18a), (18b) and (18c) are needed.

Constraints (18a): Defines if box i is on pallet k.

Constraints (18b): σ_{ik}^{X} must be less than or equal to γ_{i}^{X}, which is the X coordinate for box i's mass, i.e. the centre of gravity.

Constraints (18c): σ_{ik}^{X}must be higher than or equal to γ_{i}^{X} if box i is on pallet k. If box i is not on pallet k, the centre of gravity does not exist for this box on this pallet.

The interval for the centre of gravity;

$$(\frac{l'}{2}\alpha^{\mathrm{l}'})(\sum_{i=1,\ldots,n}\gamma_i P_{ik}) \leq) \sum_{i=1,\ldots,n}\sigma_{ik}^{X}\gamma_i, k = 1, \ldots, m \quad (19a)$$

$$\sum_{i=1,\ldots,n}\sigma_{ik}^{X}\gamma_i \leq \left(\frac{l'}{2}+\alpha^{\mathrm{l}'}\right)\left(\sum_{i=1,\ldots,n}\gamma_i P_{ik}\right), k = 1, \ldots, m \quad (19b)$$

Introducing a new parameter $\alpha^{\mathrm{l}'}$ to give the range in which the centre of gravity can be located. Constraints (19a) is the lower bound, and constraints (19b) is the upper bound for the centre of gravity X coordinate for all boxes on pallet k.

$$\sigma_{ik}^{Y} \leq w' P_{ik}, i = 1, \ldots, n, k = 1, \ldots, m \quad (20a)$$

$$\sigma_{ik}^{Y} \leq \gamma_{i}^{Y}, i = 1, \ldots, n, k = 1, \ldots, m \quad (20b)$$

$$\sigma_{ik}^{Y} \geq \gamma_{i}^{Y} - w'(1 - P_{ik}), i = 1, \ldots, n, k = 1, \ldots, m \quad (20c)$$

σ_{ik}^{Y} is a new variable representing $\sigma_{ik}^{Y} \equiv P_{ik}\gamma_{i}^{Y}$. Constraints (20a-c) works the same way as constraints (18a-c), but for the Y coordinate.

$$(\frac{w'}{2}-\alpha^{\mathrm{w}'})(\sum_{i=1,\ldots,n}\gamma_i P_{ik}) \leq) \sum_{i=1,\ldots,n}\sigma_{ik}^{Y}\gamma_i, k = 1, \ldots, m \quad (21a)$$

$$\sum_{i=1,\ldots,n}\sigma_{ik}^{Y}\gamma_i \leq \left(\frac{w'}{2}+\alpha^{\mathrm{w}'}\right)\left(\sum_{i=1,\ldots,n}\gamma_i P_{ik}\right), k = 1, \ldots, m \quad (21b)$$

Introducing a new parameter $\alpha^{\mathrm{w}'}$ to give the range in which the centre of gravity can be located. Constraints (21a) is the lower bound, and Constraints (21b) is the upper bound for the centre of gravity Y coordinate for all boxes on pallet k.

$$\sigma_{ik}^{Z} \leq h' P_{ik}, i = 1, \dots, n, k = 1, \dots, m \tag{22a}$$

$$\sigma_{ik}^{Z} \leq \gamma_i^{Z}, i = 1, \dots, n, k = 1, \dots, m \tag{22b}$$

$$\sigma_{ik}^{Z} \geq \gamma_i^{Z} - h'(1 - P_{ik}), i = 1, \dots, n, k = 1, \dots, m \tag{22c}$$

σ_{ik}^{Z} is a new variable representing $\sigma_{ik}^{Z} \equiv P_{ik}\gamma_i^{Z}$. Constraints (20a-c) works the same way as Constraints (18a-c), but for the Y coordinate.

$$0 \leq \sum_{i=1,\dots,n} \sigma_{ik}^{Z}\gamma_i, k = 1, \dots, m \tag{23a}$$

$$\sum_{i=1,\dots,n} \sigma_{ik}^{Z}\gamma_i \leq \alpha^{\mathrm{h}'}\left(\sum_{i=1,\dots,n} \gamma_i P_{ik}\right), k = 1, \dots, m \tag{23b}$$

Introducing a new parameter $\alpha^{h'}$ to give the upper bound in which the centre of gravity can be located. Constraints (23a) is the lower bound, and constraints (23b) is the upper bound for the centre of gravity Z coordinate for all boxes on pallet k.

Logical support:

$$S_{ij} + S_{ji} \leq 1, i, j = 1, \dots, n: i \neq j \tag{24}$$

This constraint was added to the model. This is because the model said that it was possible for box i to be supported by box j and box j to be supported by box i. This is not possible, so we added a constraint stating that if box i is supported by box j, then box j cannot be supported by box i, and vice versa.

$$U_k \leq U_{k-1}, k = 2, \dots, m \tag{25}$$

Constraints (25) says that if a pallet is used, the previous pallet must be used as well. This is the case for all the pallets from the second one up until m. This constraint was added to the original model.

$$P_{ik} \in \{0,1\}, i = 1, \dots, n, k = 1, \dots, m \tag{26a}$$

$$U_k \in \{0,1\}, k = 1, \dots, m \tag{26b}$$

$$G_i \in \{0,1\}, i = 1, \dots, n \tag{26c}$$

$$R_{iab} \in \{0,1\}, i = 1, \dots, n, a \in O^A, b \in O^B \tag{26d}$$

$$A_{ij}, B_{ij}, C_{ij}, Q_{ij}, I_{ij}, S_{ij}, \beta_{ij}^{X}, \beta_{ij}^{Y}, \beta_{ij}^{X'}, \beta_{ij}^{Y'}, \delta_{ij} \in \{0,1\}, i, j = 1, \dots, n \tag{26e}$$

$$V_{ijl} \in \{0,1\}, i, j = 1, \dots, n, l = 1, \dots, v \tag{26f}$$

Constraints (26a-f) says that these variables are binary.

$$X_i, Y_i, Z_i, X_i', Y_i', Z_i', \gamma_i^X, \gamma_i^Y, \gamma_i^Z \geq 0, i = 1, \dots, n \tag{27a}$$

$$T_{ij} \geq 0, i, j = 1, \dots, n \tag{27b}$$

$$\sigma_{ik}^X, \sigma_{ik}^Y, \sigma_{ik}^Z \geq 0, i = 1, \dots, n, k = 1, \dots, m \tag{27c}$$

Constraints (27a-c) ensures that these variables do not take a negative value.

Complexity:

Number of variables $O(n^2)$

Number of constraints $O(n^2 m)$

4.2 Modified model

The modified model used when implementing the heuristic has the same objective function but considers only constraints (2 – 6g).

4.3 Locked position model:

The model used when implementing the locked position approach has the same objective function and constraints. The difference is that it also adds constraints (28a-e), locking the coordinates of each of the packed boxes, as well as the pallet and the orientation of the boxes.

$$X_i = X\ coordinate\ for\ a\ packed\ box \tag{28a}$$

$$\mathrm{Y_i} = \mathrm{Y\ coordinate\ for\ a\ packed\ box} \tag{28b}$$

$$Z_i = Z\ \text{coordinate for a packed box} \tag{28c}$$

$$P_{ik} = 1 \tag{28d}$$

$$R_{i,X,L} = 1, if\ lenght\ of\ a\ packed\ box\ is\ paralell\ to\ the\ X - axis, 0\ otherwise \tag{28e}$$

Constraints (28a-c) locks the X, Y and Z coordinate for box i. Constraint (28d) says that box i must be on pallet k. It locks the coordinates for box i on pallet k. Constraint (28e) gives the orientation for box i. It locks this orientation according to if the length is parallel to the X- or Y-axis, with the width for box i being parallel to the opposite axis.

5. Heuristic and construction procedure

This part is dedicated to the development and creation of a heuristic that can help the model to solve larger problems. In the creation of the heuristic, inspiration has been taken from both Bischoff, Janetz, and Ratcliff (1995) and Bischoff (2006), two heuristic approaches that were created to achieve effective and stable packing.

As can be seen in the literature and the NP-hard definition of the problem, solvers for bin packing problems can only solve very small problems. Implementing stability, load balancing and fragility complicates the problem either further. A recurring conclusion in the literature is that heuristics and approximation algorithms need to be implemented to solve larger problems with reasonable running time.

These implemented heuristics are often construction heuristics and includes the two well-known approaches layer- and wall building. Construction heuristics such as the one suggested by Chien and Deng (2004) rank the boxes based on five different criteria, rank 1 being the boxes with the largest base dimensions. Chien and Deng (2004) state that the ranking system is created to obtain stable packing. The boxes with the largest base areas are packed as near as possible to the bottom of the pallet so that that proper support can be achieved. What is not considered in the ranking system, however, is the weight of the boxes. Boxes with small base areas can be heavier, even a lot heavier, than boxes with large base areas. Therefore, ranking only based on width, height and length can cause deformation and damages to the packed goods. To include weight in the packing process, the heuristic created and suggested in this thesis is based on the layer building construction heuristic, meaning that the boxes are packed layer upon layer. Layer building is chosen because it makes it easier to specify that the heaviest boxes are to be packed first and nearest the bottom of the pallet. The heuristic is visualized in the detailed flowchart in Figure 9.

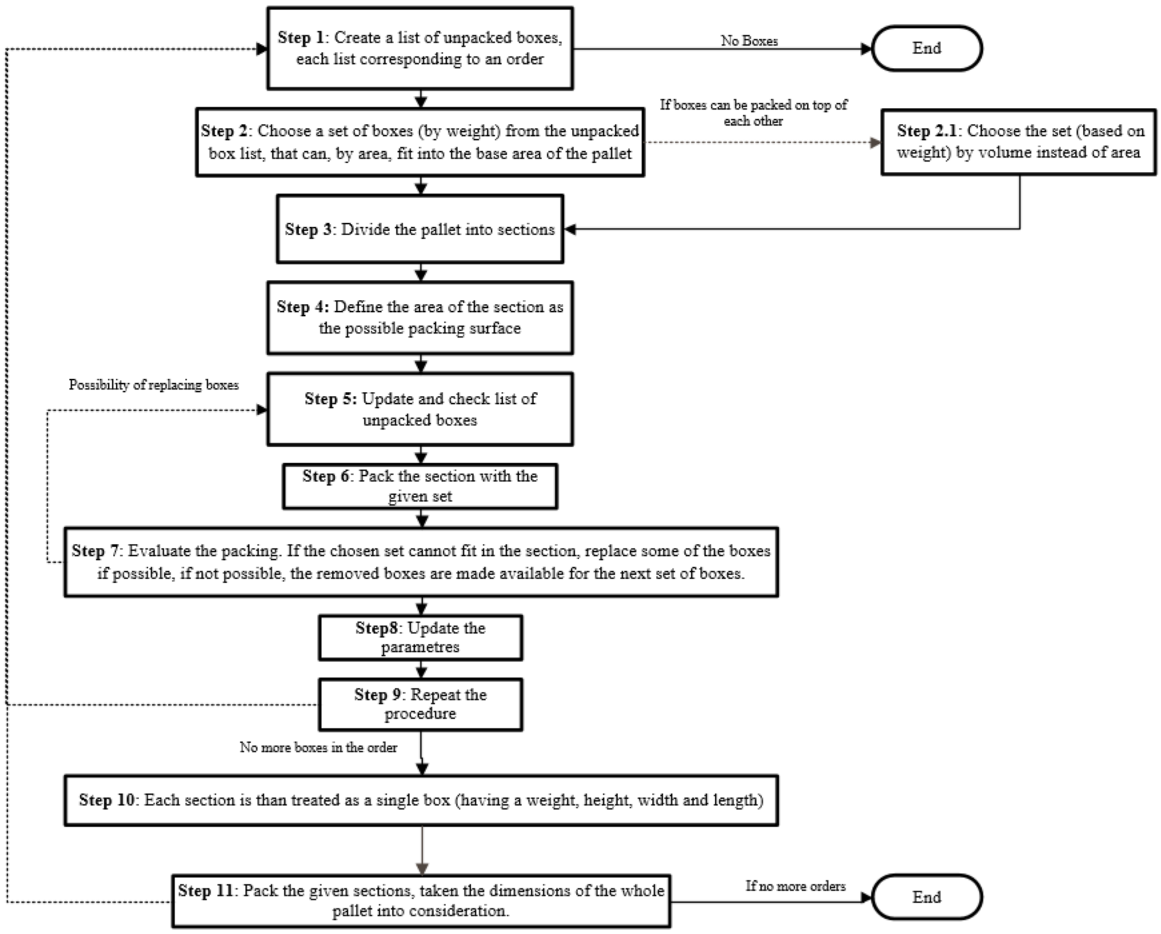


Figure 9. Flowchart of the construction heuristic (made by the authors)

Step 1. Create a list of unpacked boxes. Only boxes from the same order can be put onto the same pallet, so each order corresponds to a packing list. Create the list in such a manner that the boxes are sorted from heaviest to lightest. The list also includes the volume and area of the boxes, given by its length, width and height. End if no more boxes are left.

Step 2. Choose a set of boxes, starting with the heaviest boxes on the top of the list. Always start at the top of the list and follow it top to bottom, to ensure that the heaviest boxes are chosen first. Since the base area of the pallet is always the same and non-changing, the set of boxes are chosen based on their area. Sum up the area of the boxes, starting from heaviest, until it hits the target value that is equal to the base area, 80*120 = 9600, of the pallet. Height of the tallest box in the set is the initial height of the section.

Step 2.1. If a set chosen in step 2 have boxes that can be put on top of each other, redo the set choosing step, and base the selection on how many boxes that can fit into the given volume instead of area. The height of the largest box is given as the initial section height. The choosing is still based on weight.

Step 3. The pallet is divided into different sections. The base area of the pallet remains the same, but the height is considerably lowered. This creates layers within the pallet that allows for the problem to be solved in subsequent procedures. The height of the pallet is chosen based on the tallest box in the set of boxes that are chosen to be packed in step 2, which means that each of the sections created can have different volume and size.
Check and update the list of unpacked boxes, removing the boxes that are chosen in step 4.

Step 4. The area of the section that is created in the first step is then defined as the possible packing surface, which means that the new packing area consists of the pallets base area and the new height agreed upon in step 3.

Step 5. Check and update the list of unpacked boxes, removing the boxes that are chosen in step 2 or 2.1. The area of the section that is created in the first step is then defined as the possible packing surface, which means that the new packing area consists of the pallets base area and the new height agreed upon in step 1.

Step 6. Use the modified model to pack the section of the pallet.

Step 7. Evaluate the packing conducted in step 6. If some boxes do not fit inside the packing area, go back to step 5 and check if there are any boxes on the list that can be put onto the pallet, still choosing the heaviest possible boxes. Update the packing list by adding back the boxes that did not fit and remove the boxes that were used. When the section is packed to satisfaction, move on to the next step.

Step 8. Update all the parameters. The layer is now packed.

Step 9. Repeat the procedure until no more boxes are left in the order. When all the sets are packed, move on to the next step.

Step 10. Each of the packed layers within each order is now considered as a box with its length, width, height and weight. These boxes will be completely rectangular, having smooth edges. In practice the layers will have an uneven footprint, meaning that some boxes will float, and some gaps between the boxes are to be expected, as well as gaps between the boxes and the pallet “walls”.

Step 11. Pack the pallet with the sections created in the earlier steps, taking the parameters of the whole pallet into consideration. When there are no more orders left, and all the boxes are packed, the packing is finished.

6. Computational study

The models presented in this thesis have been tested on a real-life packing problem. The main goal is to minimise the number of pallets needed to pack all the goods. Windows 10 on a desktop computer with an Intel Core i5-6500T CPU with 2.5GHz and 16 GB of Ram was used in the experiments. The models are coded using AMPL language and solved with CPLEX 12.8.0.0.

The tool used to visualise the packing conducted is a CLP spreadsheet solver created and developed by Dr Güneş Erdoğan. By implementing the coordinates and the orientations gathered by the model, the spreadsheet solver creates a three-dimensional image of how the boxes are packed on the pallet.

6.1 Context

In the dataset provided by Driw, costumer-, order- and item-number are included, along with the section the item is placed in the warehouse and the name of the different items. The computational study only considers the length, width, height and the weight of the items, and to which of the three types the item belongs. This is the case given that the objective is to minimise the number of pallets, making the details mentioned above obsolete and unnecessary. Also included in the dataset is the id of the bin the workers have placed the boxes. This data is used to compare the packing done by the workers and the packing obtained by the approaches conducted in the study.

In the experiments conducted, eight orders are used to compare the different packing possibilities. These eight orders contain two orders that the workers have used one pallet each to pack, two orders that the workers have used two pallets each to pack, three orders that the workers have used three pallets each to pack and two orders that the workers have used four pallets each to pack. One of the orders packed on two pallets is an order where three pallets were used by the workers, but since one of the pallets was dedicated to a single item type and included almost no boxes, the order is being seen as an order packed on two pallets. The same logic can also be found on other orders.

When implementing the different procedures to pack the pallets, the model was given a time-limit of 300 seconds to obtain an answer, either successfully or by declaring the problem infeasible. The time-limit was set so that solutions could be found quickly and that the procedures would not take too much time.

To showcase the features of the non-modified model, a problem of our creation will be used. This problem will consist of packing eight rectangular boxes into one bin. The bin used in the example will have a smaller dimension than what the real pallets have. The problem will work as a viability check of the model and show how it performs under facilitated conditions.

The modified model does not include weight distribution, stability or support. The heuristic approach implemented is a layer-building heuristic, meaning that the pallet is filled up layer by layer. The constraints are not included so that the model used with the heuristic is not too complicated and time-consuming. The idea is that the heuristic may create stable and well-packed pallets on its own.

6.2 Non-modified model

This section is dedicated to showing how the full model presented by Paquay, Schyns, and Limbourg (2016) works. Firstly, a viability check of the model will be presented. It will then be implemented a heuristic, locking the positions of the boxes as they are packed onto the entirety of the pallet.

6.2.1 Viability check

The non-modified model should create well-balanced and stable packed pallets, that makes sure that the centre of gravity of each box is within the range of what is allowed. To see if the model works as promised a small packing problem is conducted and visualised in Figure 10

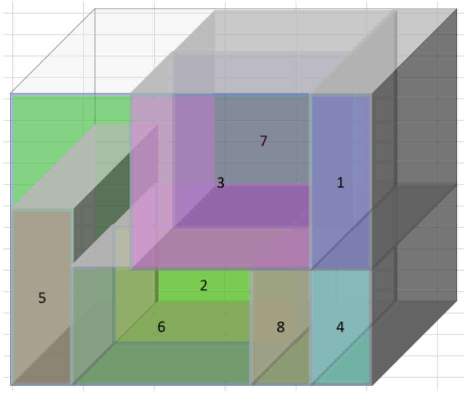


Figure 10. Packing conducted by the non-modified model (made by the authors)

Figure 10 shows that all the boxes are either supported by the floor of the pallet or by another box. It is also clear that at least three of the bottom vertices of the boxes are supported by another box or boxes. This also means that all the boxes are well supported and that the pallet is well-balanced, having uniform weight distribution. The problem visualised in Figure 10 shows that the non-modified model behaves and acts the way it should, satisfying all the constraints created for it.

6.2.2 Locked-position approach

The order packed contained 541 boxes. To begin with, the positions of 17 boxes needed to be locked. In total, the model needed to be solved 144 times, with an average of 3,8 boxes per solve. The model ended ups packing a total of 13 pallets, which is nine more than the workers used to pack the same order. Figure 11 and figure 12 visualise two of the pallets packed by the model.

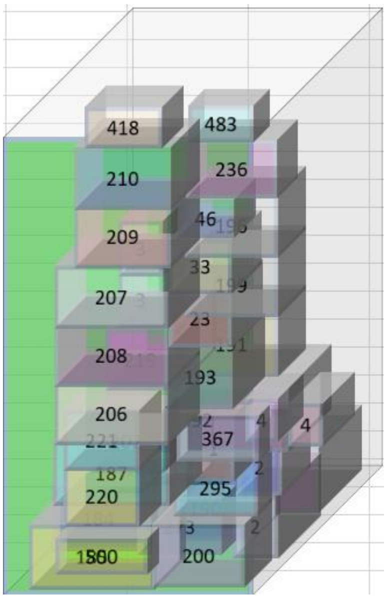


Figure 12. One of the pallets packed with the locked-position approach (made by the authors)

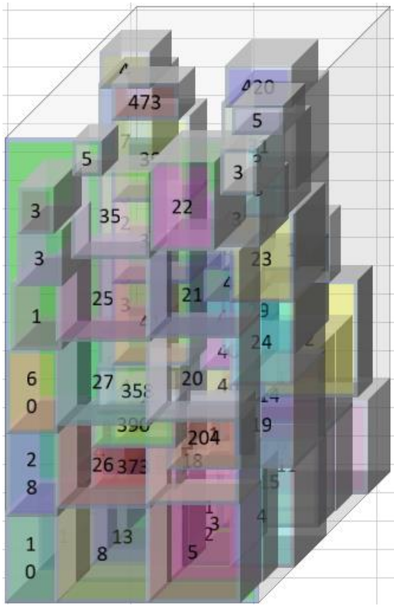


Figure 11. One of the pallets packed with the locked-position approach (made by the authors)

The visualisation of the packed pallets makes it apparent that the volume utilisation varies between the different pallets that are packed using this method. It is also clear that the biggest issue with the implementation is the constraint concerning the support of the vertices. For example, in Figure 11, it is enough space for a box to be placed above box 200 in the right bottom corner. However, because three vertices need to be supported, the box must be quite small to fit on top of box 200. The problem with the vertex support is also shown by the continuous stacking along with the height of the pallet, not utilising much of the volume around the centre. If the boxes on the top have an area smaller than the area of the next box, it cannot support the next box unless there are other boxes as well with the same height. This may create a packing solution similar to how a pyramid is shaped**.**

Locking the coordinates of boxes that have been packed makes for a static packed pallet, which means that boxes that are to be packed do not have an impact on the already packed boxes. The model packs the set of boxes with concern to the boxes already packed but without concern to the boxes that are to be packed later. This creates good packing for the given set of boxes, but not good packing for the pallet as a whole, seeing as it optimises the packing for that particular set of boxes.

Comparing the solutions achieved by the approach implemented and the solution achieved by the workers shows that there are significant discrepancies between the two, as visualised in Figure 13.

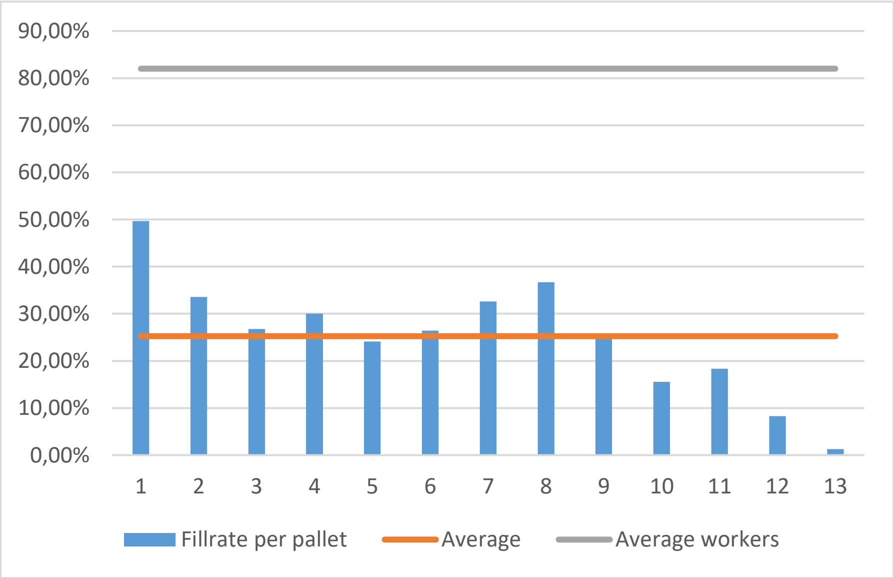


Figure 13. Fill rate of packed pallets, average fill rate and average fill rate workers (made by the authors)

The fill rates visualised in Figure 13 shows even more evident the ineffectiveness of the procedure conducted. The workers managed to achieve an average fill rate equivalent to 82%, utilising over 90% of three out of four pallets. The locked-position approach only managed to achieve an average fill rate of 25%, with the best pallet utilising, approximately 50% of the pallet.

Another issue faced with this approach concerns the aspect of time. To lock each of the positions, the constraints needed to be manually inserted into the model. This is a very tiresome procedure that takes a long time to do. Along with the allowed 300 seconds of running time for each of the implementations, it took days to conduct the packing of just this one order.

6.3 Modified model

Comparisons between the pallets packed by the workers and the pallets packed with the use of the heuristic.

6.3.1 Heuristic implementation

Table 1 provides an overview of the pallets packed by the workers and the heuristic in the eight orders that are investigated and showcases the differences between them.

Table 1. Comparison between heuristic and workers (made by the authors)

Order Number	**Number of Pallets**		**Unused Volume ($1000cm^3$)**		**Differences**	
	Model	**Workers**	**Model**	**Workers**	**Pallets**	**Utilization**
ORDER 1	1	1	1.163,517	1.163,517	0	0
ORDER 2	1	1	643,540	643,540	0	0
ORDER 3	2	2	350,387	57,643	0	0
			576,512	869,257		
ORDER 4	3	2	353,869	56,344	1	1.396,168
			949,826	520,852		
			669,634			
ORDER 5	4	3	384,752	95,710	1	1.430,185
			899,068	156,013		
			298,871	259,845		
			359,061			
			423,591	93,780		
			473,435	71,358		

ORDER 6	4	3	851,787	671,968	1	1.376.524
			464,817			
ORDER 7	5	4	500,538	122,510	1	1.420,800
			477,539	42,431		
			506,528	41,117		
			486,915	813,945		
			469,284			
ORDER 8	6	4	453,626	74,971	2	2.665,567
			372,880	66,483		
			413,051	80,476		
			686,585	97,624		
			335,683			
			741,296			

The workers managed to pack all the boxes in these eight orders on 20 pallets, while the heuristic needed 26 pallets to pack the same amount. It is accounting for a 30% increase in the pallets that are needed when the heuristic approach is implemented. These findings support a notion that a layer building heuristic solely based on weight will have issues with packing problems that have a set of strongly heterogeneous boxes. The packing conducted in this thesis has shown that basing the box selection procedure solely on weight has created packing solutions that have too much-unused volume in the different sections, subsequently leading to pallets that have much more available volume than what is seen is the case with the pallets packed by the workers.

Order 3 gave the same number of pallets with the heuristic as the workers did. Even though this is the case, the two pallets packed in this order are packed very differently. Where the pallets packed by the heuristic have similar unused volume, the once-packed by the workers have not. One of the pallets is packed full, while the other is not. Looking at the other orders, the same pattern emerges. It is apparent that the workers are packing as much as possible on the first pallet and then moving on to another pallet when the first one is as full as it can be.

This is something that the heuristic approach is not capable of doing to the same degree, thus creating two half-full pallets.

Along with order 3, order 1 and 2 were also packed on the same number of pallets with the heuristic as by the workers. This is the case mostly because the orders that end up being packed on one pallet include few items, as shown by the sizeable unused volume these two orders have. So even though the heuristic manages to pack the items on the same number of pallets as the workers, these orders are not representable of how the heuristic performs.

Figure 14 visualises the differences between how full the different pallets in the orders are packed, in percentage, by the heuristic and the workers. The fill rates from the two first orders are excluded since these orders do not represent the all-around performance of the heuristic.

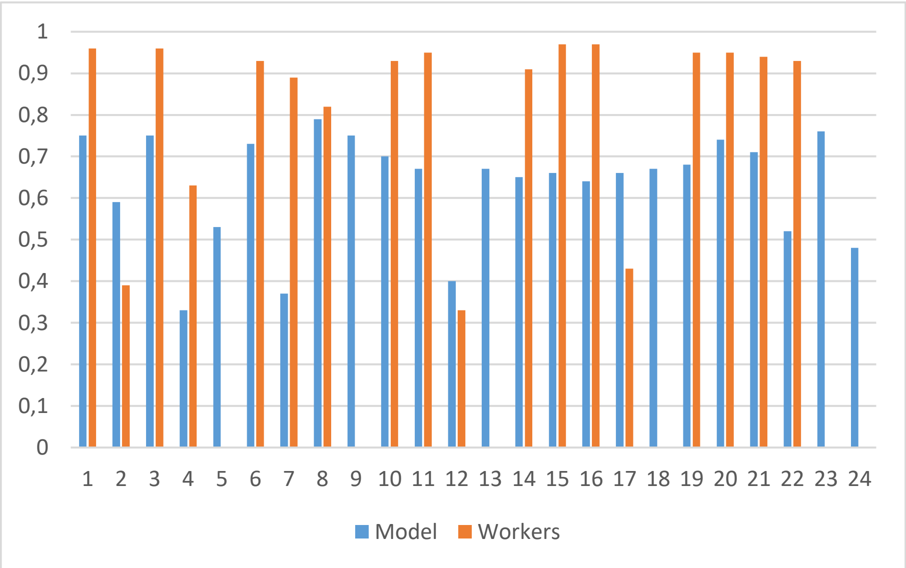


Figure 14. Fill rate comparison (made by the authors)

The fill rates of the pallets packed by the workers are far better than the once achieved by the heuristic approach. The average fill rate achieved by the workers on their 18 pallets is approximately 83%, versus 63% average for the heuristics 26 pallets. The best fill rate achieved with the heuristic is 79% on pallet 8, versus 97% achieved by the workers on both pallet 15 and 16. It is also worth mentioning that the workers manage to achieve a fill rate of over 90% on 12 out of the 18 pallets they used for the packing, not including the 89% achieved on pallet 7.

The reason behind the substantial differences in volume utilisation comes down to the box selection process. Weight selection could be a good way to get the heaviest boxes as low as possible on the pallet, but this comes at the expense of volume utilisation. Not considering the individual volume of the boxes creates a lot of difficult packing situations. A set of boxes with approximately the same weight can be chosen to be packed into a section of the pallet while the volume of the boxes is not taken into account. Situations, where very tall boxes are put in a set with boxes that do not have approximately the same height or can be stacked on top of each other to achieve a satisfying height, frequently occur when implementing the heuristic. Since the heuristic puts an "invisible" sheet between the boxes packed in different sections, visualised in Figure 15 and 16, sections like these will have much-unused volumes and significantly contribute to a large amount of available volume on the finale pallets.

Figure 16. Pallet showing the "invisible" sheet on a poorly packed pallet (made by the authors)

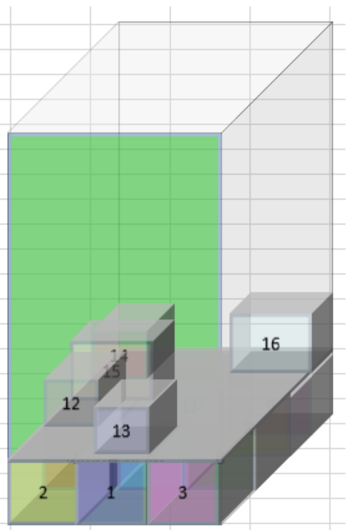


Figure 15. Pallet showing the "invisible" sheet on a well-packed pallet (made by the authors)

The bottom section of the pallet visualised in Figure 16 is taken from order 7. This section contains 55 boxes and is the worst utilised section in the entire order. Much of the reason behind this is to do with one of the boxes being a lot taller than the rest of the boxes. This box sets the height limit of the section, meaning that it should be paired with boxes of approximately the same height or boxes that can be stacked on top of each other to achieve a similar height. In this section, with the weight selection criteria for the boxes, this is not the case. Large portions of the section are unused, and packing utilisation is poor. Problems similar to this happens too often and prohibits the packing from competing with the one conducted by the workers.

In some cases, packing as the one visualised in Figure 15 occurs. This packing is one of the better that the heuristic creates. Here the boxes in the two sections have approximately the same dimensions, as well as weight. In instances when boxes are relatively equal, the heuristic approach works well, leaving little volume left unused. The problem, however, is that sections such as the one in Figure 8 are few and far apart when the weight heuristic is implemented.

Figure 17 and 18 illustrate how a packed pallet obtained with the heuristic can look like when the sections are treated as boxes and how each of the sections is packed, respectively.

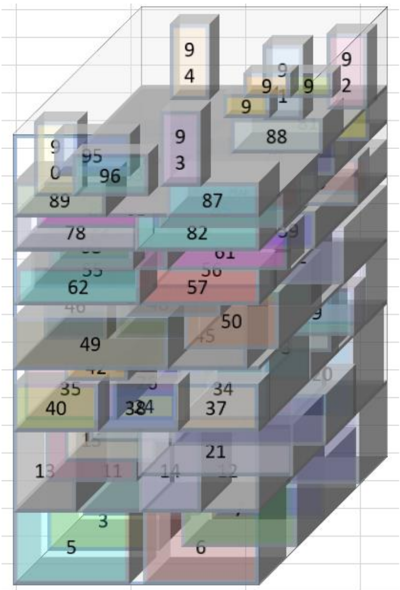


Figure 18. The packing inside each of the sections (made by the authors)

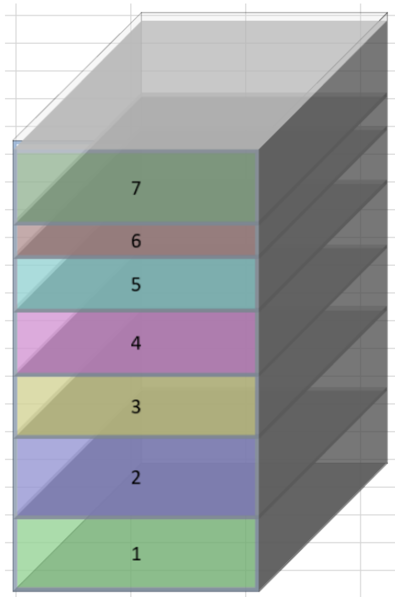


Figure 17. The packed sections creating the packed pallet (made by the authors)

Looking at the sections-based packing conducted in Figure 18, the packing comes across as nearly perfect, with an unused volume of only $78{,}720$ $(\boldsymbol{1000cm^3})$ and fill rate equivalent to 95%. The reality of the situation, however, is far from achieving the same results. As Figure 11 illustrates, each of the sections illustrated in Figure 17 has much-unused volumes. The unused volume of the packed pallets is in reality $500{,}538$ $(\boldsymbol{1000cm^3})$, and the fill rate is only approximately 65%. Figure 17 clearly illustrates the problem with locking each of the sections with the "invisible" sheet. Boxes already packed in a section cannot be moved between the sections, even though the boxes may fit better in another section. The boxes that are already packed in a section, therefore, becomes static when new sections are opened for packing. This then leads to a lack of dynamic packing, where new boxes considered can have an impact on the boxes already packed, restacking or repacking the pallet.

The inadequate volume utilisation and the substantial differences in the gaps between the boxes may cause problems when stability is concerned. The lack of the boxes in the sections to be uniformly packed, probably causes significant differences in the weight distribution and creates pallets that are not stable.

One of the main driving forces behind the decision of basing the selection process on the weight of the boxes was the idea that this would put the heaviest boxes as close to the bottom of the pallets as possible. What was discovered during the packing process, however, is that lighter boxes can create heavier sections than what the heaviest boxes can. This can happen because a lot more of the lighter boxes can fit into a section, than what is seen with the sections packed with the heavier boxes. A heavier section containing light boxes may then be put underneath a lighter section containing a small number of heavy boxes. The heaviest section is then packed underneath the lighter section, making it possible for heavy boxes to be put on top of light boxes, subsequently making it possible for deformation and damages to be inflicted on boxes.

6.3.2 Changes, height and Plasticity

This part is dedicated to showing what small changes to the heuristic may do to the packing of the pallets, and how changing from a weight perspective to a height perspective regarding the packing selection in the heuristic may contribute to better volume utilisation. Order 7 will be the focal point and the order used for comparing the packing before and after the changes to the heuristic, and when implementing the height. It will also be shown how the implementation of plasticity can change the packing.

Implementation of changes to the heuristic

The heuristic approach created states that the selection of boxes should be made based on the weight of said boxes, starting with the heaviest and working its way down to the lightest. This selection process has left much volume left in the sections that have been packed. To improve the packing conducted by the heuristic, small changes can be introduced. Table 2 compares the heuristic approach, the packing conducted by the workers and a third packing possibility created by using the heuristic along with having the height of the boxes in mind.

Table 2. Comparison between the heuristic, changes and workers (made by the authors)

Order Number	Unused Volume ($1000cm^3$)		
	Model	**Small Change**	**Workers**
ORDER 7	500,538	470,743	122,510
	477,539	475,164	42,431
	506,528	1.072,981	41,117
	486,915	150,080	813,945
	469,284	260,542	

With the implemented changes, the model still packs the boxes on five pallets, same as with the weight heuristic approach. Apart from the same number of pallets, the packing and the available volume is more like the one done by the workers, in which the volume is not as uniformly distributed as is the case with the heuristic approach based on weight. Figure 19 visualises the fill rates of the three different approaches and shows the similarities between the heuristic with changes and the workers.

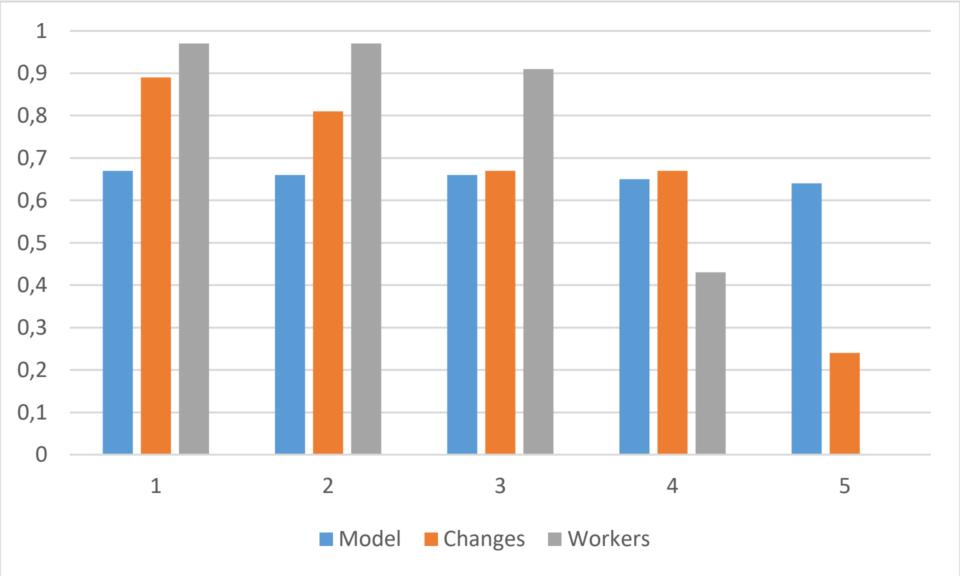


Figure 19. Fill rates for model, changes and workers (made by the authors)

The fill rates show that the small changes to the heuristic do not create only half-packed pallets, like the once seen when the heuristic based on weight is used but packs pallets more like the once created by the workers. More of the volume is filled up in the first pallets, better utilising the pallets at the start of the packing. Even though the same amounts of pallets are used, the new packing possibility shows that even small changes can make a significant impact on how the pallets are packed.

The initial heuristic managed to pack all the boxes in the order into 28 sections, yielding a total height of all the sections equal to 669 cm. Implementing the changes to the heuristic manages to reduce the number of sections to 27 and subsequently the total height of all the sections by 71 cm. Meaning that some of the sections packed are better utilised than what the initial heuristic is capable of.

The height as box selection criteria

Previous heuristic procedures like the once proposed by Martello, Pisinger, and Vigo (2000) have based the box selection process on the volume of the boxes, more specifically the non-increasing volume. As could be seen with the implementation of the heuristic with box selection based on weight, the approach is not capable of competing with the packing conducted by the workers. The small changes implemented to the heuristic, somewhat considering height, improved the packing of the pallets when the utilisation of the first packed pallets are concerned. Height, therefore, seems like the better selection criteria when the heuristic created is to be used. Table 3 compares, using order 7, the packing conducted by the heuristic based on weight, on height and the packing conducted by the workers.

Table 3. Comparison between heuristic based on weight, based on height and packing by workers (made by the authors)

Order Number	**Unused Volume ($1000cm^3$)**		
	Weight	**Height**	**Workers**
ORDER 7	500,538	398,752	122,510
	477,539	221,591	42,431
	506,528	178,893	41,117
	486,915	220,768	813,945
	469,284		

Substituting the selection of boxes based on weight with a selection based on height yields a solution that packs the boxes on the same number of pallets that the workers are capable of. Using the same heuristic, only substituting the box selecting criteria, the model provides a packing that has 1.420,800 $(1000cm^3)$ better volume utilisation. This number is equivalent to precisely one pallet.

The problem with the static boxes occurring in the packing with the weight selected boxes, do not provide the same issues when the boxes are based on weight. Given that the boxes are chosen alongside boxes with approximately the same height, they are already pared with boxes more like one another. This means that the large amount of unused volume that occur when considering the height from the top of the boxes to the top of the section do not happen when the height is the selection criteria. Since the heuristic approach uses the combined area of the boxes selected to see how many that can fit into the section, clustering the boxes based on height can lead to the same utilisation in the footprint of the pallets but a much better utilisation of the volume.

Basing the box selection on height produces the same average fill rate as the packing done by the workers, given that both manage to put the same set of boxes on four pallets. Even though the fill rates are the same, the heuristic approach with height does not manage to achieve the same high numbers that the workers manage. As shown in figure 18, the workers manage to get a fill rate of above 90% on three out of the four pallets. The heuristic manages to get above 80% on three, and a little above 70% on the fourth. This could mean that, even though the same number of pallets are used, the new heuristic approach may not be able to fill each of the pallets to its best capacity.

One of the pallets packed according to height is visualised in Figure 20. Comparing this pallet to the one packed in Figure 18 shows much improvement in volume utilisation.

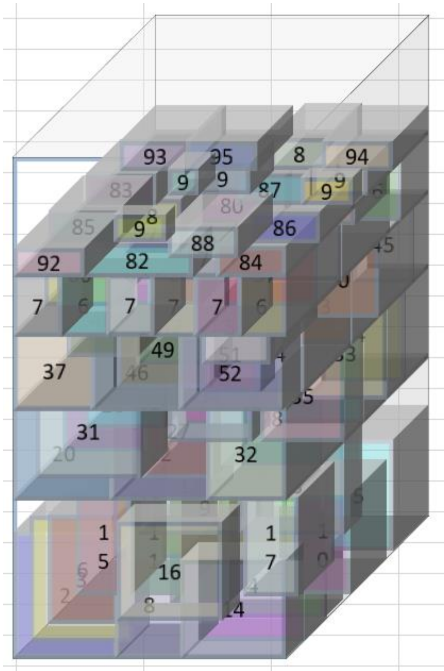


Figure 20. The packed order based on the height selection (made by the authors)

The bottom section visualised is the section containing the box that is a lot taller than the rest of the boxes in the order. Even though this box was paired with boxes similar to itself, the section is not better utilised than the one packed with the weight heuristic. However, what the section does better, in this case, is that it puts tall boxes in the same section, meaning that the other sections do not need to be as tall as the one that can be seen in the weight heuristic. This is only the case in instances where taller boxes are in a considerable minority compared to the rest of the boxes, as is the case with the order that has been packed. The other sections are visibly packed in a manner that creates proper volume utilisation. Available volume above the boxes are almost non-existing, and so is the case with the available volume between the boxes. This shows that the boxes are more uniformly distributed throughout the section, which means that the packing comes across as more stable and well packed compared to the one seen in Figure 18.

Even though the packing can be more stable with the height-based selection, the lack of independent weight consideration may cause damage and deformation to the packing, making this approach not that suitable for implementation in real-life.

Plasticity and what it can do to the packing

As mentioned in the literature regarding pallet packing, plasticity can be implemented into packing to obtain new packing possibilities. One of the things that can be implemented is, ever so slightly, changing the dimensions of the pallet. In one of the orders, eight rectangular homogeneous boxes are to be packed onto a pallet. Using both the weight- and the height heuristic makes it so that these boxes are packed in two sections, subsequently yielding a solution that packs the pallet in two layers. Looking closely at the data, it is apparent that the dimensions of the boxes are a few millimetres too long on both the length and the width to be able to put them in one layer. Figure 22 visualises the pallet packed with the original dimensions, while Figure 21 shows the packed pallet with the dimensions of the pallet ever so slightly changed.

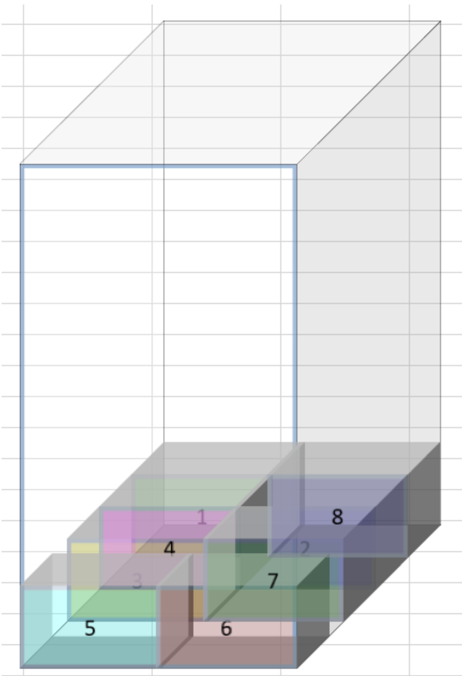


Figure 21. Order packed with the implementation of plasticity (made by the authors)

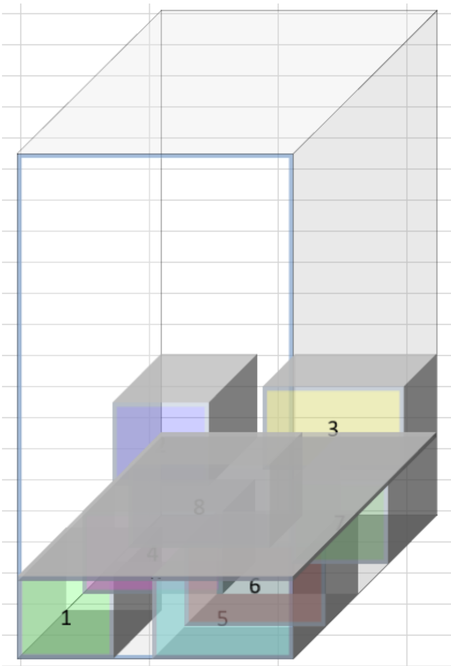


Figure 22. Order packed with the heuristic (made by the authors)

Just by extending the width of the pallet by 0,4 cm, all the boxes can be packed in one single layer. It is reasonable to think that this is the packing that has been conducted by the workers as well, as 0,2 cm overhang on each side of the pallet is a small amount and hardly visible to the naked eye. The plasticity shown in Figure 21 is something that needs to be taken into consideration when looking at the differences between the packing conducted by the heuristic implemented to the model and the workers.

Another possible implementation of plasticity is the change of height to the sections packed in the heuristic. One of the sections created using the weight-based heuristic had a height of 22,3 cm, given that the largest box in this section had this height. The smallest boxes selected into this section had a height of 11,2 cm. Given the height of the section, two of these boxes could not be packed on top of each other, being 0,1 cm to tall. Changing the section to have a height of 22,4 allowed for the possibility of having at most double the number of boxes such as these packed into the section. Making it possible to pack more items into the section, creating better volume utilisation.

7. Conclusion

Packing of containers, bins and pallets has substantial worth across a vast number of industries, spanning from warehouse-management and container stowing to goods transport across sea, road, air and railway. The industries it operates are industries with high uncertainties and complex nature. The problem considered in this thesis has focused on implementing weight as an essential factor in the packing of pallets. It has been a study dedicated to finding possible methods of packing that could compete, or even outperform, the packing conducted by experienced workers.

A new heuristic approach has been developed to check if a weight-based selection of boxes can create feasible and viable packing options. The experiments and packing have been conducted using a mixed integer linear model created for three-dimensional packing problems. The validity of the model has been tested on a small problem and found to be satisfactory. Modifications have been done to the model when the heuristic approaches concerning weight and height have been implemented, disregarding weight distribution, stability and support. The first approach researched the locking of the coordinates of already packed boxes. The second approach used the construction heuristic created to pack pallets based on the weight of the boxes. The third approach considered how the height of the boxes could be implemented to achieve better packing. The final approach took a little look at how plasticity can be implemented to the packing.

Locking the coordinates of already packed boxes turned out to not yield satisfactory packing solutions. The approach needed more than thrice the number of pallets as the workers did, thus creating packing that utilised little of the volume available on each pallet. The optimisation of each of the packing procedures did only take the packages already packed into consideration, disregarding the boxes left to be packed. This made for a packing that only optimised the packing at that exact stage, disregarding the entirety of the pallets and the boxes.

Weight as the decision variable in the box selection process has shown to be far from optimal in the instances we have tested, creating packet pallets with large amounts of unused volume. It has been shown that the weight selection approach has problems with clustering the boxes when they are strongly heterogeneous. The notion of packing the heaviest boxes as near the bottom as possible, so to not deform or destroy lighter boxes, has not been shown to work the way it was intended. They have sections with heavy boxes being put on top of sections with light boxes. Stability and packing patterns have also shown to be less than satisfactory, considering the poorly packed pallets. The issues faced are mostly concerning the heuristics clustering of boxes with significant differences in shape and size and creating uneven packed sections and pallets.

With the order tested, height as the decision variable in the box selection process has been shown to yield better volume utilisation than the ones based on weight. Creating better-packed pallets and more uniformly packed sections. The height selection approach has in instances tested shown to be able to pack the same set of boxes on the same number of pallets as the workers. This has been done without regard to the final condition of the boxes, and the implementation of a height-only box selection criteria can have difficulties when implemented in real-life instances given the practical relevance of box condition.

Implementing plasticity to the order tested has shown that small changes to the pallet dimensions, as well as the height of the sections packed, can change the packing of even very small problems. Plasticity in the constraints considered may create packing possibilities that more resembles how the packing is done in real life.

The experiments conducted and the instances tested in this thesis has shown that the approaches implemented have a hard time competing with the packing conducted by the workers. Locking the coordinates of boxes already packed has shown to create solutions that are not even close to what the workers are capable of. The weight-based heuristic fared better than the locked coordinate procedure but could still not compete with the workers. Basing the box selection process on height is the only approach that managed to pack the same set of boxes on the same number of pallets as the workers, at least in the instances tested. Even though this is the case, it was only achieved when disregarding the condition of the boxes.

The computations and experiments conducted in this thesis have shown that it will, with any approach, be hard to compete with the workers, and even harder to outperform them. The fill rates with the packing conducted by the workers can seem to be too high to improve, at least when looking at pallet reduction.

7.1 Limitations

Although the research conducted in this thesis managed to answer the problem it set out to test, limitations do exist. Firstly, the time available only made it possible to test the approaches to a small number of instances. Secondly, the transferability of the approaches conducted may not be existing towards other industries with different data and goals, given that all the instances tested stems from one particular industry. Thirdly, time is also a factor when concerned with the time allowed for the model to conduct the packing. Allowing the models to run for a more extended period could yield different solutions than what is obtained in this thesis. Finally, the computation study only researches a small number of possible packing methods, which, could lead to findings that are only relatable in small instances.

7.2 Suggestions for future research

The limitations suggested makes room for further research on three-dimensional bin packing with weight restrictions and improvements that could help the packing further along. Better improvements to the weight and height-based selection approaches could be implemented, clustering the boxes based on both in a more significant matter than what the approaches in this thesis have been able to do. An approach similar to the infill box approach investigated in the literature review could also be of value in future research, making it possible for items to be taken out of their initial boxes and placed in between said boxes. More research could also be conducted across different industries, trying to find packing solutions that could either be implemented in more than one industry or that could inspire some industries to look at applications not yet implemented and tried out. A closer look at the effects of plasticity could also make the field of packing more applicable to real-life scenarios.

Reference list

Alonso, M. T., R. Alvarez-Valdes, M. Iori, F. Parreño, and J. M. Tamarit. 2017. "Mathematical models for multicontainer loading problems." *Omega* 66:106-117. doi: https://doi.org/10.1016/j.omega.2016.02.002.

Amiouny, Samir V., John J. Bartholdi, III, John H. Vande Vate, and Jixian Zhang. 1992. "Balanced Loading." *Operations Research* 40 (2):238.

Bansal, N., A. Caprara, and M. Sviridenko. 2006. "Improved approximation algorithms for multidimensional bin packing problems." 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), 21-24 Oct. 2006.

Bartók, Tamás, and Csanád Imreh. 2011. "Pickup and delivery vehicle routing with multidimensional loading constraints." *Acta Cybernetica (Szeged)* 20 (1):17-33.

Bischoff, E. E. 2006. "Three-dimensional packing of items with limited load bearing strength." *European Journal of Operational Research* 168 (3):952-966. doi: https://doi.org/10.1016/j.ejor.2004.04.037.

Bischoff, E. E., F. Janetz, and M. S. W. Ratcliff. 1995. "Loading pallets with non-identical items." *European Journal of Operational Research* 84 (3):681-692. doi: https://doi.org/10.1016/0377-2217(95)00031-K.

Bischoff, E. E., and M. S. W. Ratcliff. 1995. "Issues in the development of approaches to container loading." *Omega* 23 (4):377-390. doi: https://doi.org/10.1016/0305-0483(95)00015-G.

Bortfeldt, Andreas. 2012. "A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints." *Computers & Operations Research* 39 (9):2248-2257. doi: https://doi.org/10.1016/j.cor.2011.11.008.

Bortfeldt, Andreas, and Gerhard Wäscher. 2013. "Constraints in container loading – A state-of-the-art review." *European Journal of Operational Research* 229 (1):1-20. doi: https://doi.org/10.1016/j.ejor.2012.12.006.

Chekuri, C., and S. Khanna. 2004. "On Multidimensional Packing Problems." *SIAM Journal on Computing* 33 (4):837-851. doi: 10.1137/S0097539799356265.

Chen, C. S., S. M. Lee, and Q. S. Shen. 1995. "An analytical model for the container loading problem." *European Journal of Operational Research* 80 (1):68-76. doi: https://doi.org/10.1016/0377-2217(94)00002-T.

Chien, Chen-Fu, and Jing-Feng Deng. 2004. "A container packing support system for determining and visualizing container packing patterns." *Decision Support Systems* 37 (1):23-34. doi: https://doi.org/10.1016/S0167-9236(02)00192-6.

Crainic, Teodor Gabriel, Guido Perboli, and Roberto Tadei. 2008. "Extreme point-based heuristics for three-dimensional bin packing." *Informs Journal on computing* 20 (3):368-384. doi: https://doi.org/10.1287/ijoc.1070.0250.

Delorme, Maxence, Manuel Iori, and Silvano Martello. 2018. "BPPLIB: a library for bin packing and cutting stock problems." *Optimization Letters* 12 (2):235-250.

den Boef, Edgar , Jan Korst, Silvano Martello, David Pisinger, and Daniele Vigo. 2005. "Erratum to “the three-dimensional bin packing problem”: robot-packable and orthogonal variants of packing problems." *Operations Research* 53 (4):735-736. doi: https://doi.org/10.1287/opre.1050.0210.

Dyckhoff, Harald. 1990. "A typology of cutting and packing problems." *European Journal of Operational Research* 44 (2):145-159. doi: https://doi.org/10.1016/0377-2217(90)90350-K.

Haessler, Robert W, and F Brian Talbot. 1990. "Load planning for shipments of low density products." doi: http://dx.doi.org/10.1016/0377-2217(90)90364-H.

Hifi, Mhand, Imed Kacem, Stéphane Nègre, and Lei Wu. 2010. "A linear programming approach for the three-dimensional bin-packing problem." *Electronic Notes in Discrete Mathematics* 36:993-1000.

Hill, Rodney. 1998. *The mathematical theory of plasticity*. Vol. 11: Oxford university press.

Junqueira, Leonardo, Reinaldo Morabito, and Denise Sato Yamashita. 2012. "Three-dimensional container loading models with cargo stability and load bearing constraints." *Computers & Operations Research* 39 (1):74-85. doi: https://doi.org/10.1016/j.cor.2010.07.017.

Kaas, J. H. 2001. "Neural Plasticity." In *International Encyclopedia of the Social & Behavioral Sciences*, edited by Neil J. Smelser and Paul B. Baltes, 10542-10546. Oxford: Pergamon.

Korte, Bernhard, and Jens Vygen. 2006. *Combinatorial optimization : theory and algorithms*. 3rd ed. ed. Vol. 21, *Algorithms and combinatorics*. Berlin: Springer.

Liu, D. S., K. C. Tan, S. Y. Huang, C. K. Goh, and W. K. Ho. 2008. "On solving multiobjective bin packing problems using evolutionary particle swarm optimization." *European Journal of Operational Research* 190 (2):357-382. doi: https://doi.org/10.1016/j.ejor.2007.06.032.

Martello, Silvano, David Pisinger, and Daniele Vigo. 2000. "The three-dimensional bin packing problem." *Operations Research* 48 (2):256-267.

Martello, Silvano, David Pisinger, Daniele Vigo, Edgar Den Boef, and Jan Korst. 2007. "Algorithm 864: General and robot-packable variants of the three-dimensional bin packing problem." *ACM Transactions on Mathematical Software (TOMS)* 33 (1):1-12.

Monaco, Maria Flavia, Marcello Sammarra, and Gregorio Sorrentino. 2014. "The Terminal-Oriented Ship Stowage Planning Problem." *European Journal of Operational Research* 239 (1):256-265. doi: https://doi.org/10.1016/j.ejor.2014.05.030.

Nemhauser, George L., and Laurence A. Wolsey. 1988. *Integer and combinatorial optimization*, *Wiley-Interscience series in discrete mathematics and optimization*. New York: Wiley.

Paquay, C., M. Schyns, and S. Limbourg. 2016. "A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application." *International Transactions in Operational Research* 23 (1-2):187-213. doi: 10.1111/itor.12111.

Ramos, António G., Elsa Silva, and José F. Oliveira. 2018. "A new load balance methodology for container loading problem in road transportation." *European Journal of Operational Research* 266 (3):1140-1152. doi: https://doi.org/10.1016/j.ejor.2017.10.050.

Sheng, Liu, Zhao Hongxia, Dong Xisong, and Cheng Changjian. 2016. "A heuristic algorithm for container loading of pallets with infill boxes." *European Journal of Operational Research* 252 (3):728-736. doi: https://doi.org/10.1016/j.ejor.2016.01.025.

Trivella, Alessio, and David Pisinger. 2016. "The load-balanced multi-dimensional bin-packing problem." *Computers & Operations Research* 74:152-164. doi: https://doi.org/10.1016/j.cor.2016.04.020.

Tsai, Russell D, Eric M Malstrom, and Way Kuo. 1991. "A three dimensional dynamic palletizing heuristic." In *Material Handling'90*, 181-201. Berlin, Heidelberg: Springer.

Wäscher, Gerhard, Heike Haußner, and Holger Schumann. 2007. "An improved typology of cutting and packing problems." *European Journal of Operational Research* 183 (3):1109-1130. doi: https://doi.org/10.1016/j.ejor.2005.12.047.

Yang, Guang. 2016. "gbp: a bin packing problem solver.", Last Modified 19.11.16, accessed 22.11.18. https://cran.r-project.org/web/packages/gbp/vignettes/gbp-vignette.html.

Appendix

Sets, parameters and variables

Notation/Description/mathematical model	**AMPL code**
Indexation:	
$i, j = an\ arbitrary\ box$	
$k = an\ arbitrary\ pallet$	
$l = an\ arbitrary\ vertex$	
$a - an\ arbitrary\ axis$	
$b - an\ arbitrary\ \ side\ of\ the\ box$	
Sets	
$O^A = set\ of\ axes\ \{X, Y, Z\}$	set AXIS
$O^B = set\ of\ box\ sides\ \{L, W, H\}$	set SIDES
Parameters:	
$n = number\ of\ boxes\ to\ be\ packed$	param n >= 0
$m = number\ of\ possible\ pallets\ to\ use$	param m >= 0
$v = number\ of\ vertices\ for\ a\ box$	param v >= 0
Box dimensions	
$l_i = length\ of\ box\ i, \forall\ i = 1, \dots, n$	param length {1..n} >= 0
$w_i = width\ of\ box\ i, \forall\ i = 1, \dots, n$	param width {1..n} >= 0
$h_i = height\ of\ box\ i, \forall\ i = 1, \dots, n$	param height {1..n} >= 0
$\gamma_i = \ weight\ of\ box\ i, \forall\ i = 1, \dots, n$	param weight {1..n} >= 0
Pallet dimensions	
$l' = max\ length\ of\ pallet$	param pallet_length >= 0
$w' = max\ width\ of\ pallet$	param pallet_width >= 0
$h' = max\ height\ of\ pallet$	param pallet_height >= 0

$\gamma' = max\ weight\ of\ pallet$	param pallet_weight >= 0
Allowable range	
$\alpha^{l'} = allowable\ range\ for\ central\ gravity\ length$	param allowable_range_l
$\alpha^{w'} = allowable\ range\ for\ central\ gravity\ width$	param allowable_range_w
$\alpha^{h'} = allowable\ range\ for\ central\ gravity\ height$	param allowable_range_h

Variables:

Notation/Description/mathematical model	**AMPL code**
$P_{ik} = 1\ if\ box\ i\ is\ on\ pallet\ k, 0\ otherwise, i = 1, \dots, n, k = 1, \dots, m$	var Box_on_pallet {i in 1..n, k in 1..m} binary
$U_k = 1\ if\ pallet\ k\ is\ used, 0\ otherwise, k = 1, \dots, m$	var Used_pallet {k in 1..m} binary
Coordinates	
$X_i = front\ left\ bottom\ coordinate\ X\ for\ box\ i, i = 1, \dots, n$	var X {i in 1..n} >= 0
$Y_i = front\ left\ bottom\ coordinate\ Y\ for\ box\ i, i = 1, \dots, n$	var Y {i in 1..n} >= 0
$Z_i = front\ left\ bottom\ coordinate\ Z\ for\ box\ i, i = 1, \dots, n$	var Z {i in 1..n} >= 0
$X_i' = back\ right\ top\ coordinate\ X\ for\ box\ i, i = 1, \dots, n$	var Xr{i in 1..n} >= 0
$Y_i' = back\ right\ top\ coordinate\ Y\ for\ box\ i, i = 1, \dots, n$	var Yr{i in 1..n} >= 0
$Z_i' = back\ right\ top\ coordinate\ Z\ for\ box\ i, i = 1, \dots, n$	var Zr{i in 1..n} >= 0
Orientation/positioning	
$R_{iab} = 1\ if\ the\ box\ side\ b\ for\ box\ i\ is\ parallel\ the\ axis\ a,$ $0\ otherwise, i = 1, \dots, n,\ a \in O^A, b \in O^B$	var Position {i in 1..n, a in AXIS, b in SIDES} binary
Relative position for box i and j	
$A_{ij} = 1\ if\ box\ i\ is\ on\ the\ right\ side\ of\ box\ j, 0\ otherwise,$ $i, j = 1, \dots, n$	var Right{i in 1..n, j in 1..n} binary

$B_{ij} = 1\ if\ box\ i\ is\ behind\ box\ j, 0\ otherwise, i, j = 1, \ldots, n$	var Behind{i in 1..n, j in 1..n} binary
$C_{ij} = 1\ if\ box\ i\ is\ above\ box\ j, 0\ otherwise, i, j = 1, \ldots, n$	var Above{i in 1..n, j in 1..n} binary
Stability	
$G_i = 1\ if\ box\ i\ is\ on\ the\ ground, 0\ otherwise, i = 1, \ldots, n$	var Ground{i in 1..n} binary
$Q_{ij} = 0\ if\ box\ j\ has\ the\ suitable\ height\ to\ support\ box\ i,$ $1\ otherwise,\ \ i, j = 1, \ldots, n$	var Suitable_height{i in 1..n, j in 1..n} binary
$I_{ij} =$ $0\ if\ the\ projections\ on\ the\ XY\ plane\ of\ the\ box\ i\ and\ j\ have\ a$ $non\text{-}empty\ intersection, 1\ otherwise,\ \ i, j = 1, \ldots, n$	var No_empty_intersection{i in 1..n, j in 1..n} binary
$S_{ij} = 1\ if\ box\ j\ supports\ box\ i\ and\ are\ on\ the\ same\ pallet,$ $0\ otherwise,\ \ i, j = 1, \ldots, n$	var Support{i in 1..n, j in 1..n} binary
$V_{ijl} =$ $1\ if\ vertex\ l\ for\ box\ i\ is\ supported\ by\ box\ j, 0\ otherwise,$ $i, j = 1, \ldots, n,\ \ l = 1, \ldots, v$	Vertex_support{i in 1..n, j in 1..n, l in 1..v} binary
Box vertex support	
$\beta_{ij}^{X} = 1\ if\ X_j >\ X_i, 0\ or\ 1\ otherwise, i, j = 1, \ldots, n$	var X_less{i in 1..n, j in 1..n} binary
$\beta_{ij}^{Y} = 1\ if\ Y_j > Y_i,\ \ 0\ or\ 1\ otherwise, i, j = 1, \ldots, n$	var Y_less{i in 1..n, j in 1..n} binary
$\beta_{ij}^{X'} = 1\ if\ Xr_j >\ Xr_i, 0\ or\ 1\ otherwise, i, j = 1, \ldots, n$	var Xr_less{i in 1..n, j in 1..n} binary
$\beta_{ij}^{Y'} = 1\ if\ Yr_j >\ Yr_i, 0\ or\ 1\ otherwise, i, j = 1, \ldots, n$	var Yr_less{i in 1..n, j in 1..n} binary
Absolute variables	

$T_{ij} = Represent\ absolute\ value\ \left\|Z_j' - Z_i\right\|, i, j = 1, \dots, n$	var Absolute {i in 1..n, j in 1..n} >= 0
$\delta_{ij} = 1\ if\ Z_j' \geq Z_i, 0\ otherwise, i, j = 1, \dots, n$	var Binary {i in 1..n, j in 1..n} binary
Gravity variables	
$\gamma_i^X = X\ coordinate\ for\ mass\ i, i = 1, \dots, n$	var X_mass_coordinate {1..n} >= 0
$\gamma_i^Y = Y\ coordinate\ for\ mass\ i, i = 1, \dots, n$	var Y_mass_coordinate {1..n} >= 0
$\gamma_i^Z = Z\ coordinate\ for\ mass\ i, i = 1, \dots, n$	var Z_mass_coordinate {1..n} >= 0
$\sigma_{ik}^X =$ $for\ selecting\ only\ boxes\ that\ are\ on\ pallet\ k\ and\ have\ X\ coordinates,\ i = 1, \dots, n,\ k = 1, \dots, m$	var Xij {i in 1..n, k in 1..m} >= 0
$\sigma_{ik}^Y =$ $for\ selecting\ only\ boxes\ that\ are\ on\ pallet\ k\ and\ have\ Y\ coordinates,\ i = 1, \dots, n,\ k = 1, \dots, m$	var Yij {i in 1..n, k in 1..m} >= 0
$\sigma_{ik}^Z =$ $for\ selecting\ only\ boxes\ that\ are\ on\ pallet\ k\ and\ have\ Z\ coordinates,\ i = 1, \dots, n,\ k = 1, \dots, m$	var Zij {i in 1..n, k in 1..m} >= 0

Mathematical model	AMPL name
Objective	
(1) $\min \sum_{k=1,\dots,m} l'w'h'U_k - \sum_{i=1,\dots,n,} l_i w_i h_i$	Unused_Volume

S.t.

Mathematical model	AMPL name
Logical/bounds	
(2) $\sum_{k=1,\dots,m} P_{ik} = 1, i = 1, \dots, n$	Box_on_one_pallet {i in 1..n}
(3) $\sum_{i=1,\dots,n} \gamma_i P_{ik} \le \gamma' U_k, k = 1, \dots, m$	Weight_capacity {k in 1..m}
(4a) $X_i' \le \sum_{k=1,\dots,m} l' P_{ik}, i = 1, \dots, n$	Cont_Size_Xr {i in 1..n}
(4b) $Y_i' \le \sum_{k=1,\dots,m} w' P_{ik}, i = 1, \dots, n$	Cont_Size_Yr {i in 1..n}
(4c) $Z_i' \le \sum_{k=1,\dots,m} h' P_{ik}, i = 1, \dots, n$	Cont_Size_Zr {i in 1..n}
Orientation/rotation	
(5a) $\sum_{a \in O^A} R_{iab} = 1, i = 1, \dots, n, b \in O^B$	Rotation1 {i in 1..N, b in SIDES}
(5b) $\sum_{b \in O^B} R_{iab} = 1, i = 1, \dots, n, a \in O^A$	Rotation2 {i in 1..N, a in AXIS}
(5c) $X_i' - X_i = l_i R_{i,\mathrm{X},\mathrm{L}} + w_i R_{i,\mathrm{X},\mathrm{W}} + h_i R_{i,\mathrm{X},\mathrm{H}}, i = 1, \dots, n$	Rotation3 {i in 1..N, a in AXIS, b in SIDES}
(5d) $Y_i' - Y_i = l_i R_{i,\mathrm{Y},\mathrm{L}} + w_i R_{i,\mathrm{Y},\mathrm{W}} + h_i R_{i,\mathrm{Y},\mathrm{H}}, i = 1, \dots, n$	Rotation4 {i in 1..N, a in AXIS, b in SIDES}
(5e) $Z_i' - Z_i = l_i R_{i,\mathrm{Z},\mathrm{L}} + w_i R_{i,\mathrm{Z},\mathrm{W}} + h_i R_{i,\mathrm{Z},\mathrm{H}}, i = 1, \dots, n$	Rotation5 {i in 1..N, a in AXIS, b in SIDES}
(5f) $R_{i,\mathrm{Z},\mathrm{H}} = 1, i = 1, \dots, n$	Side_up {i in 1..N, a in AXIS, b in SIDES}
Overlapping	
(6a) $A_{ij} + A_{ji} + B_{ij} + B_{ji} + C_{ij} + C_{ji} \ge P_{ik} + P_{jk} - 1, i, j = 1, \dots, n, k = 1, \dots, m: i \ne j$	Overlap_check {i in 1..n, j in 1..n, k in 1..m: i<>j}
(6b) $X_j' \le X_i + (1 - A_{ij}) l', i, j = 1, \dots, n: i \ne j$	Overlapping1 {i in 1..n, j in 1..n: i<>j}

(6c) $X_i + 1 \le X_j' + A_{ij}l', i, j = 1, \dots, n\colon i \ne j$	Overlapping2 {i in 1..n, j in 1..n: i<>j}
(6d)$Y_j' \le Y_i + \left(1 - B_{ij}\right)w', i, j = 1, \dots, n\colon i \ne j$	Overlapping3 {i in 1..n, j in 1..n: i<>j}
(6e) $Y_i + 1 \le Y_j' + B_{ij}w', i, j = 1, \dots, n\colon i \ne j$	Overlapping4 {i in 1..n, j in 1..n: i<>j}
(6f) $Z_j' \le Z_i + \left(1 - C_{ij}\right)h', i, j = 1, \dots, n\colon i \ne j$	Overlapping5 {i in 1..n, j in 1..n: i<>j}
(6g) $Z_i \le Z_j' + C_{ij}h', i, j = 1, \dots, n\colon i \ne j$	Overlapping6 {i in 1..n, j in 1..n: i<>j}
Defining box stability	
(7) $\sum_{j=1,\dots,n:\, i \ne j} \sum_{l=1,\dots,v} V_{ijl} \ge 3\,(1 - G_i), i = 1, \dots, n$	Stability {i in 1..N}
(8) $Z_i \le (1 - G_i)h', i = 1, \dots, n$	On_the_ground {i in 1..N}
Defining suitable height	
(9a)$Z_j' - Z_i \le T_{ij}, i, j = 1, \dots, n\colon i \ne j$	Absolute1 {i in 1..n, j in 1..n: i<>j }
(9b)$Z_i - Z_j' \le T_{ij}, i, j = 1, \dots, n\colon i \ne j$	Absolute2 {i in 1..n, j in 1..n: i<>j }
(9c)$T_{ij} \le Z_j' - Z_i + 2\left(1 - \delta_{ij}\right)h', i, j = 1, \dots, n\colon i \ne j$	Absolute3 {i in 1..n, j in 1..n: i<>j }
(9d)$T_{ij} \le Z_i - Z_j' + 2\delta_{ij}h', i, j = 1, \dots, n\colon i \ne j$	Absolute4 {i in 1..n, j in 1..n: i<>j }
(10a) $Q_{ij} \le T_{ij}, i, j = 1, \dots, n\colon i \ne j$	Absolute5 {i in 1..n, j in 1..n: i<>j }
(10b) $T_{ij} \le Q_{ij}h', i, j = 1, \dots, n\colon i \ne j$	Absolute6 {i in 1..n, j in 1..n: i<>j }
Continue stability	
(11a) $I_{ij} \le A_{ij} + A_{ji} + B_{ij} + B_{ji}, i, j = 1, \dots, n\colon i \ne j$	Shared_projection1 {i in 1..n, j in 1..n: i<>j}
(11b)$A_{ij} + A_{ji} + B_{ij} + B_{ji} \le 2I_{ij}, i, j = 1, \dots, n\colon i \ne j$	Shared_projection2 {i in 1..n, j in 1..n: i<>j}
(12a) $\left(1 - S_{ij}\right) \le Q_{ij} + I_{ij}, i, j = 1, \dots, n\colon i \ne j$	Support_bottom_top1 {i in 1..n, j in 1..n: i<>j}

(12b) $Q_{ij} + I_{ij} \leq 2(1 - S_{ij}), i, j = 1, \dots, n: i \neq j$	Support_bottom_top2 {i in 1..n, j in 1..n: i<>j}
(13a) $P_{ik} - P_{ki} \leq (1 - S_{ij}), i, j = 1, \dots, n,\ k = 1, \dots, m: i \neq j$	Box_support_same_pallet1 {i in 1..n, j in 1..n, k in 1..m: i<>j}
(13b) $P_{ki} - P_{ik} \leq (1 - S_{ij}), i, j = 1, \dots, n,\ k = 1, \dots, m: i \neq j$	Box_support_same_pallet2 {i in 1..n, j in 1..n, k in 1..m: i<>j}
(14) $V_{ijl} \leq S_{ij}, i, j = 1, \dots, n,\ l = 1, \dots, v: i \neq j$	Certified_support {i in 1..n, j in 1..n, l in 1..v: i<>j}
(15a) $X_j \leq X_i + \beta_{ij}^X l', i, j = 1, \dots, n: i \neq j$	Lapping1 {i in 1..n, j in 1..n: i<>j}
(15b) $Y_j \leq Y_i + \beta_{ij}^Y w', i, j = 1, \dots, n: i \neq j$	Lapping2 {i in 1..n, j in 1..n: i<>j}
(15c) $X_i' \leq X_j' + \beta_{ij}^{X'} l', i, j = 1, \dots, n: i \neq j$	Lapping3 {i in 1..n, j in 1..n: i<>j}
(15d) $Y_i' \leq Y_j' + \beta_{ij}^{Y'} w', i, j = 1, \dots, n: i \neq j$	Lapping4 {i in 1..n, j in 1..n: i<>j}
(16a) $\beta_{ij}^X + \beta_{ij}^Y \leq 2(1 - V_{ij1}), i, j = 1, \dots, n, l = 1, \dots, v: i \neq j$	Box_support1 {i in 1..n, j in 1..n, l in 1..v: i<>j}
(16b) $\beta_{ij}^Y + \beta_{ij}^{X'} \leq 2(1 - V_{ij2}), i, j = 1, \dots, n, l = 1, \dots, v: i \neq j$	Box_support2 {i in 1..n, j in 1..n, l in 1..v: i<>j}
(16c) $\beta_{ij}^{X'} + \beta_{ij}^{Y'} \leq 2(1 - V_{ij3}), i, j = 1, \dots, n, l = 1, \dots, v: i \neq j$	Box_support3 {i in 1..n, j in 1..n, l in 1..v: i<>j}
(16d) $\beta_{ij}^X + \beta_{ij}^{Y'} \leq 2(1 - V_{ij4}), i, j = 1, \dots, n, l = 1, \dots, v: i \neq j$	Box_support4 {i in 1..n, j in 1..n, l in 1..v: i<>j}
Weight distribution	
(17a) $\gamma_i^X = \frac{X_i + X_i'}{2}, i = 1, \dots, n$	Box_mass_coordinate_X {i in 1..n}
(17b) $\gamma_i^Y = \frac{Y_i + Y_i'}{2}, i = 1, \dots, n$	Box_mass_coordinate_Y {i in 1..n}
(17c) $\gamma_i^Z = \frac{Z_i + Z_i'}{2}, i = 1, \dots, n$	Box_mass_coordinate_Z {i in 1..n}

(18a) $\sigma_{ik}^{X} \leq l'P_{ik}, i = 1, \dots, n, k = 1, \dots, m$	Range_X1 {i in 1..n, k in 1..m}
(18b) $\sigma_{ik}^{X} \leq \gamma_{i}^{X}, i = 1, \dots, n, k = 1, \dots, m$	Range_X2 {i in 1..n, k in 1..m}
(18c) $\sigma_{ik}^{X} \geq \gamma_{i}^{X} - l'(1 - P_{ik}), i = 1, \dots, n, k = 1, \dots, m$	Range_X3 {i in 1..n, k in 1..m}
(19a) $(\frac{l'}{2} - \alpha^{l'})(\sum_{i=1,\dots,n} \gamma_i P_{ik}) \leq \sum_{i=1,\dots,n} \sigma_{ik}^{X} \gamma_i, k = 1, \dots, m$	Neighbourhood_X1 {k in 1..m}
(19b) $\sum_{i=1,\dots,n} \sigma_{ik}^{X} \gamma_i \leq \left(\frac{l'}{2} + \alpha^{l'}\right)\left(\sum_{i=1,\dots,n} \gamma_i P_{ik}\right),$ $k = 1, \dots, m$	Neighbourhood_X2 {k in 1..m}
(20a) $\sigma_{ik}^{Y} \leq w'P_{ik}, i = 1, \dots, n, k = 1, \dots, m$	Range_Y1 {i in 1..n, k in 1..m}
(20b) $\sigma_{ik}^{Y} \leq \gamma_{i}^{Y}, i = 1, \dots, n, k = 1, \dots, m$	Range_Y2 {i in 1..n, k in 1..m}
(20c) $\sigma_{ik}^{Y} \geq \gamma_{i}^{Y} - w'(1 - P_{ik}), i = 1, \dots, n, k = 1, \dots, m$	Range_Y3 {i in 1..n, k in 1..m}
(21a) $(\frac{w'}{2} - \alpha^{w'})(\sum_{i=1,\dots,n} \gamma_i P_{ik}) \leq \sum_{i=1,\dots,n} \sigma_{ik}^{Y} \gamma_i,$ $k = 1, \dots, m$	Neighbourhood_Y1 {k in 1..m}
(21b) $\sum_{i=1,\dots,n} \sigma_{ik}^{Y} \gamma_i \leq \left(\frac{w'}{2} + \alpha^{w'}\right)\left(\sum_{i=1,\dots,n} \gamma_i P_{ik}\right),$ $k = 1, \dots, m$	Neighbourhood_Y2 {k in 1..m}
(22a) $\sigma_{ik}^{Z} \leq h'P_{ik}, i = 1, \dots, n, k = 1, \dots, m$	Range_Z1 {i in 1..n, k in 1..m}
(22b) $\sigma_{ik}^{Z} \leq \gamma_{i}^{Z}, i = 1, \dots, n, k = 1, \dots, m$	Range_Z2 {i in 1..n, k in 1..m}
(22c) $\sigma_{ik}^{Z} \geq \gamma_{i}^{Z} - h'(1 - P_{ik}), i = 1, \dots, n, k = 1, \dots, m$	Range_Z3 {i in 1..n, k in 1..m}
(23a) $0 \leq \sum_{i=1,\dots,n} \sigma_{ik}^{Z} \gamma_i, k = 1, \dots, m$	Neighbourhood_Z1 {k in 1..m}
(23b) $\sum_{i=1,\dots,n} \sigma_{ik}^{Z} \gamma_i \leq \alpha^{h'}\left(\sum_{i=1,\dots,n} \gamma_i P_{ik}\right), k = 1, \dots, m$	Neighbourhood_Z2 {k in 1..m}

(24) $S_{ij} + S_{ji} \leq 1, i, j = 1, \dots, n: i \neq j$	Logic_support {i in 1..n, j in 1..n: i<>j}
(25) $U_k \leq U_{k-1}, k = 2, \dots, m$	Next_pallet {k in 2..m}:
(26a) $P_{ik} \in \{0, 1\}, i = 1, \dots, n, k = 1, \dots, m$	
(26b) $U_k \in \{0, 1\}, k = 1, \dots, m$	
(26c) $G_i \in \{0, 1\}, i = 1, \dots, n$	
(26d) $R_{iab} \in \{0, 1\}, i = 1, \dots, n, a \in O^A, b \in O^B$	
(26e) $A_{ij}, B_{ij}, C_{ij}, Q_{ij}, I_{ij}, S_{ij}, \beta_{ij}^X, \beta_{ij}^Y, \beta_{ij}^{X'}, \beta_{ij}^{Y'}, \delta_{ij} \in \{0, 1\}, i, j = 1, \dots, n$	
(26f) $V_{ijl} \in \{0, 1\}, i, j = 1, \dots, n, l = 1, \dots, v$	
(27a) $X_i, Y_i, Z_i, X_i', Y_i', Z_i', \gamma_i^X, \gamma_i^Y, \gamma_i^Z \geq 0, i = 1, \dots, n$	
(27b) $T_{ij} \geq 0, i, j = 1, \dots, n$	
(27c) $\sigma_{ik}^X, \sigma_{ik}^Y, \sigma_{ik}^Z \geq 0, i = 1, \dots, n, k = 1, \dots, m$	

AMPL-code

```
set AXIS;   # set of axis: X,Y,Z
set SIDES;  # set of boxsides: L,W,H

param n >= 0;     # number of boxes to be packed
param m >= 0;     # number of pallets
param v >= 0;     # number of vertices


param length{1..n} >= 0;      # length of box i
param width{1..n} >= 0;       # weight of box i
param height{1..n} >= 0;      # height of box i
param weight{1..n} >= 0;      # weight of box i


param pallet_length >= 0;     # max length of pallet
param pallet_width >= 0;      # max width of pallet
param pallet_height >= 0;     # max height of pallet
param pallet_weight >= 0;     # max weight of pallet

param E >= 0;           # small number

var Box_on_pallet{i in 1..n, k in 1..m} binary;
# 1 if box i is on pallet k, 0 otherwise
var Used_pallet{k in 1..m} binary;  # 1 if pallet k is used, 0 otherwise

var X{i in 1..n} >= 0;  # front left bottom coordinate X for box i
var Y{i in 1..n} >= 0;  # front left bottom coordinate Y for box i
var Z{i in 1..n} >= 0;  # front left bottom coordinate Z for box i
var Xr{i in 1..n} >= 0; # back right top coordinate X for box i
var Yr{i in 1..n} >= 0; # back right top coordinate Y for box i
var Zr{i in 1..n} >= 0; # back right top coordinate Z for box i

var Right{i in 1..n, j in 1..n} binary;
# 1 if box i is on the right side of box j, 0 otherwise
var Behind{i in 1..n, j in 1..n} binary;
# 1 if box i is behind box j, 0 otherwise
var Above{i in 1..n, j in 1..n} binary;
# 1 if box i is above box j, 0 otherwise

var Position {i in 1..n, a in AXIS, b in SIDES} binary;
# 1 if the boxside b for box i is paralell to the axis a, 0 otherwise

var Ground{i in 1..n} binary;
# 1 if box i is on the ground, 0 otherwise

var Suitable_height{i in 1..n, j in 1..n} binary;
# 0 if box j has the suitable height to support box i, 1 otherwise
var No_empty_intersection{i in 1..n, j in 1..n} binary;
# 0 if the projections on the XY plane of the boxs i and j have a
nonempty intersection, 1 otherwise.
var Support{i in 1..n, j in 1..n} binary;
# 1 if box j supports box i and are on the same pallet, 0 otherwise
var Vertex_support{i in 1..n, j in 1..n, l in 1..v} binary;
# 1 if vertex l for box i is supported by box j, 0 otherwise

var X_less{i in 1..n, j in 1..n} binary;  # 0 if X j <= X i, 1 otherwise
var Y_less{i in 1..n, j in 1..n} binary;  # 0 if Y j <= Y i, 1 otherwise
```

```
var Xr_less{i in 1..n, j in 1..n} binary; # 0 if Xr j <= Xr i, 1
otherwise
var Yr_less{i in 1..n, j in 1..n} binary; # 0 if Yr j <= Yr i, 1
otherwise
minimize Unused_Volume:
sum{k in 1..m} Used_pallet[k] * pallet_length * pallet_width *
pallet_height - sum{i in 1..n} (length[i] * width[i] * height[i]);
#  Minimizing total volume: Total volume of pallets used - total volume
of boxes

subject to Next_pallet {k in 2..m}:
Used_pallet[k] <= Used_pallet[k-1];
# Says that if a pallet is used, pallet k-1 has to be used as well. This
is for the seccond pallet to m.

subject to Box_on_one_pallet {i in 1..n}:
sum{k in 1..m} Box_on_pallet[i,k] = 1;
# each box is placed on one and only one pallet


subject to Weight_capacity {k in 1..m}:
sum{i in 1..n} weight[i] * Box_on_pallet[i,k] <= pallet_weight *
Used_pallet[k];
# weight on pallet k cannot exceed pallet k's weight capacity



subject to Cont_Size_Xr {i in 1..n}:
Xr[i] <= sum{k in 1..m} pallet_length * Box_on_pallet[i,k];
# The back right top X coordinate for box i cannot exceed the length of
pallet

subject to Cont_Size_Yr {i in 1..n}:
Yr[i] <= sum{k in 1..m} pallet_width * Box_on_pallet[i,k];
# The back right top Y coordinate for box i cannot exceed the width of
pallet

subject to Cont_Size_Zr {i in 1..n}:
Zr[i] <= sum{k in 1..m} pallet_height * Box_on_pallet[i,k];
# The back right top Z coordinate for box i cannot exceed the height of
pallet



subject to Rotation1 {i in 1..n, b in SIDES}:
sum{a in AXIS} Position[i,a,b] = 1;
# box side b for box i must be parallel to one and only one axis.

subject to Rotation2 {i in 1..n, a in AXIS}:
sum{b in SIDES} Position[i,a,b] = 1;
# axis a for box i must be parallel to one and only one box side.

subject to Rotation3 {i in 1..n, a in AXIS, b in SIDES}:
Xr[i] - X[i] = Position[i,"X","L"] * length[i] + Position[i,"X","W"] *
width[i] + Position[i,"X","H"] * height[i];
# Position 3, 4 and 5 says that the boxes can rotate orthogonally on the
pallet
# The difference between back right top X coordinate and front left
bottom X coordinate for box i must be equal to the way box i is
positioned
```

```
subject to Rotation4 {i in 1..n, a in AXIS, b in SIDES}:
Yr[i] - Y[i] = Position[i,"Y","L"] * length[i] + Position[i,"Y","W"] *
width[i] + Position[i,"Y","H"] * height[i];
# Position 3, 4 and 5 says that the boxes can rotate orthogonally on the
pallet
subject to Rotation5 {i in 1..n, a in AXIS, b in SIDES}:
Zr[i] - Z[i] = Position[i,"Z","L"] * length[i] + Position[i,"Z","W"] *
width[i] + Position[i,"Z","H"] * height[i];
# Position 3, 4 and 5 says that the boxes can rotate orthogonally on the
pallet

subject to Side_up {i in 1..n, a in AXIS, b in SIDES}:
Position[i,"Z","H"] = 1;
# Box side "H" (height) for box i must be parallel to the "Z" axis



subject to Overlap_check {i in 1..n, j in 1..n, k in 1..m: i<>j}:
Right[i,j] + Right[j,i] + Behind[i,j] + Behind[j,i] + Above[i,j] +
Above[j,i] >= (Box_on_pallet[i,k] + Box_on_pallet[j,k]) - 1;
# An overlap can only happen if box i and box j are on the same pallet
# If the box i and box j are not right, behind or above each other
(LHS=0), then box i and box j are not on the same pallet

subject to Overlapping1 {i in 1..n, j in 1..n: i<>j}:
Xr[j] <= X[i] + (1 - Right[i,j]) * pallet_length;
# Defines Right

subject to Overlapping2 {i in 1..n, j in 1..n: i<>j}:
X[i] + E <= Xr[j] + Right[i,j] * pallet_length;
# Defines Right

subject to Overlapping3 {i in 1..n, j in 1..n: i<>j}:
Yr[j] <= Y[i] + (1 - Behind[i,j]) * pallet_width;
# Defines Behind

subject to Overlapping4 {i in 1..n, j in 1..n: i<>j}:
Y[i] + E <= Yr[j] + Behind[i,j] * pallet_width;
# Defines Behind

subject to Overlapping5 {i in 1..n, j in 1..n: i<>j}:
Zr[j] <= Z[i] + (1 - Above[i,j]) * pallet_height;
# Defines Above

subject to Overlapping6 {i in 1..n, j in 1..n: i<>j}:
Z[i] + E <= Zr[j] + Above[i,j] * pallet_height;
# Defines Above. Added by us because Above did not work correctly.



subject to Stability {i in 1..n}:
sum{j in 1..n, l in 1..v: i<>j} Vertex_support[i,j,l] >= 3 * (1 -
Ground[i]);
# ensures that at least 3 vertices are supported if box i is not on the
ground. I and j must be different.


subject to On_the_ground {i in 1..n}:
Z[i] <= (1 - Ground[i]) * pallet_height;
```

```
# If ground is equal to one, Z coordinate for box i has to be zero. Then
the box i is on the ground.
# if ground i equals 1, then box i is on the ground



var Absolute {i in 1..n, j in 1..n} >= 0; # Represent absolute value
|Zrj-Zi|
var Binary {i in 1..n, j in 1..n} binary; # 1 if Zrj>=Zi,0 otherwise


# Defines the variable Suitable_height by using the absolute value |Zr-
Zi| and binary variable 1 if Zr>=Z

subject to Absolute1 {i in 1..n, j in 1..n: i<>j}:
Zr[j] - Z[i] <= Absolute[i,j];
#

subject to Absolute2 {i in 1..n, j in 1..n: i<>j}:
Z[i] - Zr[j] <= Absolute[i,j];
#

subject to Absolute3 {i in 1..n, j in 1..n: i<>j}:
Absolute[i,j] <= Zr[j] - Z[i] + 2 * pallet_height * (1 - Binary[i,j]);
#

subject to Absolute4 {i in 1..n, j in 1..n: i<>j}:
Absolute[i,j]  <= Z[i] - Zr[j] + 2 * pallet_height * Binary[i,j];
#

subject to Absolute5 {i in 1..n, j in 1..n: i<>j}:
E*Suitable_height[i,j] <= Absolute[i,j] ;
# If the absolute value is zero, the suitable height must be 0. If
suitable height is 1, then absolute value is bigger or equal to one.

subject to Absolute6 {i in 1..n, j in 1..n: i<>j}:
Absolute[i,j] <= Suitable_height[i,j] * pallet_height;
# If suitable_height is zero, the absolute value must be 0. If the
absolute value is bigger than zero, then suitable height must be one.


subject to Shared_projection1 {i in 1..n, j in 1..n: i<>j}:
No_empty_intersection[i,j] <= Right[i,j] + Right[j,i] + Behind[i,j] +
Behind[j,i];
# If there are a non-empty intersection between box i and box j, then box
i cannot be right of box j or otherwise and box i cannot be behind box j
or otherwise.


subject to Shared_projection2 {i in 1..n, j in 1..n: i<>j}:
Right[i,j] + Right[j,i] + Behind[i,j] + Behind[j,i] <= 2 *
No_empty_intersection[i,j];
# If there are a non-empty intersection between box i and box j, then box
i cannot be right of box j or otherwise and box i cannot be behind box j
or otherwise.

subject to Support_bottom_top1 {i in 1..n, j in 1..n: i<>j}:
(1 - Support[i,j]) <= Suitable_height[i,j] + No_empty_intersection[i,j];
# If box i is supported by box j, then box j has the suitable height to
support box i and there is a non-empty intersection between the boxes.
```

```
subject to Support_bottom_top2 {i in 1..n, j in 1..n: i<>j}:
Suitable_height[i,j] + No_empty_intersection[i,j] <= 2 * (1 -
Support[i,j]);
# If box i is supported by box j, then box j has the suitable height to
support box i and there is a non-empty intersection between the boxes.

subject to Box_support_same_bin1 {i in 1..n, j in 1..n, k in 1..m: i<>j}:
Box_on_pallet[i,k] - Box_on_pallet[j,k] <= 1 - Support[i,j];
# Box i can only be supported by box j if they are on the same pallet

subject to Box_support_same_bin2 {i in 1..n, j in 1..n, k in 1..m: i<>j}:
Box_on_pallet[j,k] - Box_on_pallet[i,k] <= 1 - Support[i,j];
# Box i can only be supported by box j if they are on the same pallet

subject to Certified_support {i in 1..n, j in 1..n, l in 1..v: i<>j}:
Vertex_support[i,j,l] <= Support[i,j];
# If vertex l for box i is supported by box j, then box j supports box i

subject to Lapping1 {i in 1..n, j in 1..n: i<>j}:
X[j] <= X[i] + X_less[i,j] * pallet_length;
# Defines X_less. If lhs is bigger than the first term on rhs, then
X_less has to take value 1. Else it can choose itself as long as it works
with box support constraints.

subject to Lapping2 {i in 1..n, j in 1..n: i<>j}:
Y[j] <= Y[i] + Y_less[i,j] * pallet_width;
# Defines Y_less. If lhs is bigger than the first term on rhs, then
Y_less has to take value 1. Else it can choose itself as long as it works
with box support constraints.

subject to Lapping3 {i in 1..n, j in 1..n: i<>j}:
Xr[i] <= Xr[j] + Xr_less[i,j] * pallet_length;
# Defines Xr_less. If lhs is bigger than the first term on rhs, then
Xr_less has to take value 1. Else it can choose itself as long as it
works with box support constraints.

subject to Lapping4 {i in 1..n, j in 1..n: i<>j}:
Yr[i] <= Yr[j] + Yr_less[i,j] * pallet_width;
# Defines Yr_less. If lhs is bigger than the first term on rhs, then
Yr_less has to take value 1. Else it can choose itself as long as it
works with box support constraints.



subject to Box_support1 {i in 1..n, j in 1..n: i<>j}:
X_less[i,j] + Y_less[i,j] <= sum {l in 1..v} 2 * (1 -
Vertex_support[i,j,1]);
# XY plane for box i and j if vertex l is supported by box j.

subject to Box_support2 {i in 1..n, j in 1..n: i<>j}:
Y_less[i,j] + Xr_less[i,j] <= sum {l in 1..v} 2 * (1 -
Vertex_support[i,j,2]);
# XY plane for box i and j if vertex l is supported by box j.

subject to Box_support3 {i in 1..n, j in 1..n: i<>j}:
Xr_less[i,j] + Yr_less[i,j] <= sum {l in 1..v} 2 * (1 -
Vertex_support[i,j,3]);
# XY plane for box i and j if vertex l is supported by box j.

subject to Box_support4 {i in 1..n, j in 1..n: i<>j}:
```

```
X_less[i,j] + Yr_less[i,j] <= sum {l in 1..v} 2 * (1 -
Vertex_support[i,j,4]);
# XY plane for box i and j if vertex l is supported by box j.



########################################################################
#
############################Weight
constraints############################
########################################################################
#

param allowable_range_l >=0; #
param allowable_range_w >=0; #
param allowable_range_h >=0; #

var X_mass_coordinate {1..n} >= 0;  # X coordinate for mass i
var Y_mass_coordinate {1..n} >= 0;  # Y coordinate for mass i
var Z_mass_coordinate {1..n} >= 0;  # Z coordinate for mass i
var Xij {i in 1..n, k in 1..m} >= 0;      # For selecting only boxes that
are on pallet k and have X coordinates
var Yij {i in 1..n, k in 1..m} >= 0;      # For selecting only boxes that
are on pallet k and have Y coordinates
var Zij {i in 1..n, k in 1..m} >= 0;      # For selecting only boxes that
are on pallet k and have Z coordinates



subject to Box_mass_coordinate_X {i in 1..n}:
X_mass_coordinate[i] = (X[i] + Xr[i])/2;
# X coordinate of mass for item i, assuming weight is uniformly
distributed in box

subject to Box_mass_coordinate_Y {i in 1..n}:
Y_mass_coordinate[i] = (Y[i] + Yr[i])/2;
# Y coordinate of mass for item i, assuming weight is uniformly
distributed in box

subject to Box_mass_coordinate_Z {i in 1..n}:
Z_mass_coordinate[i] = (Z[i] + Zr[i])/2;
# Z coordinate of mass for box i, assuming weight is uniformly
distributed in box

# Xij = Box_on_cont * X_mass_coordinate (triple equal sign). Real
variable.
# Range X1, X2 and X3 are linear constraints. The Xij definition has to
satisy X1, X2 and X3.

subject to Range_X1 {i in 1..n, k in 1..m}:
Xij[i,k] <= pallet_length * Box_on_pallet[i,k];
#

subject to Range_X2 {i in 1..n, k in 1..m}:
Xij[i,k] <= X_mass_coordinate[i];
#

subject to Range_X3 {i in 1..n, k in 1..m}:
Xij[i,k] >= X_mass_coordinate[i] - pallet_length * (1 -
Box_on_pallet[i,k]);
```

```
#

subject to Neighbourhood_X1 {k in 1..m}:
((pallet_length / 2) - allowable_range_l) * (sum {i in 1..n} weight[i] *
Box_on_pallet[i,k]) <= sum {i in 1..n} Xij[i,k] * weight[i];
# Ensures that the X central gravity of the pallet is in the
neighbourhood of L/2.


subject to Neighbourhood_X2 {k in 1..m}:
sum {i in 1..n} Xij[i,k] * weight[i] <=((pallet_length / 2) +
allowable_range_l) * (sum {i in 1..n} weight[i] * Box_on_pallet[i,k]);
#


# Yij = Box_on_cont * Y_mass_coordinate (triple equal sign). Real
variable.
# Range Y1, Y2 and Y3 are linear constraints. The Yij definition has to
satisfy Y1, Y2 and Y3.

subject to Range_Y1 {i in 1..n, k in 1..m}:
Yij[i,k] <= pallet_width * Box_on_pallet[i,k];
#

subject to Range_Y2 {i in 1..n, k in 1..m}:
Yij[i,k] <= Y_mass_coordinate[i];
#

subject to Range_Y3 {i in 1..n, k in 1..m}:
Yij[i,k] >= Y_mass_coordinate[i] - pallet_width * (1 -
Box_on_pallet[i,k]);
#


subject to Neighbourhood_Y1 {k in 1..m}:
((pallet_width / 2) - allowable_range_w) * (sum {i in 1..n} weight[i] *
Box_on_pallet[i,k]) <= sum {i in 1..n} Yij[i,k] * weight[i];
# Ensures that the Y central gravity of the pallet is in the
neighbourhood of W/2.

subject to Neighbourhood_Y2 {k in 1..m}:
sum {i in 1..n} Yij[i,k] * weight[i] <=((pallet_width / 2) +
allowable_range_w) * (sum {i in 1..n} weight[i] * Box_on_pallet[i,k]);
#

# Zij = Box_on_cont * Z_mass_coordinate (triple equal sign). Real
variable.
# Range Z1, Z2 and Z3 are linear constraints. The Zij definition has to
satisfy Z1, Z2 and Z3.

subject to Range_Z1 {i in 1..n, k in 1..m}:
Zij[i,k] <= pallet_height * Box_on_pallet[i,k];
#

subject to Range_Z2 {i in 1..n, k in 1..m}:
Zij[i,k] <= Z_mass_coordinate[i];
#

subject to Range_Z3 {i in 1..n, k in 1..m}:
Zij[i,k] >= Z_mass_coordinate[i] - pallet_height * (1 -
Box_on_pallet[i,k]);
```

```
#

subject to Neighbourhood_Z1 {k in 1..m}:
0 <= sum {i in 1..n} Zij[i,k] * weight[i];
# Ensures that the Z central gravity can lie as low as possible.

subject to Neighbourhood_Z2 {k in 1..m}:
sum {i in 1..n} Zij[i,k] * weight[i] <= allowable_range_h * (sum {i in
1..n} weight[i] * Box_on_pallet[i,k]);
#

##############################New
constraint##############################

subject to Logic_support {i in 1..n, j in 1..n: i<>j}:
Support[i,j] + Support[j,i] <= 1;
# Box i cannot be supported by box j if box j is supported by box i and
otherwise
```