Modelling and Optimisation of Space Allocation and layout Problems

Deemah M. Aljuhani

Supervisor: Professor Lazaros Papageorgiou

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department Chemical Engineering

University College London

July 18, 2021

I, Deemah M. Aljuhani, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Acknowledgements

I am grateful for the opportunity that lead me to start my PhD. journey! This research journey would not have been possible without the help of many people. First and foremost, I would like to thank my supervisor Professor. Lazaros Papageorgiou for accepting me to collaborate on this research with him. I appreciate the patience, support, and guidance given throughout this time. I am also grateful to King Abdulaziz University and the Saudi Culture Bureau in London for the funding and scholarship. I appreciate all the support given by my colleagues in the CPSE research group, especially Jude, Vassilis, Ioannis, Robert, and Yena. I would also like to thank my friends Amjad and Mariam for always being there for me, supporting me throughout these years, especially during the Covid 19 Pandemic! I'm am so grateful for your friendship.

I would like to express my greatest gratitude to my Mom and Dad for constantly pushing me forward, for all the love, prayers, and support. I am also grateful in the first and last place to my husband Majdi, for the continues support, love, and sacrifice! Thank you for always being there! I would not have been at this stage I am today without you. And of course, a special thanks goes to my lovely and adoring children, Faris, Yazan, and Maria, for believing in me and supporting me over the days and nights I was partially absent.

Abstract

This thesis investigates the development of optimisation-based, decision-making frameworks for allocation problems related to manufacturing, warehousing, logistics, and retailing. Since associated costs with these areas constitute significant parts to the overall supply chain cost, mathematical models of enhanced fidelity are required to obtain optimal decisions for i) pallet loading, ii) assortment, and iii) product shelf space, which will be the main research focus of this thesis. For the Manufactures Pallet loading problems (MPLP), novel single- and multi-objective Mixed Integer Linear Programming (MILP) models have been proposed, which generate optimal layouts of improved 2D structure based on a block representation. The approach uses a Complexity Index metric, which aids in comparing 2 pallet layouts that share the same pallet size and number of boxes loaded but with different box arrangements. The proposed algorithm has been tested against available data-sets in literature. In the area of Assortments (optimal 2D packing within given containers) , an iterative MILP algorithm has been developed to provide a diverse set of solutions within pre-specified range of key performance metrics. In addition, a basic software prototype, based on AIMMS platform, has been developed using a user-friendly interface so as to facilitate user interaction with a visual display of the solutions obtained. In Shelf- Space Allocation (SSAP) problem, the relationship between the demand and the retailer shelf space allocated to each item is defined as space elasticity. Most of existing literature considers the problem with stationary demand and fixed space elasticities. In this part of the thesis, a dynamic framework has been proposed to forecast space elasticities based on historical data using standard time-series methodologies. In addition, an optimisation mathematical model has been implemented using the forecasted space elasticities to provide the retailer with optimal shelf space thus resulting into closer match between supply and demand and increased profitability. The applicability and effectiveness of the proposed framework is demonstrated through a number of tests and comparisons against literature data-sets.

Impact Statment

In this research, a range of topics related to real-life supply chain operations has been considered, focusing on the area of manufacturing, warehousing, logistics, and retailing. Where novel techniques using mathematical programming models as well as dynamic time series linear models have been developed.

The first topic considered is in the area of the Pallet Loading Problem (PLP), where a specific type of problem known as the Manufacturer's Pallet Loading Problem (MPLP) has been considered. Two linear mathematical models have been developed, producing less complex layouts for the pallet loading, where the complexity of the pallet can be measured using a novel metric known as the complexity index. Such complexity-reduced layouts are essential in the manufacturing plants when speedy pallet construction is needed. This will affect the supply chain operation cost by allowing more pallets to be loaded in a shorter time frame.

The second problem in this research is in the domain of Cutting and Packing, where the problem considered is known as Design of Assortments. The problem statement has been developed by the 11th AIMMS-MOPTA Optimisation Modelling Conference committee. A Mixed Integer Linear Programming (MILP) algorithm has been developed to provide the user with diverse sets of assortments, using a platform known as AIMMS. A Graphical User Interface (GUI) has been created to allow the users to fine-tune the results with a visualisation interface. Such assortments problems are needed in many real-life applications such as paper, wood, textile cutting, and newspaper layouts.

The final problem discussed is the Shelf Space Allocation Problems (SSAP), where decisions on how much product to stock on shelves are essential. Such problems are seen in all retail businesses, from small shops, supermarkets to departmental stores. In this research, a dynamic framework using multiple standard time-series methodologies for forecasting space elasticises has been considered, with an application on real-life historical data.

All the problems considered in this research are essential in most supply chain and manufacturing facilities. However, the mathematical models can be adapted to different areas of research where similar problem structures apply. For example, the Manufacturer's Pallet Loading Problem (MPLP) can be applied to a wide range of areas such as and not limited to; plant layouts, cuboid satellite layouts, and microchips layouts for electrical equipment. The assortment problem can be applied to the textile, paper, wood industry, and online website layout management. Finally, the forecasting techniques used in the Shelf Space Allocation Problems (SSAP) could be applied to any area where forecasting is essential, whether related to pharmaceutical, food, or fashion industries.

Contents

List of Figures

List of Tables

Acronyms

ARIMA Auto-regressive Integrated Moving Average. 28, 108, 111, 113, 114, 117, 118, 121, 122, 127, 128, 133, 136

CP Cutting and Packing. 21, 22, 26, 27, 29, 30, 90, 91, 93, 135

CSCMP Council of Supply Chain Management Professionals. 19

DPLP Distributor's Pallet Loading Problem. 31, 33, 137

GUI Graphical User Interface. 28, 90, 105, 106

LR Linear Regression. 28, 108, 111, 113, 117, 118, 128, 133, 136

LSTM Long Short Term Memory. 11, 28, 108, 111, 114, 118, 123–125, 128, 132, 133, 136, 159

MILP Mixed Integer Linear Programming. 6, 15, 27, 30, 36, 45, 49, 61, 62, 76, 88, 90, 94, 95, 98, 106, 127, 134, 135

MPLP Manufacturer's Pallet Loading Problem. 6, 7, 27, 29, 31, 33, 36, 61, 88, 134, 136

PLP Pallet Loading Problem. 6, 26, 27, 33

SCM Supply Chain Management. 18–20

SSAP Shelf Space Allocation Problems. 7, 10, 24, 26, 28, 29, 108–111, 115, 116, 118, 127, 132–135

SVR Support Vector Regression. 28, 108, 111, 114, 118, 122, 123, 128, 133, 136

Chapter 1

Introduction

This chapter introduces the fundamental topics addressed throughout this thesis, defining the main terminologies and project objectives followed by the thesis outline.

1.1 Supply Chain

The supply chain architecture requires strategic decisions on where to place facilities, allocate resources, select and build supplier and distribution networks, and organize the supply chain interfaces between different parties in an uncertain world. In the early 1980s, the term Supply Chain Management (SCM) was initially introduced by Oliver et al. (1982), and since then, it has gained considerable attention. Researchers have attempted to offer a SCM structure since the early 1990s, where Bechtel and Jayaram (1997) have reviewed a wide range of literature and established specific SCM assumptions that need to be questioned in the future.

Shorter product life cycles and expanded product selection add to the expense and complexity of the supply chain. Instead of looking at the problem from the perspective of an individual organisation, outsourcing, globalisation, and market fragmentation made it necessary to look at the problem from the perspective of the whole supply chain. Furthermore, advances in information technology have facilitated real-time information exchange, cooperation, and decision-making amongst firms. According to Min et al. (2019) the focus of the SCM has shifted over the years to meet different market trends and technological advances; for example, the channel control has moved more toward the end-user or consumers. Where they no longer only seek product and service benefits but also price cuts and promotions, such changes have forced businesses with the help of technological advances to come up with innovative ways to fulfill those personalised requirements.

In general, the SCM definitions have varied throughout the literature, but they can all be classified into three divisions, according to Mentzer et al. (2001): a management philosophy, implementation of a management philosophy, and a set of management processes. For a comprehensive definition and according to the Council of Supply Chain Management Professionals (CSCMP) the SCM can be defined as follows (Vitasek, 2016):

"Supply Chain Management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies. Supply Chain Management is an integrating function with primary responsibility for linking major business functions and business processes within and across companies into a cohesive and high-performing business model. It includes all of the logistics management activities noted above, as well as manufacturing operations, and it drives coordination of processes and activities with and across marketing, sales, product design, finance and information technology."

Effective management of the supply chain demands a transition from managing specific roles to managing a series of integrated processes. And although there is still no "industry standard" for what these processes should, management has concluded that processes must be implemented first to optimise product flows in organisations. The importance of getting standard business processes in place allows the use of one common language between all organisations in the supply chain to connect with the processes of their companies with those of the other ones. Based on the Global Supply Chain Forum, there are eight primary processes as the following (Cooper et al., 1997):

- Customer Relationship Management
- Customer Service Management
- Demand Management
- Order Fulfillment
- Manufacturing Flow Management
- Procurement
- Product Development and Commercialisation
- Returns.

A full overlook on the SCM structure and the processes associated is presented in figure 1.1:

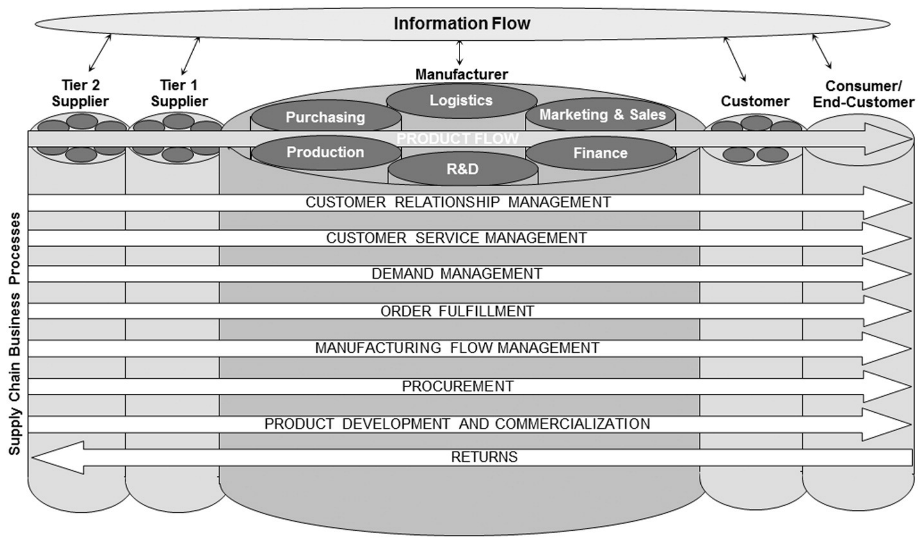


Figure 1.1: Supply Chain Management Structure and Processes (Lambert and Cooper, 2000)

1.2 Pallet Loading

For efficient supply chains, the configuration of pallets is a critical function, as efficient handling can reduce unnecessary operational expenditures, such as loading, storage, transport, and workforce, and as a result, it will successfully drive customer service satisfaction. The area has been given broad attention from both industry and academia in the past 30 years and is recognised as Cutting and Packing (CP), knapsack problems, strip packing, and bin packing. The problem has extensive applications in operations research, such as packing, cargo, transport, and warehouse management. Mass production is an interest to most manufacturing companies, and small pallet layout configuration changes can result in substantial space savings and dramatically reduce storage and transportation costs. In terms of workforce, the manual generation of layouts is expensive, so methods for packaging optimisation are being pursued.

Dyckhoff (1990) defined common features and properties for such problems and proposed a classification scheme to promote the sharing of knowledge across various disciplines. He differentiates between the problems of packing involving spatial dimensions and those involving non-spatial dimensions. The first category consists of problems with cutting and packing or loading, described by up to three dimensions in space, such as cutting stock problems, vehicle loading, and pallet loading. The other category covers abstract problems of ”cutting and packing,” including non-spatial aspects such as weight, time, or economic dimensions such as budgeting, coin change, and line balancing.

Four significant features characterise Dyckhoff’s classification system (Dyckhoff, 1990) for the Cutting and Packing (CP) problems; dimensionality; where it is considered as the most significant aspect, and it determines the minimum number of dimensions needed to characterise the pattern geometry ranging from one-dimensional up to N-dimensional problems. The type of assignment; describes whether it is appropriate to assign all objects and items or just a selection. Assortment of objects; where it distinguishes between problems with similar or different shapes of objects. And finally, the assortments of items or objects, where they can be either congruent sizes, few items of different sizes, many items of different sizes, or many items of relatively few sizes.

In general, the pallet loading problem aims to load as many items as possible into a pallet, and they could be divided into two types of problems; The Manufacturer's Pallet Loading Problem and the Distributor's Pallet Loading Problem. For the Manufacturer's Pallet Loading Problem, the manufacture has identical items and requires them to be loaded on identical pallet sizes, which will later be loaded onto trucks or containers for shipping. Usually, these problems require packing in real-time, immediately at the end of the production line, and in some particular cases, they could be done before shipping based on specific customer shipping requirements or restrictions. On the other hand, the Distributor's Pallet Loading Problem appears at distributors warehouse, where they have to fulfil customer orders containing several products of different sizes and load them on identical pallet sizes. Due to the nature of having specific customer order requirements at the distributor's warehouse that frequently change, such problems in this area are higher in cost than those at the manufacturer's sites.

1.3 Cutting and Packing

Cutting and Packing (CP) problems emerge in various industrial and logistical applications, such as metal, glass, textile, furniture, leather, and wood industries; for that reason, they have gained wide attention from the research community. All problems in this area have a similar underlying geometric framework, having two components, small and large, involving either cutting small pieces of objects from a big sheet/area or loading predetermined-sized pieces into a bigger container or bin. The objective of the problem can either be minimising the trim or space loss, decreasing material or transportation costs, maximising the number of goods loaded, or maximising the net profit.

The items to cut and the area of packing can either be homogeneous (identical in sizes) or heterogeneous (different sizes). The items to cut or pack can be regular shapes (rectangular) or irregular (non-rectangular, circles, etc.) shapes as shown in figure 1.2. Finally, such problems can be viewed from 1, 2, or multiple dimension aspects and might allow item orientation depending on the application and industry it appears within. The challenges in larger industries may be a mixture of two or more of these fundamental types of problems.

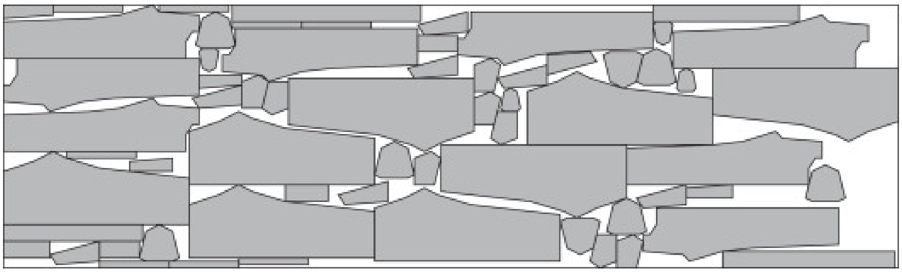


Figure 1.2: Illustration of irregular items (Leao et al., 2020)

In general the problems could be defined as the following (Hinxman, 1980), (Dowsland and Dowsland, 1992):

- Trim-loss problems: Where the main objective is to minimise the total cost of stock by placing the order list of items into predefined stock sheets.
- Assortment problems:Where the main objective is to select the best items from a stock sheet to fulfil an order list.
- Cutting stock problem : Where the main objective is to cut pieces from a set of stock sheets for a specific order list. This problem can be divided into two sub-problems; the Assortment problem (determining which sheets to maintain in stock) and the Trim loss problem (determining which cutting pattern to use to minimise waste).
- Knapsack problems: Where the main objective is to maximise the total value of the items packed or loaded. Given that each item in the order has a value assigned to it (Martello and Toth, 1990).

- Loading problem: Where the main objective is to maximise the number of boxes loaded into a pallet or minimise the total waste after loading. Such problems as discussed above can either be homogeneous or heterogeneous depending on which point of view they are being looked at; Manufactures or Distributors.

1.4 Shelf Space Allocation

Since shopping is a daily occurrence in our lives, people tend to spend a lot on daily shopping activities. Retail businesses such as supermarkets normally use this phenomenon in order to make the most of their profit. They often conduct research with the aim to influence the customers' purchasing decisions. The need for these retailers to maximise profits compels them to create a design for modelling customer's behaviour and setting simulation optimised frameworks. As an example, choosing which products to stock on the shelves and how much space to allocate for each product is a central and crucial decision as such decisions affect customer satisfaction and retailer's profit. Also when observing customer shopping behaviours, it has been noticed that product choices are influenced by many in-store factors, especially when unplanned purchases are made or certain loyal products are out of stock. Such behaviours have raised retailers awareness in cleverly displaying the products to achieve better product visibility and ensuring the right products at the right time to enhance the customer shopping experience (Chandon et al., 2009).

Such management problems, according to Galai et al. (2016) known as Shelf Space Allocation Problems (SSAP) have long been well-thought-out and studied by scientists and marketing experts. For example, the first published studies on Shelf Space Allocation Problems (SSAP) can be traced back to the seventies. However, because most optimisation models developed have practical restrictions, the research findings are unlikely to be used in practise. The restrictions are assumed to be caused by their simplicity as well as a lack of important practical characteristics. Furthermore, their restrictions are tied to several difficult-to-approximate parameters. As a result, there has been a mismatch between software applications, business operations, and research.

According to the US retailers survey (Keltz and Sterneckert, 2009), improvement of the overall profitability and sales, reducing the stock levels, improving product availability, and enhancing the customer shopping experience are the main drivers for shelf space planning activities. However, due to several problems in that area, the survey determined that the advantages obtained do not yet fulfil expectations. Shelf space planning usually follows assortment planning, which is done independently for each category (Koul et al., 2016). The assortment is frequently done with a mid-term planning horizon in mind. Because it is preferred by store space planning, this is sometimes referred to as micro-space planning. Individual strategies are becoming increasingly unfeasible as the number of stores grows. They usually result in the clustering of stores based on demand and space patterns. Existing customer-centric trends, on the other hand, argue that ”one strategy does not fit all” and defend store-specific space planning (Koul et al., 2016).

On the other hand, the retailers embrace the use of planograms to plan for product placement. A planogram is an illustration of a specific part of a store. It displays exactly where every product ought to be physically placed and the number of faces the product should hold; an example of the planogram could be seen in figure 1.3.

Today’s available commercial planogram tools are used mainly for visual and handling purposes and have excessive human interference and manual adjustment due to the limited existence of mathematical optimisation (Desmet and Renaudin, 1998; Dreze et al., 1994; Hansen et al., 2010; Hubner and Kuhn, 2012; Irion et al., 2011). Also, such software’s fail to include demand effects while planning the shelf allocation of the products.

Figure 1.3: Planogram Illustration in Retail shops

1.5 Project Objectives

This research work consists of real-life supply chain application problems, where the first problem is related to the area of Pallet Loading Problem (PLP), the second problem is in the domain of Cutting and Packing (CP), and the last is related to Shelf Space Allocation Problems (SSAP). Such problems generally consist of two sets of objects 'Large' and 'Small'; the 'Large' objects, or Bins refer to the pallet, shelf or any other medium that would be used as a base of loading and configuration, and 'Small' items usually refer to boxes, products or items that would be packed into the Bins. They share similar constraints in general, such as; loading the items within the bin size and avoiding any overlapping between items (sharing the same region in space); additional constraints could be including depending on the objective of the model.

All the addressed problems arise in the domain of manufacturing, warehousing, logistics, and retailing. Since transportation, warehousing, and storage costs contribute the most of the total logistics cost, and in order to increase the material handling efficiency in manufacturers, distributors, and retailers site, mathematical techniques are required to optimise the process of assortment, shelf space, and pallet loading. Thus, this work focuses on developing efficient new techniques using mathematical programming; single- and multi-objective optimisation, as well as dynamic time series linear methods to handle the problems in this area. The topics addressed in this study and the main contributions made by this dissertation are listed below.

1.5.1 Manufactures Pallet loading Problem (MPLP)

In the area of PLP, the Manufacturer's Pallet Loading Problem (MPLP) tackling homogeneous (identical) items have been investigated. Where the problem from a 2-Dimensional aspect using single- and multi-objective Mixed Integer Linear Programming (MILP) models with applicable constraints have been studied. A block approach; grouping boxes that share the same orientation along the x and y axis has been implemented, considering the complexity of pallet loading, ensuring optimal results are always found with less complex graphical layouts compared with existing literature data-sets. A novel complexity metric to calculate the complexity of pallet loading has been created; that provides a new way to compare 2 pallets of the same size with different graphical layouts. Also, a detailed comparison between the single- and multi-objective Mixed Integer Linear Programming models has been presented.

1.5.2 Cutting and Packing Problems (C&P)

In the area of Cutting and Packing (CP) a new problem statement defined as Automated Design of Assortments, presented for the first time in the 11th AIMMS-MOPTA Optimisation Modelling Competition has been investigated. Where a Mixed Integer Linear Programming (MILP) algorithm has been proposed to tackle the problem by providing the user with a variety of item assortment (diversity of solutions), ensuring the maximum number of items are packed into rectangular containers. Along with the variety of assortments generated by the model, the exact geometric locations of items packed are given. The validity of the proposed model has been tested using appropriate data-sets given by the competition community. The model was developed using AIMMS platform, where a user-friendly GUI has been developed to facilitate user interaction with an option to tune the near-optimality parameter and a graphical display of the solution assortments.

1.5.3 Shelf Space Allocation Problems (SSAP)

Such problems can be traced back to the seventies; however, the research outcome in terms of mathematical models developed has many realistic limitations preventing it from being applied and used in practice. The limitations are thought to occur due to their simplicity as well as lack of key practical features. In addition, their limitations are linked to numerous parameters that are not easy to approximate, space elasticises being one of the main parameters. As a result, misalignment between the software applications, business processes, and research are seen. From this perspective, a dynamic framework using a selection of Time series linear methods; Linear Regression (LR), Support Vector Regression (SVR), Auto-regressive Integrated Moving Average (ARIMA), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models are proposed to forecast the space elasticity utilising historical data. Where a comparison between all approaches is presented and to further verify the completeness of the model, the estimated space elasticities were used in the traditional SSAP model to compare the number of facing and the projected increase in sales against the historical data.

1.6 Thesis Outline

As different problems in the domain of manufacturing, retails, and logistics have been investigated, the remaining of the thesis is formatted in the following way; four main chapters, each corresponding to the development of the problem statement, where the literature review section for Chapter 2 and 3 is combined due to both problems being in the same area of research, Chapter 4 and 5 have their own literature survey. Each chapter is then followed by its preliminary results and concluding remarks, and a conclusion for the area of research is reported in Chapter 6.

Chapter 2, develops a single-objective model for the Manufacturer's Pallet Loading Problem (MPLP) in the 2-Dimensional aspect. The chapter begins with a literature review on the Pallet loading problem in general, moving towards specific MPLP problems and current research available. Next, the methodology, along with the main components of the proposed algorithm, is described in detail. Afterward, the theoretical and algorithmic frameworks are established, several case studies are employed to demonstrate of the validity of the model. Based on the case studies, a short discussion on the computational complexity of the proposed model is carried out. Chapter 3, presents a multi-objective optimisation method for the same problem statement, following the same structure as chapter 2, with an additional section of comparison between both the single- and multi-objective model determining the main differences and improvements.

Chapter 4, focuses on Assortment problems in the field of Cutting and Packing (CP), where the chapter starts by demonstrating related literature in the area, followed by the problem definition and the Mathematical formulation of the proposed model. Following that, the findings based on specific data sets acquired are shown. Finally, the findings of the study are presented.

Chapter 5, is dedicated to identifying and exploring the area of Shelf Space Allocation Problems (SSAP), where first a literature survey is conducted on the topic followed by an overview of the problem statement. Next, the Methodology to tackle the problem is proposed, followed by a section describing the data used, and finally, the results and conclusions with some insights of improvements and future work direction with remarks are presented. Finally, in Chapter 6, conclusions, recommendations, and possible future research directions are highlighted.

Chapter 2

Single-Objective Manufacture's Pallet Loading Problem

In this chapter, a novel Mixed Integer Linear Programming (MILP) model that generates layouts of an improved structure based on the block representation is presented. Where each block groups boxes with the same orientation along the X and Y axis. The proposed optimisation-based approach has been tested against available literature data sets with supported graphical layout structures. A new Metric known as the Complexity measure is introduced, where it aids in describing the complexity of pallet loading. Up to our knowledge, this is the first model testing the block approach using a linear mathematical model, as all previous block approaches have been tested using heuristic algorithms

2.1 Introduction

Pallet configuration, also known as Cutting and Packing (CP), knapsack problems, strip packing, and bin packing, is a key feature for efficient supply chains with applications in packing, cargo, shipping, and warehouse operations (Dyckhoff, 1990). Due to the fact that cutting problems aim to find an efficient and optimal way to cut large objects into smaller ones, while packing problems position small objects optimally to shape a larger one, most researchers refer to packing, cutting, or packing and cutting problems as pallet loading problems in the majority of the literature. Many of the problems are structured in the same way, with two types of objects: 'Big' and 'Small.'. The container, pallet, or any other medium used as a basis for loading and configuration is referred to as a 'big' object or Bin. 'Small' items are usually boxes or products that are packed into the Bins. Based on if similar or different kinds of boxes or items are packed, the bins could be homogeneous or heterogeneous (Scheithauer and Terno, 1996b):

- Distributor's Pallet Loading Problem (DPLP): Loaded items are heterogeneous (Different)
- Manufacturer's Pallet Loading Problem (MPLP): Loaded items are homogeneous (Identical)

For the Distributor's Pallet Loading Problem (DPLP), usually, such problems are looked into from a 3-Dimensional perspective, where heterogeneous pallet loading is considered. Items can range from weak heterogeneity to high heterogeneity (many items but few item types) and (few items and many item types), respectively. Pallets or containers can also range from being identical, weakly, or strongly heterogeneous depending on the type used. The following typology summarizes the above based on the objective of the problem at hand:

A- If the objective of the problem is *input minimisation*, then using the minimum number of pallets/containers to pack all items is desirable. The combination of the above-given class assortments generates the following six distinct problems:

- Single stock-size cutting stock problem (SSSCSP) for identical pallets/containers and weakly heterogeneous items.
- Single bin-size bin packing problem (SBSBPP) for for identical pallets/containers and strongly heterogeneous items.

- Multiple stock-size cutting stock problem (MSSCSP) for weakly heterogeneous pallets and items.

- Multiple bin-size bin packing problem (MBSBPP) for weakly heterogeneous pallets/containers and strongly heterogeneous items.

- Residual cutting stock problem (RCSP) for strongly heterogeneous pallets/-containers and weakly heterogeneous items.

- Residual bin packing problem (RBPP) for strongly heterogeneous pallets and items.

B- If the objective of the problem is *output maximisation*, then maximizing the number of items packed onto a set number of given pallets/containers is needed. The container could either be single or multiple. In this case, seven different problems could be defined as the following:

- Identical item packing problem (IIPP) for single pallets/containers where the items are identical.

- Single large object placement problem (SLOPP) for single pallets/containers where the items are weakly heterogeneous.

- Single knapsack problem (SKP) for single pallets/containers where the items are strongly heterogeneous.

- Multiple identical large object placement problem (MILOPP) for multiple identical sized pallets/containers where the items are weakly heterogeneous.

- Multiple heterogeneous large object placement problem (MHLOPP) for multiple weakly or strongly heterogeneous sized pallets/containers where the items are weakly heterogeneous.

- Multiple identical knapsack problem (MIKP) for multiple identical sized pallets/containers where the items are strongly heterogeneous.

- Multiple heterogeneous knapsack problem (MHKP) for weakly or strongly heterogeneous sized pallets/containers where the items are strongly heterogeneous.

In this research, we focus on the Manufacturer's Pallet Loading Problem (MPLP) that is considered as a class of the Distributor's Pallet Loading Problem (DPLP); where based on the above topology, the problem is classified as an Identical item packing problem (IIPP), and the objective of the model is output maximisation.

The key constraints common in this research area are usually the following:

- Item edges are parallel to borders of the bins
- Items must be packed within the dimensions of the bins
- Items can not overlap with one another

Additional constraints, including box orientation, cargo weight, stability, weight distribution, and health and safety considerations, such as no combination of pharmaceuticals and detergents in a single pallet, may be considered depending on the problem at hand.

2.2 Literature Review

As mentioned in the Introduction section, two sub-problems fall under the Pallet Loading Problem (PLP); the Manufacturer's Pallet Loading Problem (MPLP); which supports homogeneous pallet loading, and the Distributor's Pallet Loading Problem (DPLP); which is concerned of heterogeneous pallet loading and known by its relatively more complex characteristics and longer computational time (Hodgson, 1982). In this thesis, the main focus is on the MPLP; where the aim is to maximise the number of boxes loaded onto the pallets using a block representation. Where each block groups boxes with the same orientation along the X and Y axis. Below is the literature review related to this area.

Mathematical models are used to configure the pallets effectively, where they would determine the best product placement and try to optimise the number of boxes or reduce the space wasted between the boxes during packing. Researchers have considered two approaches to solving such problems: mathematical optimisation models and heuristic models. When pallet and box sizes are pre-specified and available, mathematical models are applicable. Linear programming, as in (Arghavani and Abdou, 1996; Tsai et al., 1993), are one of the most popular mathematical methods used in literature to solve such problems. Heuristic approaches use the same definition but with different parameters and methods to reach feasible solutions. Computation time increases significantly when exact algorithms are used, while heuristic methods tend to minimise time and find near-optimal solutions.

When solving problems in this area, box placement decisions are made before pallet packing; there are three standard techniques; layer by layer building, column stacking, and random stacking methods. The first method results in a layered pallet configuration where each layer is individually built up to the product height, and then that is repeated until the maximum pallet height is reached. The column stacking technique results in column stacked pallet load, where boxes are packed vertically until the pallet height is reached, repeated for the entire pallet load. Finally, the random stacking technique stacks the boxes one at a time with no predefined box placement decisions allowing the boxes to freely locate as long as they are within the pallet dimensions. The final technique proves to be better in interlocking the boxes providing better stability.

Such problems have been known to be NP-hard; although exact algorithms can deal with complex problem structures, they still struggle to provide optimal solutions within reasonable computational time. An exact algorithm using a 0 1 mathematical model was proposed by Beasley (1985). Dowsland (1987a) used graph theory to solve such problems, later Bhattacharya et al. (1998) proposed the maximal breadth filling sequence depth-first algorithm. A branch-and-cut algorithm was proposed by Alvarez-Valdés et al. (2005a), and Lau et al. (2009) proposed a hybrid approach for the multi-pallet loading operations. Martins and Dell (2007) reported the set of all pallet loading problems instances with an area ratio (pallet area divided by box area) of less than 101 boxes.

Martins and Dell (2008) proposed an exact algorithm using HVZ coding, representing the horizontal box, vertical box, and zero box, respectively. Ji and Jin (2009) proposed a best-first branch and bound algorithm using a staircase structure. Many block heuristics have been developed, where each block is constructed using boxes that share the same orientation, including G4- and G5- structure based heuristics (Lim et al., 2012; Scheithauer and Terno, 1996a). A five block-based heuristics for large-scale pallet loading problems was developed by Ji and Jin (2009). A high-order non-guillotine (HONG) approach was proposed by Martins and Dell (2008), which takes into account more than five-block structures. Although block heuristics proved to solve most data-sets to near-optimal solutions, still more complex examples struggle to reach an optimal or near optimal solution. Lins et al. (2003), and Birgin et al. (2010) both proposed a recursive approach using an L-shaped structure; they assume that their algorithm was optimal as it always found optimal solutions for all test problems. A listing of some of the literature published in the area of 2-Dimensional pallet loading and the solution method used can be seen in table 2.1.

As it can be seen from table 2.1, the pallet configuration and loading problems have been widely studied over the past by various researchers due to the fact that the application in this area is extremely useful. Many researchers have explored different solving methods ranging from exact algorithms, heuristics, and combining both. As observed from the table, exact algorithms combined with heuristic approaches account for the majority of the models because the pallet loading problem is defined as NP-hard, and such heuristic methods help reduce the computational time.

Table 2.1: Literature Published for the MILP (Vargas-Osorio and Zuniga, 2016)

Authors	Method used	
	Exact algorithm	Heuristic method
Dowsland (1987a)	X	
Dowsland (1987b)	X	
Tarnowski et al. (1994)	X	
Morabito and Morales (1998)	X	X
Farago and Morabito (2000)		X
Amaral and Wright (2001)		X
Young-Gun and Kang (2001)		X
Yamassaki and Pureza (2003)		X
Alvarez-Valdés et al. (2005a)		X
Birgin et al. (2005)		X
Alvarez-Valdés et al. (2005b)	X	X
Pureza and Morabito (2006)	X	X
Wu and Ting (2007)	X	X
Ribeiro and Lorena (2007)	X	X
Martins and Dell (2008)	X	X
Kocjan and Holmström (2008)	X	X
Yi et al. (2009)	X	X
Birgin et al. (2010)	X	X

While all approaches seem to be effective in solving such problems, there are still several real-world limitations, such as the difficulties in running massive packing problem instances, the need for certain particular dimension boxes, and the need for a long computational time to obtain reasonable and efficient performance. Often, in some situations, it has been observed that some limitations can be overlooked to reach a higher utilisation percentage, such as pallet weight due to product packing restrictions or transportation weight. The same principle applies to warehouse racks and truck height limits since these restrictions are seen in real-world operations. It is worth noting that the literature cited in this chapter is in the core of the original MPLP, still more research for the problem exists but with extended problem statements and constraints, which have not been considered here.

This chapter proposes a novel Mixed Integer Linear Programming (MILP) model that generates layouts of an improved structure based on the block representation. Where each block groups boxes with the same orientation along the X and Y axis. Where next, we look into the problem statement in detail, followed by the mathematical formulation. Later, we introduce a new metric known as a Complexity measure followed by a Results and Discussion section where data-sets from literature are used to test and validate the proposed algorithm supported with graphical layouts of the pallet loading. Finally, concluding remarks are given, noting the main contribution of this chapter.

2.3 Problem Definition

For pallet loading and configuration, warehouses are usually restricted by many internal constraints, including pallet sizes, rack height, and any client demands for pre-shipment configuration modifications. The aim is to build up the pallet to accommodate the most amount of weakly homogeneous (identical) boxes as soon as the product comes off the production lines. Based on the delivery market, different pallet dimensions are usually used. The following assumptions are made:

- All Bins (pallets) and Items (boxes) used are required to be rectangular
- All Items (boxes) can rotate in a 2D, as seen in figure 2.1
- Items (boxes) cannot overlap with each other
- One layer of pallet layout is generated in 2D and could be replicated to achieve the pallet height desired by manufacturers

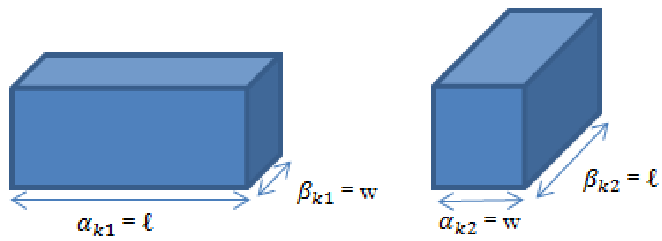


Figure 2.1: 2-Dimensional Box orientations

Overall, the proposed problem can be stated as follows:

Given:

- Dimensions of items (Boxes) in length and width
- Dimensions of Bins (Pallets)
- Upper bound on the number of blocks

Determine:

- The number of boxes loaded in each block with its exact geometrical location; so as to maximise the number of boxes for a given size pallet

2.4 Mathematical Formulation

As seen in figure 2.1 and table 2.2, the pallet configuration problem can be defined as a single sized box with a given dimension that can rotate in 2D defined as α_k and β_k, where k is the number of possible orientations, either $k = 1$ or $k = 2$. Blocks i and j must be used to configure and stack these boxes orthogonally within the pallet, each with the same individual box orientation. These blocks are grouped into a pallet with length and width limits of X^{max} and Y^{max}, respectively. Figure 2.2 shows the block solution, which consists of 5 blocks, each of which is made up of several individual boxes arranged in the same orientation. The detailed mathematical formulation is defined below:

Table 2.2: Possible box orientations

	α	β
$K1$	length	width
$K2$	width	length

2.4.1 Nomenclature

The indices and parameters associated with the proposed model are listed below:

Indices

i, j	Blocks, where j is an alias of i
k	Possible orientations
$r, \bar{r}$	Boxes forming blocks

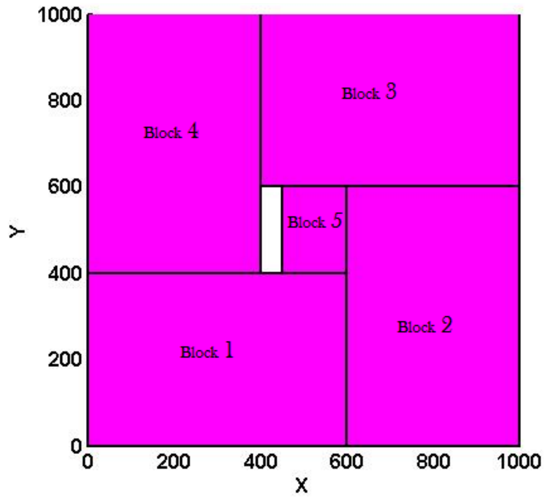


Figure 2.2: 5-Block illustration

Parameters

X^{max}, Y^{max}	Pallet dimension in Length and Width
α_k, β_k	Box dimension
U, M	Appropriate Upper-bound values
UB	Upper-bound on the objective function
ε	Relativity small number

The formulation is based on the following key variables:

Binary Variables

Y_{ik}	1 if orientation k is used for block i; 0 otherwise
ww_i	Penalty binary variable for each block i
ZL_{ir}	r boxes along the X axis for each block i
ZW_{ir}	r boxes along the Y axis for each block i
$E1_{ij}, E2_{ij}$	Non overlapping binary, a set of values that prevent blocks i and j from overlapping in the X and Y direction
$T_{ir\bar{r}}$	Auxiliary variable

Positive Variables

BL_i	Length of each block i
BW_i	Width of each block i
SL_i	Number of boxes across the length of the X axis for each block i
SW_i	Number of boxes across the width of the Y axis for each block i
X_i, Y_i	Coordinates of geometrical center of block i
B_i	Number of individual boxes forming each block i

Integer Variables

NL_i	Number of boxes across the X axis forming block i
NW_i	Number of boxes across the Y axis forming block i

2.4.2 Optimisation Model

The combination of cardinality $r \cdot \bar{r}$ determiners the number of boxes selected for each block i using the Auxiliary binary variable $T_{ir\bar{r}}$; where it is equal to 1 if that combination of block size is chosen and is zero otherwise; as the following in equation 2.1:

$$B_i = \sum_{r\bar{r}} r \cdot \bar{r} \cdot T_{ir\bar{r}} \quad \forall i \tag{2.1}$$

For the blocks to be used in a sequence; i.e., in order, an ordering constraint has been introduced as in the following equation, equation 2.2:

$$B_i \geq B_{i+1} \quad \forall i \tag{2.2}$$

Length Orientation:

The number of individual boxes for each block i across the X axis are found by introducing equations, 2.3 and 2.4. Equation 2.5 guarantees that a length is only determined if the binary variable Y_{ik} is active. Finally, equation 2.6 determines the total length used by each block i, which is the box length chosen α_k multiplied by the number of individual boxes chosen in SL_{ik}.

$$NL_i = \sum_k SL_{ik} \quad \forall i \tag{2.3}$$

$$NL_i = \sum_r r \cdot ZL_{ir} \quad \forall i \tag{2.4}$$

$$SL_{ik} \leq U \cdot Y_{ik} \quad \forall i, k \tag{2.5}$$

$$BL_i = \sum_k \alpha_k \cdot SL_{ik} \quad \forall i \tag{2.6}$$

Width Orientation:

The width is controlled in a similar approach as the following:

$$NW_i = \sum_k SW_{ik} \quad \forall i \tag{2.7}$$

$$NW_i = \sum_r r \cdot ZW_{ir} \quad \forall i \tag{2.8}$$

$$SW_{ik} \leq U \cdot Y_{ik} \quad \forall i, k \tag{2.9}$$

$$BW_i = \sum_k \beta_k \cdot SW_{ik} \quad \forall i \tag{2.10}$$

Lower Bound:

Lower bound design constraints have been considered as in equations 2.11 and 2.12. Where a lower bound on the coordinates of the geometrical center of each block i and j has been considered to avoid intersection of blocks with the origin of axis.

$$X_i \geq \frac{BL_i}{2} \quad \forall i \tag{2.11}$$

$$Y_i \geq \frac{BW_i}{2} \quad \forall i \tag{2.12}$$

Upper Bound:

Upper bound constraints in a similar way, ensure that the blocks are allocated within the pallet dimensions and the rectangular space is defined by the corners (0,0) of the pallet dimensions X^{max}, Y^{max} as follows:

$$X_i + \frac{BL_i}{2} \leq X^{max} \quad \forall i \tag{2.13}$$

$$Y_i + \frac{BW_i}{2} \leq Y^{max} \quad \forall i \tag{2.14}$$

Non-overlapping:

Non-overlapping binary variables $E1_{i,j}$ and $E2_{i,j}$ are introduced here to avoid blocks i and j from occupying the same space/area in the X and Y axis as presented in Papageorgiou and Rotstein (1998). The Big M upper-bound value used here varies depending on the problem dimension/size, where it is equal to the pallet length X^{max}.

Non overlapping in the X direction:

$$X_i - X_j + M(E1_{ij} + E2_{ij}) \geq \frac{BL_i + BL_j}{2} \qquad \forall j \geq i \tag{2.15}$$

$$X_j - X_i + M(1 - E1_{ij} + E2_{ij}) \geq \frac{BL_i + BL_j}{2} \quad \forall j \geq i \tag{2.16}$$

Non overlapping in the Y direction:

$$Y_i - Y_j + M(1 + E1_{ij} - E2_{ij}) \geq \frac{BW_i + BW_j}{2} \quad \forall j \geq i \tag{2.17}$$

$$Y_j - Y_i + M(2 - E1_{ij} - E2_{ij}) \geq \frac{BW_i + BW_j}{2} \quad \forall j \geq i \tag{2.18}$$

2.4.3 Design Constraints

The number of boxes in each direction must be less than or equal to the maximum number of boxes that can be positioned in that direction as in equations 2.19 and 2.20:

$$T_{ir\bar{r}} \leq ZL_{ir} \quad \forall i, r, \bar{r} \tag{2.19}$$

$$T_{ir\bar{r}} \leq ZW_{i\bar{r}} \quad \forall i, r, \bar{r} \tag{2.20}$$

Due to the geometry of the proposed approach, where similar optimal solutions occur, alternate symmetrical layout solutions can be found, requiring longer CPU times. So, in order to increase CPU performance, symmetry-breaking restrictions are imposed, as seen in Westerlund and Papageorgiou (2004); by fixing the first box to the 'bottom left corner' in the following manner:

$$X_1 = \frac{BL_1}{2} \tag{2.21}$$

$$Y_1 = \frac{BW_1}{2} \tag{2.22}$$

$$BL_1 \geq BW_1 \tag{2.23}$$

Finally, in order to minimise computational effort, an upper limit on the objective function has been set to the maximum utilisation percentage for the pallet/box size as follows:

$$\mathrm{UB} = \frac{\mathrm{X}^{\max} \cdot \mathrm{Y}^{\max}}{\alpha_{\mathrm{k1}} \cdot \beta_{\mathrm{k1}}} \tag{2.24}$$

2.4.4 Objective Function

As previously mentioned, this model aims to increase the utilisation percentage of the pallet, which is accomplished by increasing the number of individual boxes r in each block i and j. We also add a penalty ε, a relatively small number set to 0.001 here, to ensure that we optimise the number of boxes included in each block while also minimising the number of blocks used to adhere to the complexity measures discussed later in the chapter.

$$max \qquad \sum_i B_i - \varepsilon \sum_i ww_i \tag{2.25}$$

We need to ensure that the binary variable ww_i is active only when Y_{ik} is active as in equation 2.26 to be apply the penalty in the objective function equation 2.25. And to ensure boxes are only assigned to blocks that have been selected, equations 2.27 and 2.28 have been introduced:

$$\sum_k Y_{ik}{=}ww_i \quad \forall i \tag{2.26}$$

$$\sum_r ZL_{ir}{=}ww_i \quad \forall i \tag{2.27}$$

$$\sum_{\mathrm{r}} ZW_{\mathrm{ir}}{=}ww_i \quad \forall i \tag{2.28}$$

Thus, Model *s_MPLP* consists of Objective function equation 2.25 subject to constraints 2.1-2.24 and 2.26-2.28.

2.5 Complexity Measure

As previously mentioned, the proposed model's key goal is to help provide less complicated graphical layouts of the pallet loading problem, and to that end, a new metric called the Complexity measure is introduced. The metric can be used to compare two pallet layouts that have the same pallet size X^{max} and Y^{max}, as well as the same number of packed boxes B_i, but were created using different mathematical methods, resulting in different box arrangements. The following mathematical formula can be used to define the layout complexity measure:

$$\zeta = \frac{Vrchange + Hrchange}{(2\sum_i B_i) - xr - yc} \tag{2.29}$$

Where ζ represents the average number of orientation changes between the boxes in the pallet and can take any value between 0 and 1, with the following description of complexity:

- 0 indicates a least complex arrangement
- 1 indicates a most complex arrangement

Vertical and Horizontal orientation changes in the X and Y axis of the pallet are captured by the $Vrchange$ and $Hrchange$ variables, respectively. The number of boxes loaded in the pallet is provided by the MILP model proposed above and is denoted by B_i. As seen in figure 2.3, the xr and yr variables reflect the number of boxes in the first row and column along the X and Y axis, respectively.

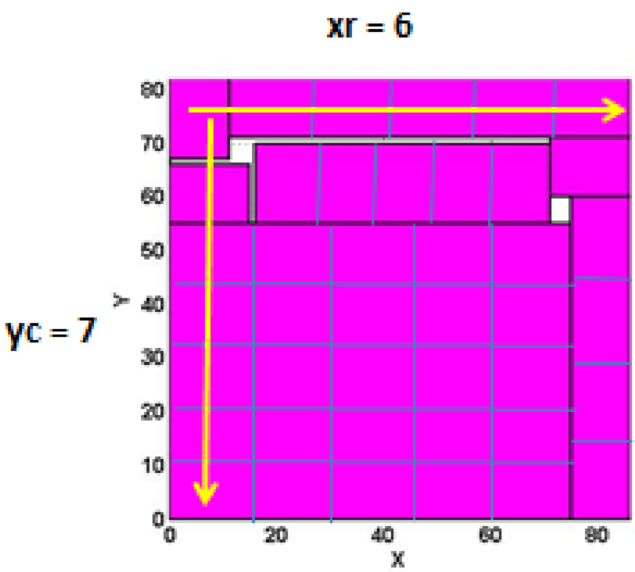


Figure 2.3: Calculation of Xr and Yc for Complexity Index ζ

The flowchart presented in figure 2.4 demonstrates the calculation method in detail. We first calculate the Xr and Yc values, the number of boxes on the first row and column of the pallet. We then calculate the vertical $Vrchange$ and horizontal $Hrchange$ changes simultaneously, assigning a value of zero for the first row and column boxes across the pallet, as these boxes are the pallet's starting points and no rows or columns of boxes occur before them. We then move in the vertical and horizontal direction of the pallet, looking for a change in orientation between any box and its adjust box; if a change is detected, a value of 1 is assigned, and if no change is seen, a value of 0 is given. The process is repeated for every box until all boxes have been assigned a value of (0 or 1). The number of orientation changes is then summed up and applied in the complexity index equation, equation 2.29 to find the value of ζ.
In more detail, and to better understand the definition of the complexity index (0-1) values, consider two different pallet loading scenarios: one with just one block and the other with several small blocks equal to the number of individual boxes in the pallet. Cases (a) and (b) are shown in Figure 2.5, respectively. When only one block is used, the complexity index ζ equals zero, indicating a simple pallet configuration, where all the individual boxes that make up the block are positioned in the same direction, resulting in $Vrchange$ and $Hrchange$ values of *zero*. In the second case, each block has a different orientation from its adjacent box with no blocks combining them. When applying the complexity index equation above, ζ will be equal to 1, indicating a very complex arrangement. Table 2.3 shows the calculations of both layouts generated. The above Complexity index formulation will be used in the next section to compare the proposed approach model layouts with various Literature layouts.

Table 2.3: Complexity Index for One and Multi Block Examples

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
One Block	0	0	9	3	3	**0**	**1**
Multi Block	15	16	20	5	4	**1**	**20**

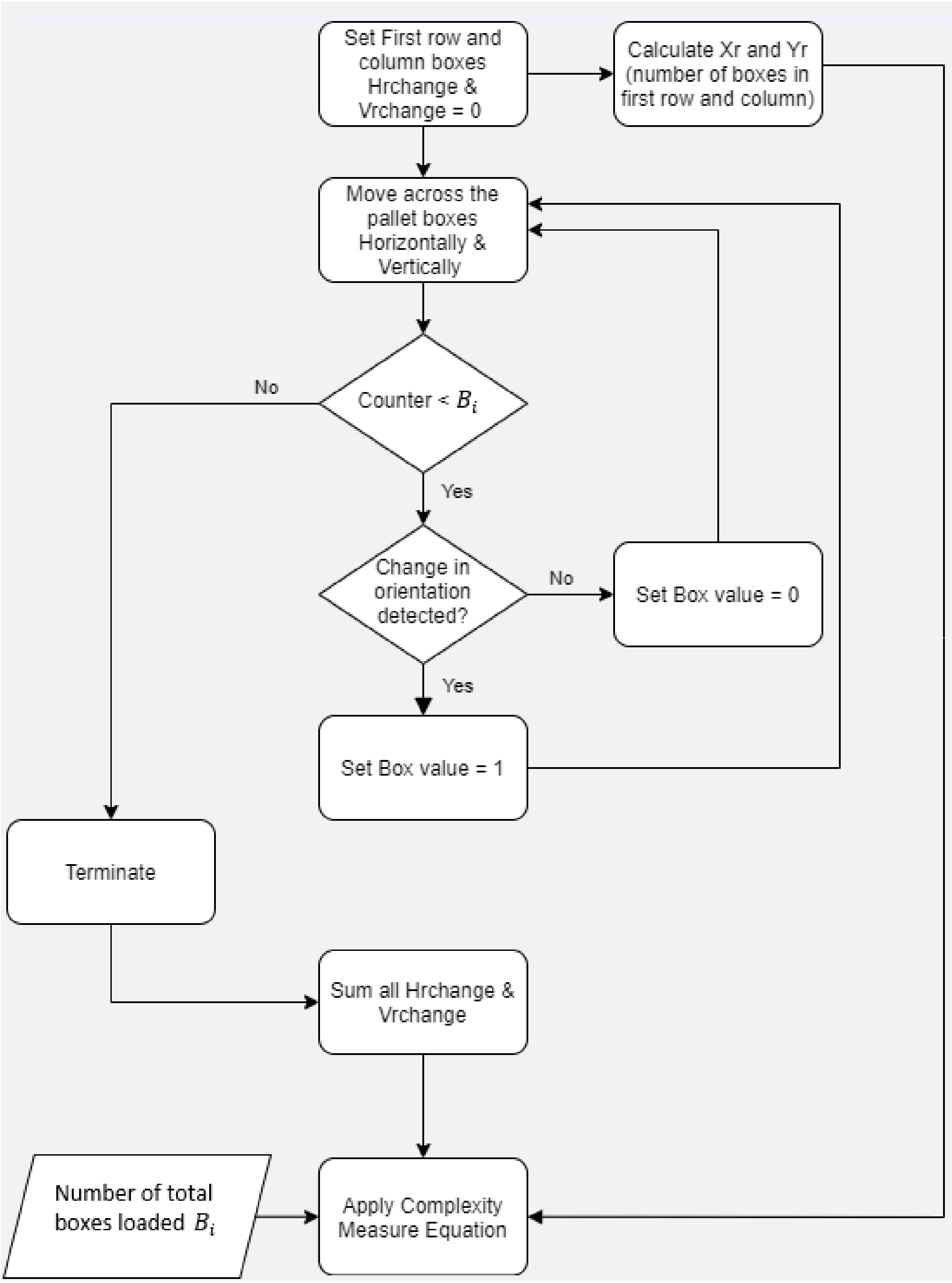
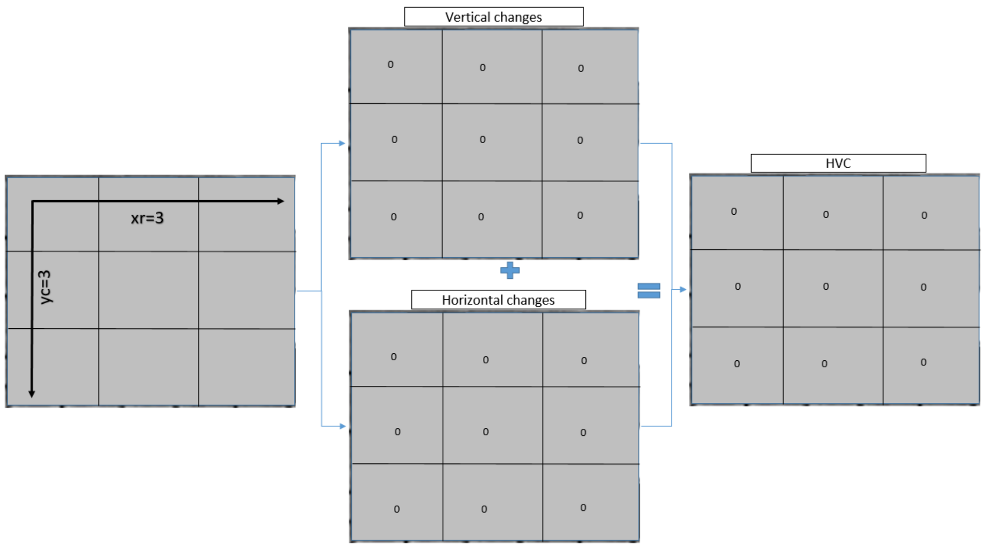
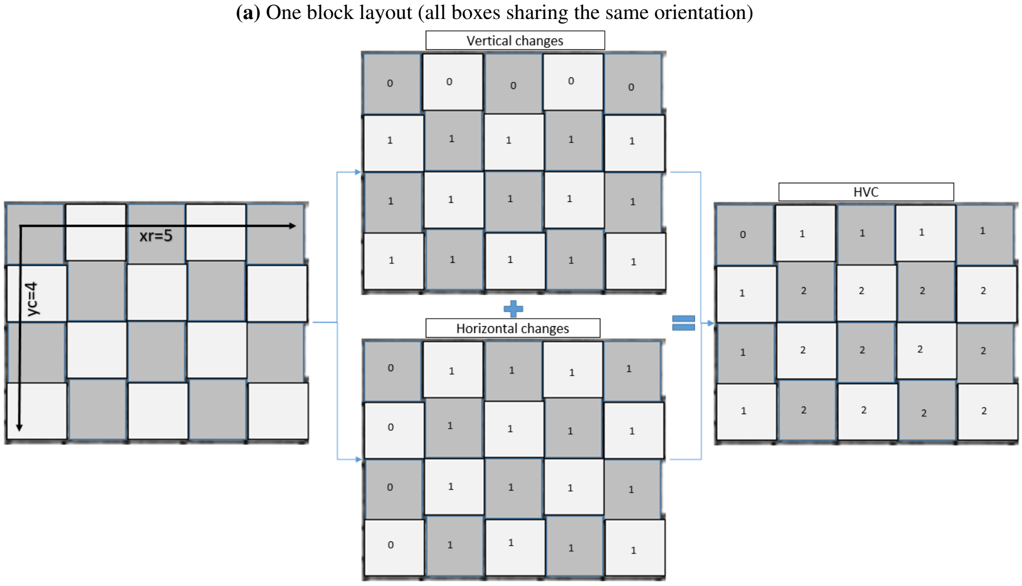


Figure 2.4: Complexity Measure Flowchart



(a) One block layout (all boxes sharing the same orientation)



(b) Multi block Layout (all adjacent boxes have different orientations)

Figure 2.5: Complexity index (0-1) examples

2.6 Results and Discussion

2.6.1 Benchmark Instances

To assess and evaluate the presented model's performance, the results were compared to the Benchmark instances published in Silva et al. (2016) review paper on the pallet loading problems. The instances cover a wide range of instances from literature since 1982 up to recent years, ranging from small, medium to large instances. They have also been tested in at least two pallet loading models from literature being solved in either heuristic, exact methods, or both. Therefore they represent a wide range of instances that can be used in evaluating the performance. The instances are presented in table 2.4, instances (1-25) and (35-45) have been solved using both exact algorithms and heuristic approaches. While for instances (26-34) and (46-55), they have been solved using only heuristic approaches. The pallet dimensions are presented as X^{max} and Y^{max} for the X and Y direction. α and β are the individual box dimensions, z and s_MPLP are the number of boxes loaded into the pallet from literature and the s_MPLP model, respectively. The time required to run each instance using the s_MPLP model is presented in the CPU column and measured in seconds. The last column in the table indicates where the instances have been tested in the literature.

It should be mentioned that the instances were solved using the GAMS modelling system with the CPLEX Mixed Integer Linear Programming optimisation package, on an Intel®Xeon®E5-1620 CPU with 16GB RAM PC and with a relative termination tolerance of 1 %. The computation times for the instances were between 5 seconds for simple instances to 1000 seconds for more challenging ones (average around 400s). It should be noted that the computational effort times reported here vary from one instance to the other based on many factors, such as the pallet to box ratio, which is defined as $L.W/\alpha.\beta$. Also the number of boxes being loaded and the non-overlapping constraints have a direct effect on the CPU times. But given that these problems are classified as NP-hard, the computing times are considered to be within a reasonable computational effort range. Furthermore, the

Table 2.4: Literature Data-Set (a) with *s_MPLP* model results (Silva et al., 2016)

#	L	W	α	β	z	*s_MPLP*	CPU (s)	Papers where instances were used
1	1000	1000	205	159	30	30	16.64	(Bischoff and Dowsland, 1982),
2	1000	1000	200	150	33	33	615.8	(Young-Gun and Kang, 2001),
								(Wu and Ting, 2007)
3	14	10	3	2	23	23	4.3	(Dowsland, 1984), (Young-Gun and Kang, 2001), (Wu and Ting, 2007)
4	22	16	5	3	23	23	10.6	(Dowsland, 1984),
5	86	82	15	11	42	42	863.1	(Arenales and Morabito, 1995),
								(Bhattacharya et al., 1998), (Morabito and Morales, 1998), (Young-Gun and Kang, 2001)
6	30	22	7	4	23	23	32.2	(Dowsland, 1984),
7	46	34	11	6	23	23	63.2	(Arenales and Morabito, 1995),
8	50	36	11	7	23	23	53.87	(Bhattacharya et al., 1998),
9	53	51	9	7	42	42	617.8	(Young-Gun and Kang, 2001),
10	63	60	11	8	42	42	331.2	(Wu and Ting, 2007)
11	76	73	13	10	42	42	1000	
12	87	47	7	6	97	97	1000	(Neliβen, 1995), (Bhattacharya et al., 1998), (Morabito and Morales, 1998), (Amaral and Wright, 2001), (Young-Gun and Kang, 2001), (Pureza and Morabito, 2006), (Wu and Ting, 2007), (Martins and Dell, 2008)
13	57	44	12	5	41	41	1000	(Neliβen, 1995), (Morabito and Morales, 1998), (Young-Gun and Kang, 2001), (Pureza and Morabito, 2006), (Wu and Ting, 2007), (Martins and Dell, 2008)
14	40	33	7	4	46	46	193.2	(Neliβen, 1995), (Bhattacharya et al., 1998), (Young-Gun and Kang, 2001), (Wu and Ting, 2007)
15	3750	3063	646	375	46	46	175.6	(Neliβen, 1995),
16	1200	800	176	135	38	38	114.7	(Bhattacharya et al., 1998),
17	34	23	5	4	38	38	123.8	(Young-Gun and Kang, 2001),
								(Wu and Ting, 2007)
18	300	200	21	19	149	149	535.2	(Scheithauer and Terno, 1996b), (Morabito and Morales, 1998), (Martins and Dell, 2008)

Continued

#	L	W	α	β	z	s_MPLP	CPU (s)	Papers where instances were used
19	40	25	7	3	47	47	623.2	(Scheithauer and Terno, 1996a),
20	52	33	9	4	47	47	531.1	(Morabito and Morales, 1998),
21	56	52	12	5	48	48	422.6	(Pureza and Morabito, 2006),
								(Wu and Ting, 2007),
								(Martins and Dell, 2008)
22	43	26	7	3	53	53	389.5	(Morabito and Morales, 1998),
23	153	100	24	7	90	90	470.1	(Lins et al., 2003),
								(Pureza and Morabito, 2006),
								(Birgin et al., 2010),
								(Wu and Ting, 2007),
								(Martins and Dell, 2008)
24	42	39	9	4	45	45	348.1	(Morabito and Morales, 1998),
25	124	81	21	10	47	47	692.2	(Pureza and Morabito, 2006),
								(Wu and Ting, 2007),
								(Martins and Dell, 2008)
26	100	64	17	10	36	36	247.3	(Amaral and Wright, 2001),
27	100	82	22	8	45	45	498.45	(Ribeiro and Lorena, 2007)
28	100	83	22	8	45	45	503.7	
29	32	22	5	4	34	34	313.6	
30	32	27	5	4	42	42	395.3	
31	40	26	7	4	36	36	335.2	
32	53	26	7	4	48	48	168.4	
33	37	30	8	3	45	45	207.5	
34	81	39	9	7	49	49	195.5	
35	61	38	6	5	77	77	597.7	(Lins et al., 2003),
								(Pureza and Morabito, 2006),
								(Birgin et al., 2010),
								(Wu and Ting, 2007),
								(Martins and Dell, 2008),
								(Ribeiro and Lorena, 2007),
36	63	44	8	5	69	69	263.8	(Lins et al., 2003),
37	61	35	10	3	71	71	426.3	(Pureza and Morabito, 2006),
38	61	38	10	3	77	77	658.3	(Birgin et al., 2010),
39	93	46	13	4	82	82	432.4	(Wu and Ting, 2007),
40	106	59	13	5	96	96	545.9	(Martins and Dell, 2008)
41	141	71	13	8	96	96	164.5	
42	108	65	10	7	100	100	212.4	

Continued

#	L	W	α	β	z	*s_MPLP*	CPU (s)	Papers where instances were used
43	86	52	9	5	99	99	657.1	(Lins et al., 2003),
44	74	46	7	5	97	97	543.4	(Pureza and Morabito, 2006), (Birgin et al., 2010), (Wu and Ting, 2007), (Alvarez-Valdés et al., 2005a)
45	67	44	6	5	97	97	581.7	(Lins L, 2003), (Pureza and Morabito, 2006), (Wu and Ting, 2007)
46	49	28	8	3	57	57	467.8	(Lins et al., 2003),
47	57	34	7	4	69	69	524.8	(Birgin et al., 2010),
48	67	37	11	3	75	75	162.4	(Martins and Dell, 2008)
49	67	40	11	3	81	81	280.2	
50	74	49	11	4	82	82	181.9	
51	2296	1230	136	94	219	219	321.2	(Birgin et al., 2010),
52	2536	1312	144	84	273	273	515.3	(Ribeiro and Lorena, 2007)
53	2252	1470	144	84	271	271	510.7	
54	1470	1458	144	84	175	175	689.4	
55	2296	1230	135	92	226	226	350.4	
		Average time					395.9	

s_MPLP model obtained the same results in terms of the number of boxes packed into pallets as those reported in the literature, but with less complex layouts, as clearly demonstrated in the next section.

As stated before, the main advantage of the proposed model is its ability to freely choose the number of blocks needed to construct the pallet. However, due to computational effort, the number of blocks i,j have been given an upper bound of 15 blocks in all instances tested. This number has been sufficient in solving all problems, where the model in all instances tested has used a maximum of 7 blocks. Another point to mention is the upper bound on the number of individual boxes $r,\bar{r}$ forming each block; it has been set as the max number of boxes that could fit along the X or Y axis if the smallest dimension is used: max(L,W)/min(α,β). The value of this upper bound changes depending on the size of each instance.

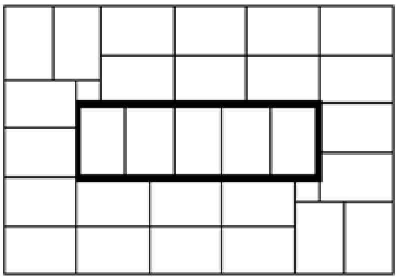
2.6.2 *s_MPLP* Model Layouts

The presented algorithm in this study, as previously stated, has a structure similar to that of heuristic block approaches as the 4-block pattern, G4-structure, and 5 block structure. The resemblance arises from the concept of combining boxes with the same orientation into a single block. Apart from being an optimisation-based model, the proposed algorithm differs from heuristic approaches in that it is an n-block approach in which the boxes are not forced into a predefined number of blocks. Where, depending on the problem size, the boxes are freely grouped into blocks on the pallet dimension. Up to our knowledge, the linear-based approach for block layout in this research has not been provided elsewhere, and the originality of the Complexity measure for layout comparisons is a novel approach. The complexity metric will be examined in the next section, with comparisons between the *s_MPLP* model and Literature approach layouts. A selection of multiple cases from table 2.4 was evaluated and compared; instances 4, 5, 14, 17, 21, 18, and ultimately instance 51, as they span a range of pallet/box sizes.

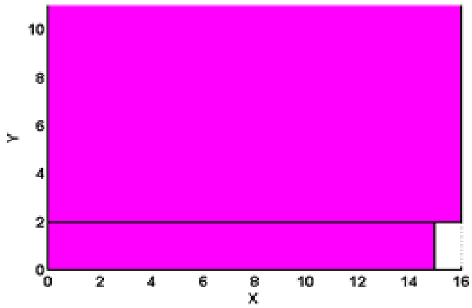
2.6.3 Graphical Layout Comparison

Starting by instance number 4, in table 2.4, a visual comparison is shown in figure 2.6 which shows how the *s_MPLP* layout generated by the *s_MPLP* model is more simplified when considering the number of blocks, where the boxes are grouped into only 2 blocks as opposed to the literature layout used in Young-Gun and Kang (2001) that employs a heuristic 5 block approach, which has used a total of 7 blocks. The simplification associated with the *s_MPLP* model has a distinct benefit for the supply chain cost, as it will decrease the pallet loading time in both human and robotic loading.

Figure 2.7 presents the process for calculating the number of *Vrchange* and *Hrchange* orientation changes across the X and Y axis for each box. Each box will have a value of 0 or 1, with 0 denoting the absence of a previous neighbouring box orientation change and 1 denoting that a box's orientation has changed compared to its neighbouring adjunct box. For each box, we add the number of modifications to determine the total number of changes. Once the Complexity metric described in equation 2.29 is applied to compute the Complexity Index ζ, we may conclude the results, which are listed below in table 2.5.



(a)Literature Layout



(b) s_MPLP Layout

Figure 2.6: Layouts of Instance 4: (a)-Young-Gun and Kang (2001)

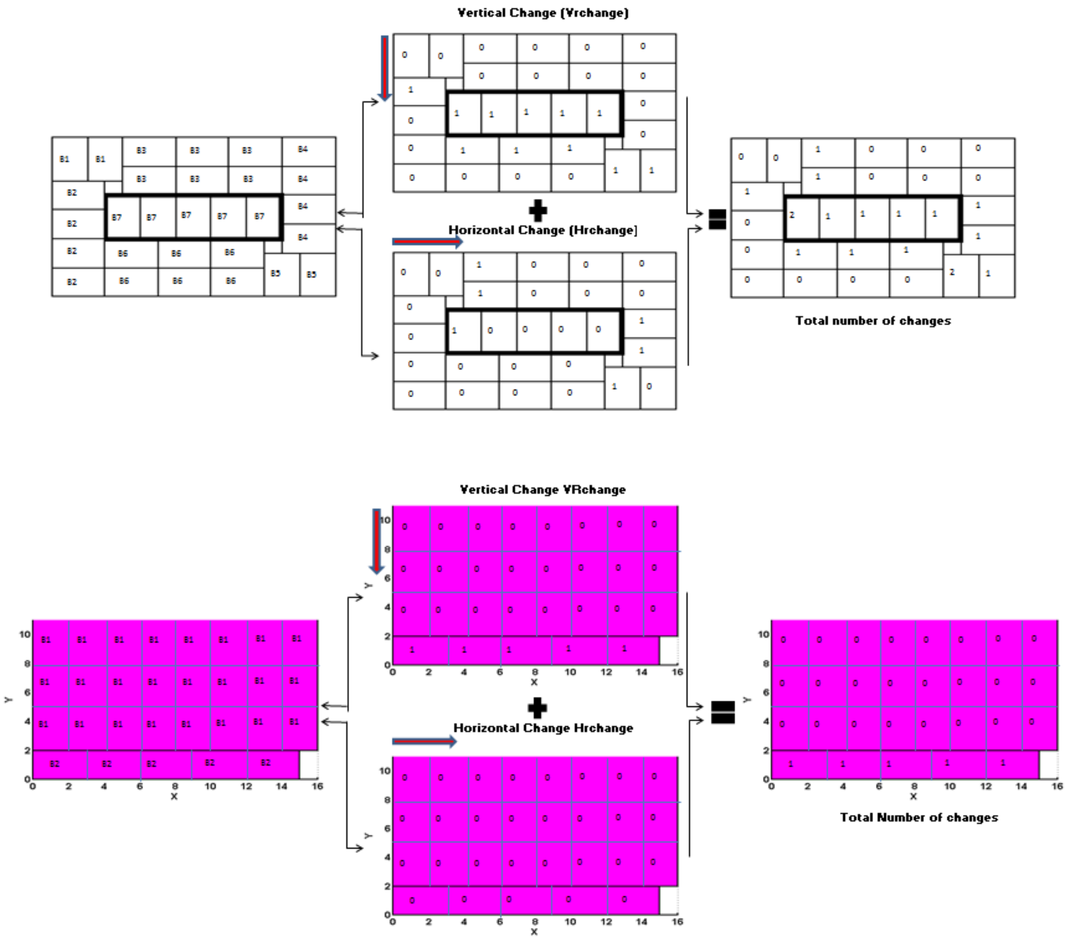


Figure 2.7: Calculation of Complexity Index for Instance 4

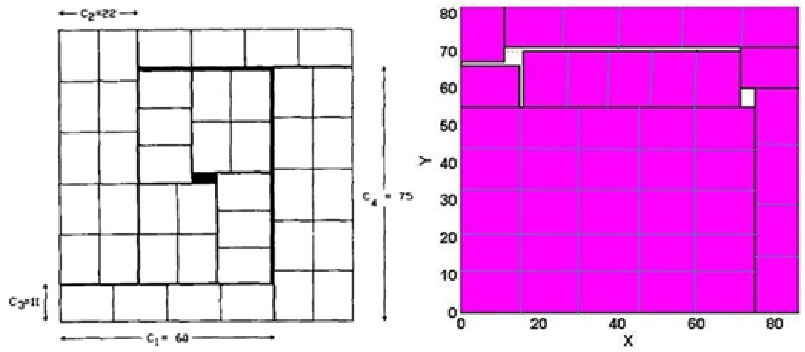
Table 2.5: Complexity Index for Instance 4

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	5	0	29	8	4	**0.10**	**2**
Literature	11	6	29	6	5	**0.36**	**7**

As it can be observed from table 2.5, the complexity index ζ for the *s_MPLP* model is 0.10, which is closer to zero resulting in a less complicated box arrangement compared to literature complexity index ζ of 0.36, which is more complicated. It should be noted that the main drive for the low complexity index value for the *s_MPLP* model approach is the penalty term added in the objective function, that results in a lower number of blocks used. In literature, it is noticed that the $VRchanges$ and $HRchanges$; number of orientation changes are more, as opposed to the $VRchanges$ and $HRchanges$ observed in the *s_MPLP* model. A noteworthy comparison is the total number of blocks used to load the pallet: in the literature layout, 7 blocks were used, whereas, in the *s_MPLP* model, only 2 blocks were used.

In a similar manner, Instance 5 from table 2.4 is analysed, and figure 2.8 illustrates the layout from literature generated using a AND/OR-graph technique as in Arenales and Morabito (1995) and the layout from the *s_MPLP* model, respectively. The Complexity Index comparison is displayed in table 2.6, and the Complexity calculation method is shown in figure 2.9. The *s_MPLP* model has a ζ value of 0.025, whereas the literature has a ζ value of 0.33. For the *s_MPLP* model and literature, the number of blocks required to construct the pallet are 6 and 8, respectively. Although the difference in the complexity index ζ and the number of blocks needed to construct the pallet may not appear to be substantial, when considering facilities that load thousands of pallets every day, considerable time savings may be realised.

For instance 14 from table 2.4, the layouts of both literature (Birgin et al., 2010) and the *s_MPLP* model can be seen in figure 2.10. The Complexity Index for both cases can be seen in table 2.7. The layout presented in the literature is quite hard



(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.8: Layouts of Instance 5: (a)-Arenales and Morabito (1995)

Table 2.6: Instance 5 Complexity Index

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	11	7	42	6	7	**0.25**	**7**
Literature	12	12	42	6	6	**0.33**	**8**

to load due to many orientation changes in the pallet, where 9 blocks have been used. While for the *s_MPLP* model, all the boxes sharing the same orientation are grouped, requiring only 3 blocks to construct the pallet, resulting in a simpler layout. As a result, the *s_MPLP* model has a complexity index ζ of roughly 0.123, indicating a less complex layout compared to the literature complexity index ζ of 0.376, indicating a more complex layout structure.

Table 2.7: Instance 14 Complexity Index

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	9	1	46	5	6	**0.123**	**3**
Literature	25	4	46	10	5	**0.376**	**9**

Moving to instance 17 from table 2.4, the layouts are shown in figure 2.11, where again it could be seen that the *s_MPLP* model outperforms the literature layout (Birgin et al., 2010) by reducing the number of blocks from 13 blocks to only 4 blocks. Leading to a lower complexity index value ζ of 0.187 for the *s_MPLP* model compared to the literature complexity index ζ of 0.468 as seen in table 2.8.

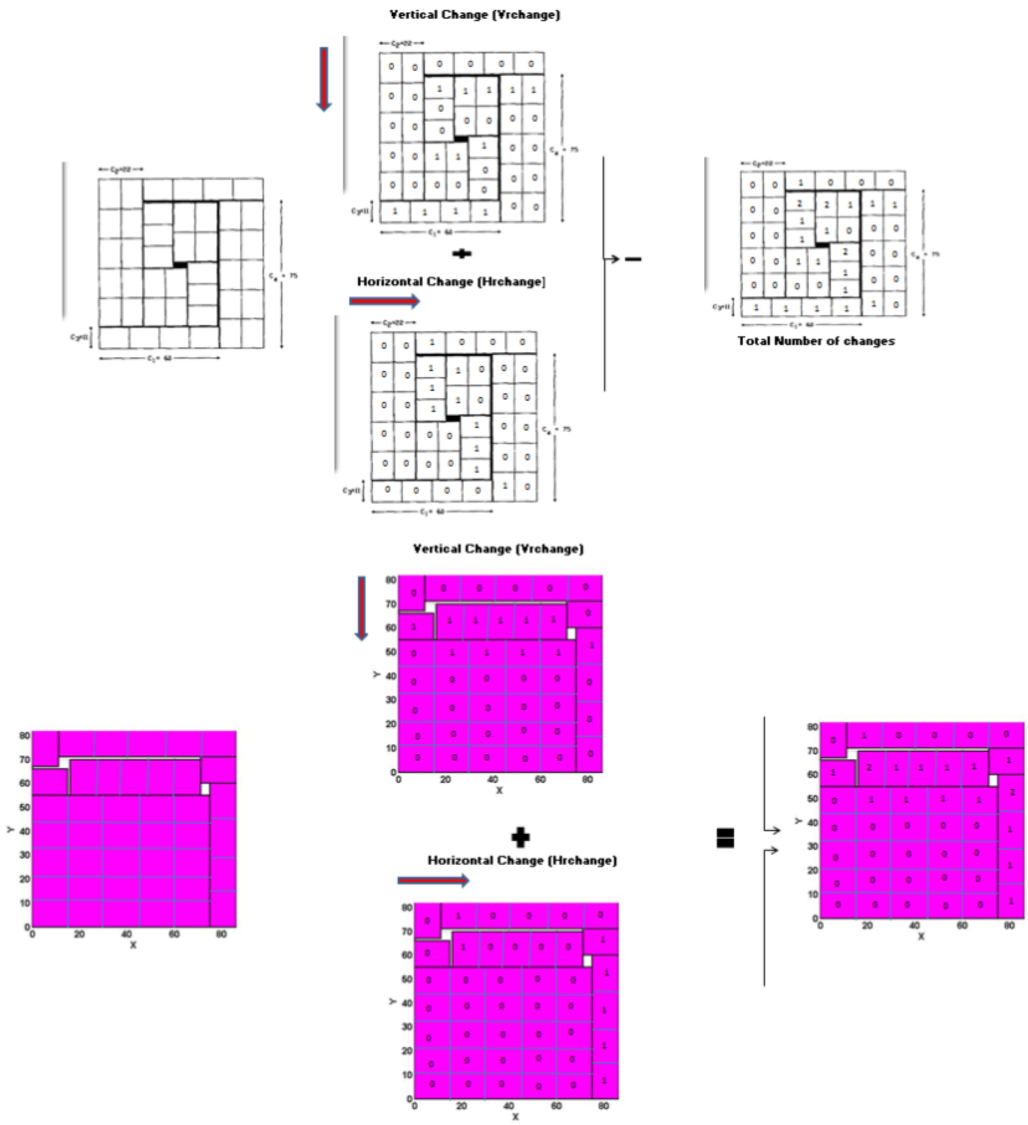
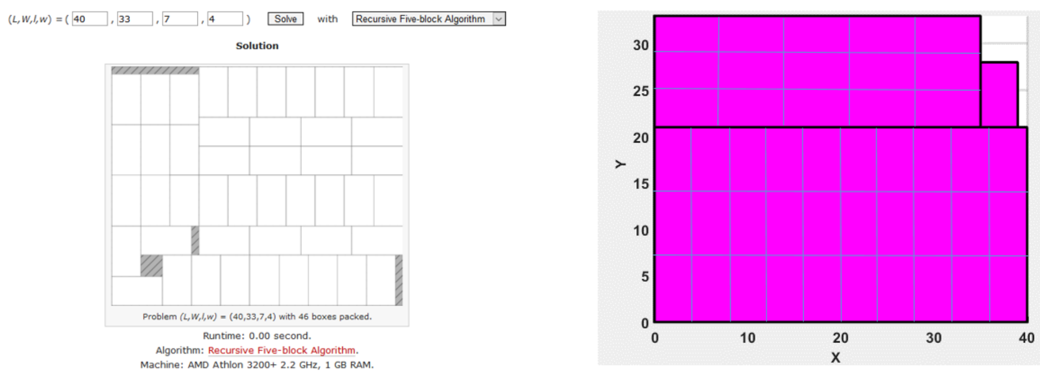
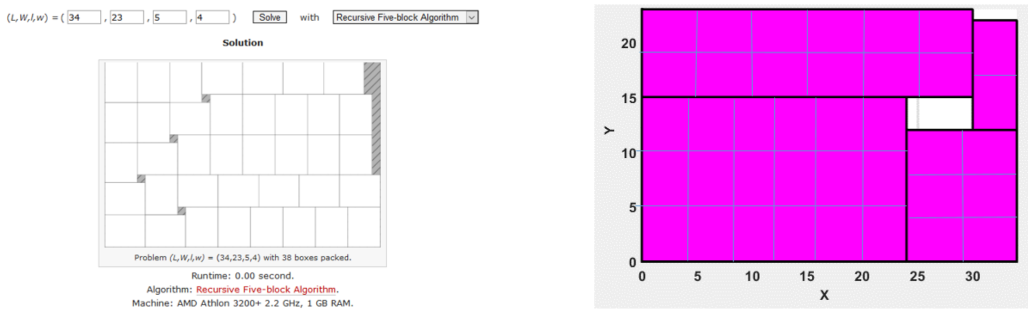


Figure 2.9: Calculation of Complexity Index for Instance 5



(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.10: Layouts of Instance 14: (a)-Birgin et al. (2010)



(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.11: Layouts of Instance 17: (a)-Birgin et al. (2010)

Table 2.8: Instance 17 Complexity Index

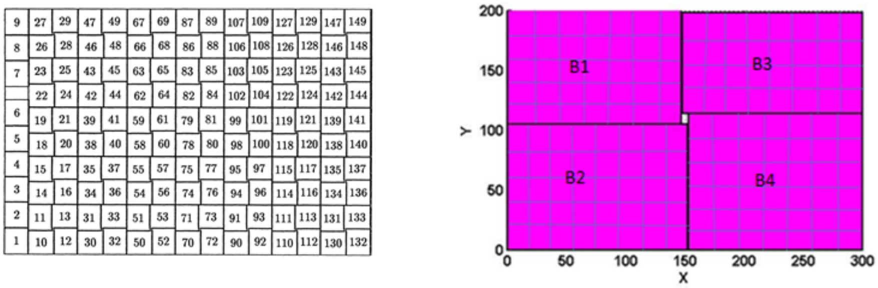
Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	7	5	38	7	5	**0.187**	**4**
Literature	22	8	38	7	5	**0.468**	**13**

When we look at Instance 18 from table 2.4, we can see that the number of boxes to be loaded is considered high; 149 boxes. The layouts are shown in figure 2.12, and the comparison of Complexity Indexes are shown in table 2.9. The literature layout provided in Scheithauer and Terno (1996a) was generated employing a G4-heuristic approach. The layout is quite difficult in terms of implementation in real life since changing the orientation between mostly all neighbouring boxes is a time-intensive procedure. When compared to the *s_MPLP* model developed, the amount of changes in orientation from one block to the other has been minimised, and the boxes that share the same orientation have been grouped into blocks. As a result, the *s_MPLP* model has a complexity index ζ of roughly 0.091, which is very close to zero, indicating a less complex layout as compared to the literature complexity index ζ of 0.97, which is extremely high near 1 indicating a more complicated and challenging layout structure.

In a similar approach to the above, we look into instance 21 from table 2.4, providing the layouts of both literature (Martins and Dell, 2008) and the *s_MPLP* model in

Table 2.9: Instance 18 Complexity Index

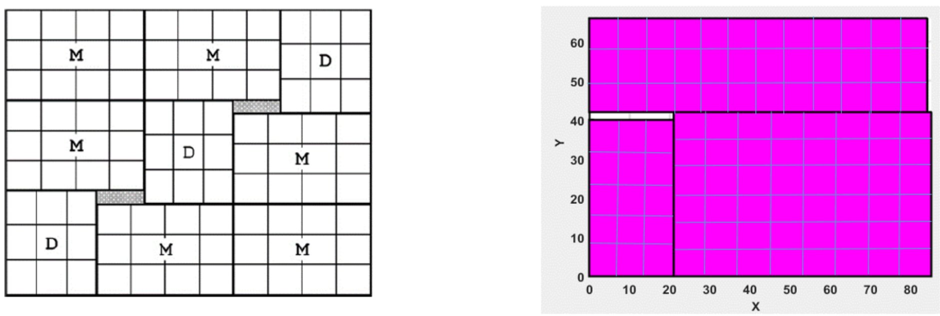
Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	15	10	149	15	10	**0.091**	**4**
Literature	126	140	149	15	9	**0.97**	**141**



(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.12: Layouts of Instance 18: (a)-Scheithauer and Terno (1996a)

figure 2.13. It could be seen from table 2.10 that the number of boxes to be loaded is relatively high; 99 boxes and the *s_MPLP* model was able to load all the boxes in 3 blocks only with a complexity index ζ of 0.078. Whereas for the literature layout presented, 9 blocks have been used with a complexity index value of 0.14.



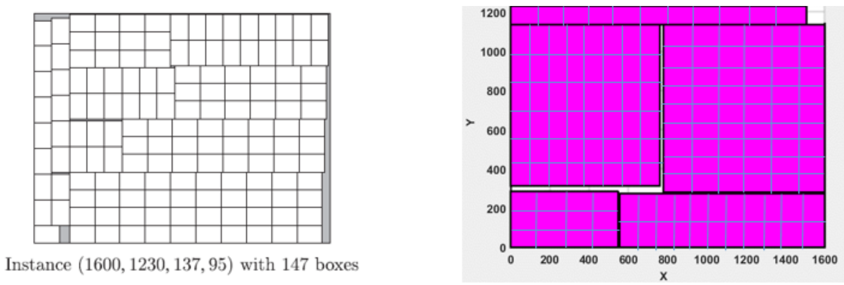
(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.13: Layouts of Instance 21: (a)-Martins and Dell (2008)

Table 2.10: Instance 21 Complexity Index

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	8	6	99	12	8	**0.078**	**3**
Literature	13	12	99	11	9	**0.14**	**9**

Finally, for instance, 51, and from figure 2.14, it could be seen that the *s_MPLP* model produces less complex layouts by using only 5 blocks, with a complexity index of 0.12. While for the literature layout (Birgin et al., 2010), the number of blocks used was more than double the ones used in the *s_MPLP* model, i.e., 11 blocks with a complexity index of 0.15 as shown in table 2.11.



(a)Literature Layout (b) *s_MPLP* Layout

Figure 2.14: Layouts of Instance 51: (a)-Birgin et al. (2010)

Table 2.11: Instance 51 Complexity Index

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
s_MPLP	23	11	147	11	10	**0.12**	**5**
Literature	26	15	147	15	9	**0.15**	**11**

By analysing the above comparisons, it can be concluded that the *s_MPLP* algorithm's alternative layouts are significantly more straightforward in terms of layout complexity yet still achieve the same optimum solution as those reported in the literature. The proposed model groups the maximum number of boxes along the X and Y axis together through blocks minimising the number of orientations in the space. In conclusion, we believe that the less complex and challenging layouts produced by the *s_MPLP* model are crucial in the supply chain operations as they will reduce both; time and cost.

2.7 Conclusions

In this chapter, a single objective MILP model is offered as a solution to the Manufacturer's Pallet Loading Problem (MPLP). The suggested methodology maximises the number of boxes loaded on the pallet by grouping them into blocks based on their orientation. Depending on the magnitude of the instances, the number of blocks necessary varies from one problem to the other. The method has been tested on various data sets and consistently produces the best results in a relatively acceptable amount of time, ranging from 5 for small instances to 1000 seconds for larger ones. Given the NP-hard nature of such problems, the observed computational efforts are deemed acceptable. Additionally, the model specifies the precise geometric placement of the blocks and generates simpler graphical layouts that surpass those in the literature.

The primary contribution of this study is the use of a linear block technique to arrange boxes with the same orientation across the X and Y axis into blocks, hence minimising orientation changes inside the pallet. Additionally, a unique and novel Complexity Measure is introduced, which enables comparing two pallets of the same size but with distinct graphical layouts. Where a new index named ζ has been introduced, it may take any value between 0 and 1, describing the complexity of the pallet layout constructed using any approach. For the instances examined, the Linear s_MPLP model resulted in a lower complexity index value while maximising the number of boxes loaded through the use of blocks. We feel that such a decrease will benefit the whole supply chain operations, and future comparison of time and cost might be used to back up the claims.

In the next chapter, the current single objective model is extended to a multi objective model, where the complexity index measure is integrated into the model, offering a more robust solution methodology.

Chapter 3

Multi-Objective Manufacture's Pallet Loading Problem

This chapter is an extension to the previous one by proposing a multi-objective Mixed Integer Linear Programming (MILP) model based on the epsilon constraint approach to generate improved layouts with low complexity index values. The complexity index formulation is now integrated into the MILP model rather than a post-processing step. The previous chapter's instances are tested again to evaluate the model's performance, including a set of more complex instances. Finally, the computational effort of the multi-objective model is compared with the single-objective model.

3.1 Introduction

In general many of the optimal packing configuration layouts presented by mathematical programming or heuristic are very complex and not applicable to current supply chains when effective, robust, and speedy operations are essential. So a Multi-Objective Mixed Integer Linear Programming (MILP) model to tackle the complexity of such problems ensuring optimal results are always found with less complex graphical layouts, is presented in this chapter. The novel model presented groups the boxes with the same orientation along the X and Y axis as a single block without limiting or forcing a number of "blocks" to be used as in the heuristic block approaches studied in literature. The proposed approach has been tested against several available data sets from literature, supported by graphical layouts for comparison.

For the remainder of this chapter, Section 3.2; defines the problem statement, followed by the mathematical formulation in Section 3.3. Section 3.4; presents the results with comparisons of existing literature. Also a comparison is made between the multi-objective model presented in this chapter and the previous single objective model. Finally, findings and concluding remarks are highlighted in Section 3.5.

3.2 Problem Definition

The problem statement here is the same as defined in Chapter 2; some notations have been changed to differentiate the models from each other. In general, the pallet configuration problem proposed is an arrangement of single boxes that hold the same orientation across the X and Y axis. Where each single box has a length and width defined as α_k and β_k and k is the number of possible orientations; throughout this chapter k has been set to $k = 2$ as we look into the problem from a 2-Dimensional aspect; figure 3.1 demonstrates the orientations. These single boxes r are grouped into blocks known as i and j, and for each block i they share the same orientation and form a matrix size equivalent to $(r \cdot \bar{r})$.

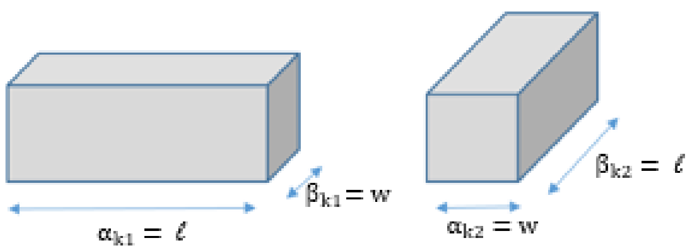


Figure 3.1: 2-Dimensional Box orientations

The main objective is to maximise the number of items packed, assuming that

the loading occurs immediately after the products are ready from the production lines. Again the assumptions made throughout this chapter are the same listed in Chapter 2, in brief; all the Bins (pallets) and Items (boxes) are rectangular, where the items (boxes) can rotate in 2D across the X and Y axis. No overlapping is allowed, and the pallet layout generated is for one layer and would be repeated for additional layers until the maximum height required by the manufacturer is reached.
The overall problem can be stated as follows:

Given:

- Items (boxes) length and width dimensions
- Bins (pallets) length and width dimensions
- Upper bound on the number of blocks allowed

Determine:

- The number of boxes loaded in each block with its exact geometrical location; so as to maximise the number of boxes for a given size pallet while reducing the complexity of loading by grouping boxes that share the same orientation in blocks.

3.3 Mathematical Formulation

3.3.1 Nomenclature

The indices and parameters associated with the *m_MPLP* model are listed below:

Indices

i, j	Blocks
k	Possible orientations
$r, \bar{r}$	Boxes forming blocks
h	Digit of the binary representation

Parameters

L, W	Pallet dimension in Length and Width
α_k, β_k	Box dimension

The following key variables have been used to formulate the problem statement:

Positive Variables

BL_i	Length of each block i
BW_i	Width of each block i
nL_i	Number of boxes across the length of the X axis for each block i
nW_i	Number of boxes across the width of the Y axis for each block i
X_i, Y_i	Coordinates of geometrical center of block i
Xl_i	Bottom left corner X point for block i
YW_i	Bottom left corner Y point for block i
Xr_i	Number of boxes that lay on the X axis for each block i
Yc_i	Number of boxes that lay on the Y axis for each block i
$XrYc_i$	Number of boxes that lay across the X and Y axis on the first row and column of the pallet
HVC	Summation of horizontal and vertical changes in the pallet
ζ	Complexity index
g	Denominator of the Complexity index

Integer Variables

BO_i	Number of boxes forming each block i

Auxiliary Variables

$T_{ir\bar{r}k} \equiv ZL_{irk} * ZW_{irk}$

$Zh_h \equiv g * u_h$

Binary variables

E_{ik}	1 if orientation k for block i is selected; 0 otherwise
p_i	1 if block i exist, 0 otherwise
ZL_{irk}	1 if box r size in orientation k is selected for the length of block i , 0 otherwise
ZW_{irk}	1 if box r size in orientation k is selected for the width of block i , 0 otherwise
$E1_{ij}, E2_{ij}$	Non overlapping binary
zX_i, zY_i	1 if block i lays on the Y or X axis respectively; 0 otherwise
u_h	1 if the nth digit of the binary representation of variable HVC is equal to 1; 0 otherwise

The following figure; 3.2 represents the main notations used across this chapter:

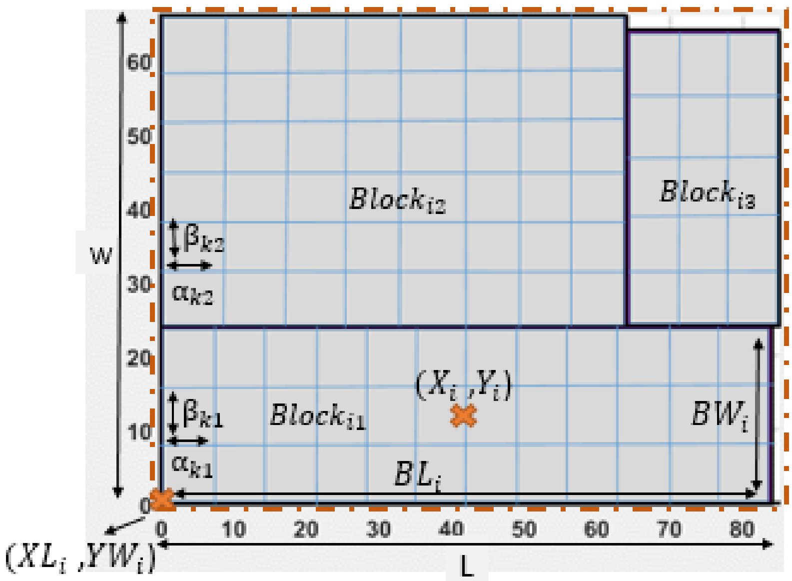


Figure 3.2: Block Notations

3.3.2 Optimisation Model

To formulate the problem statement above we start by calculating the number of boxes selected for each block i as introduced in equations 3.1 and 3.2, where nL is the number of boxes across the X axis of each block i and nW is the number of boxes across the Y axis of each block i as the following:

$$nL_i = \sum_{rk} r \cdot ZL_{irk} \quad \forall i \tag{3.1}$$

$$nW_i = \sum_{rk} r \cdot ZW_{irk} \quad \forall i \tag{3.2}$$

The total length BL_i and width BW_i used across the X and Y axis for each block i is presented below in equations 3.3 and 3.4. Where equation 3.3 calculates the total length of each block i by multiplying the length α_k with the number of boxes selected $r \cdot ZL_{irk}$ across the X axis. And equation 3.4 calculates the total width of each block i by multiplying the width β_k with the number of boxes selected $r \cdot ZW_{irk}$ across the Y axis.

$$BL_i = \sum_{k} \alpha_k \sum_{r} r \cdot ZL_{irk} \quad \forall i \tag{3.3}$$

$$BW_i = \sum_{k} \beta_k \sum_{r} r \cdot ZW_{irk} \quad \forall i \tag{3.4}$$

Lower bound constraints are introduced here to avoid intersection between the geometrical centroids of blocks i and j as in equations 3.5 and 3.6:

$$X_i \geq \frac{BL_i}{2} \quad \forall i \tag{3.5}$$

$$Y_i \geq \frac{BW_i}{2} \quad \forall i \tag{3.6}$$

Upper bound constraints in a similar way force blocks i and j to be placed within the pallet dimensions L and W as in equation 3.7 and 3.8:

$$X_i + \frac{BL_i}{2} \leq L \quad \forall i \tag{3.7}$$

$$Y_i + \frac{BW_i}{2} \leq W \quad \forall i \tag{3.8}$$

To prevent blocks i and j overlapping or occupying the same geometrical location in the X and Y axis the binary variables $E1_{i,j}$ and $E2_{i,j}$ are introduced Papageorgiou and Rotstein (1998), where the Big-M upper bound selected equals to the pallet dimension in the X axis. The following equations capture the non-overlapping constraints:

Non overlapping in the X direction:

$$X_i - X_j + M(E1_{ij} + E2_{ij}) \geq \frac{BL_i + BL_j}{2} \qquad \forall j \geq i \tag{3.9}$$

$$X_j - X_i + M(1 - E1_{ij} + E2_{ij}) \geq \frac{BL_i + BL_j}{2} \quad \forall j \geq i \tag{3.10}$$

Non overlapping in the Y direction:

$$Y_i - Y_j + M(1 + E1_{ij} - E2_{ij}) \geq \frac{BW_i + BW_j}{2} \quad \forall j \geq i \tag{3.11}$$

$$Y_j - Y_i + M(2 - E1_{ij} - E2_{ij}) \geq \frac{BW_i + BW_j}{2} \quad \forall j \geq i \tag{3.12}$$

To determine the total number of boxes BO_i in the pallet; the auxiliary binary variable $T_{ir\bar{r}k}$ is introduced, where it determines the size of the block selected through the cardinality $r \cdot \bar{r}$, and it is equal to 1 if the matrix size forming the block is selected and zero otherwise as in equation 3.13. And to ensure that the auxiliary variable $T_{ir\bar{r}k}$ is only active when a matrix size is selected across the X and Y axis, equations 3.14 and 3.15 are introduced.

$$BO_i = \sum_{r\bar{r}k} r \cdot \bar{r} \cdot T_{ir\bar{r}k} \quad \forall i \tag{3.13}$$

$$T_{ir\bar{r}k} \leq ZL_{irk} \quad \forall i, r, \bar{r}, k \tag{3.14}$$

$$T_{ir\bar{r}k} \leq ZW_{i\bar{r}k} \quad \forall i, r, \bar{r}, k \tag{3.15}$$

To ensure boxes are assigned to blocks only if that block has been selected, equations 3.16 and 3.17 are introduced using binary E_{ik}, and to ensure that an orientation is assigned only if that block has been selected equation 3.18 is included.

$$\sum_{r} ZL_{irk} \leq E_{ik} \quad \forall i, k \tag{3.16}$$

$$\sum_{r} ZW_{irk} \leq E_{ik} \quad \forall i, k \tag{3.17}$$

$$\sum_{k} E_{ik} \leq p_i \quad \forall i \tag{3.18}$$

3.3.3 Complexity Measure

The complexity measure equation as described in Chapter 2; aids in comparing 2 pallet layouts that share the same pallet size L and W as well as the same number of boxes loaded BO_i but generated using different mathematical or heuristic approaches, which as a result; provide different box arrangements. From equation 3.19, The HVC captures the horizontal and vertical orientation changes in the X and Y axis of the pallet, and BO_i is the number of boxes loaded in the pallet. The difference here arises in the way the Xr and Yc are calculated, where they represent the number of boxes in the first row and column along the X and Y axis starting from the bottom left corner of the pallet (0,0) rather than the top first row and column of the pallet as presented in the previous chapter. The new concept is provided in figure 3.3. ζ again captures the average number of changes between the boxes in the pallet, and it takes any value between 0 and 1; 0 indicating a less complex layout, whereas 1 indicates a complex layout.

$$\zeta = \frac{HVC}{(2\sum_i BO_i) - Xr - Yc} \tag{3.19}$$

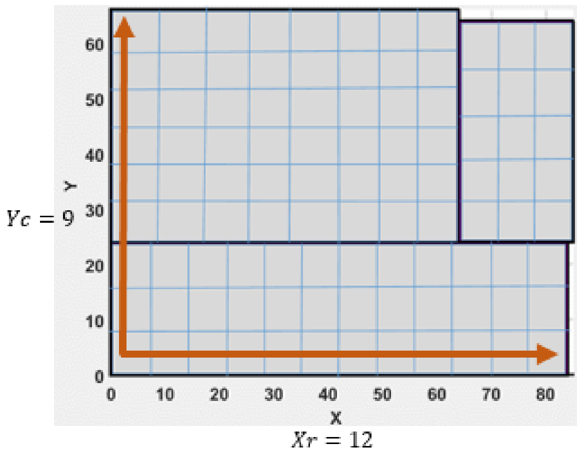


Figure 3.3: Calculation of Xr and Yc for Complexity Index ζ for model m_MPLP

The above equation for the complexity measure, equation 3.19 is in a non-linear form, and since the model proposed is a MILP model, the following presents the linearisation steps used to determine the variable values. In order to obtain the Xr and Yc values, equations 3.20 - 3.36 are introduced. Where first the bottom left corner XL and YW of each block i is determined using equations 3.20 and 3.21 for the X and Y axis, respectively.

$$XL_i = X_i - \frac{BL_i}{2} \quad \forall i \tag{3.20}$$

$$YW_i = Y_i - \frac{BW_i}{2} \quad \forall i \tag{3.21}$$

Two binary variables zX_i and zY_i are introduced to give a value of 1 if the block i has a bottom left corner on the Y or X axis respectively and 0 otherwise, an example is given in figure 3.4 and the mathematical formulation is captured in equations 3.22-3.25.

$$XL_i \geq 1 - zX_i \quad \forall i \tag{3.22}$$

$$XL_i \leq M \cdot (1 - zX_i) \quad \forall i \tag{3.23}$$

$$YW_i \geq 1 - zY_i \quad \forall i \tag{3.24}$$

$$YW_i \leq M \cdot (1 - zY_i) \quad \forall i \tag{3.25}$$

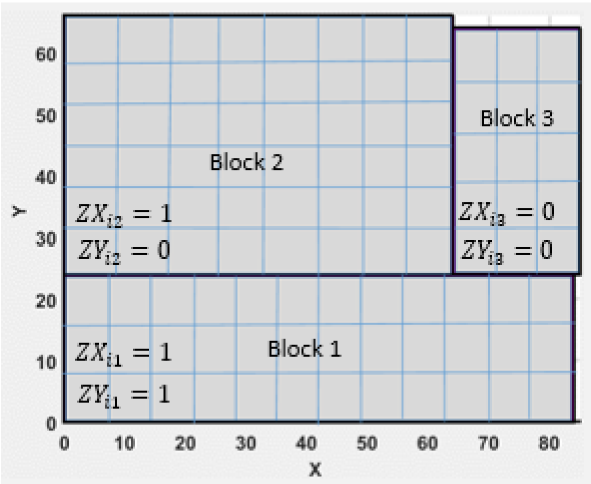


Figure 3.4: Calculation of binaries ZX and ZY

Equations, 3.26 and 3.27 multiply the binary variables zX_i and zY_i with nL_i and nW_i; the number of boxes in each block across the length and width of the pallet, to determine the number of boxes in each block and to calculate the Xr and Yc values.

$$Xr_i = zY_i \cdot nL_i \quad \forall i \tag{3.26}$$

$$Yc_i = zX_i \cdot nW_i \quad \forall i \tag{3.27}$$

Since the above two equations 3.26 and 3.27 are in a non-linear form and include a bi-linear term, we linearise them using the big-M technique. For Xr the following equations 3.28-3.30 have been introduced, where the $U1$ is equal to the (floor) of the pallet length over the box length L/α_k, to ensure we have an integer upper-bound.

$$Xr_i \leq U1 \cdot zY_i \quad \forall i \tag{3.28}$$

$$Xr_i \leq nL_i \quad \forall i \tag{3.29}$$

$$Xr_i \geq nL_i - U1 \cdot (1 - zY_i) \quad \forall i \tag{3.30}$$

In a similar way we linearise equation 3.27 using the big-M technique for Yc and the $U2$ is equal to the (floor) of the pallet width over the box width W/β_k:

$$Yc_i \leq U2 \cdot zX_i \quad \forall i \tag{3.31}$$

$$Yc_i \leq nW_i \quad \forall i \tag{3.32}$$

$$Yc_i \geq nW_i - U2 \cdot (1 - zX_i) \quad \forall i \tag{3.33}$$

Now that we have obtained the values for Xr and Yc the summation of them will give us the $XrYc$ value to substitute later in the complexity index; equation 3.19:

$$XrYc = \sum_i Xr_i + Yc_i \tag{3.34}$$

To capture the HVC, the horizontal and vertical orientation changes in the X and Y axis of the pallet; equation 3.35 is introduced, where the number of boxes across the first row and column of the pallet for each block i is subtracted from the $XrYc$

value calculated in the above equation (3.34):

$$HVC = \left[\sum_i nL_i + nW_i\right] - XrYc \tag{3.35}$$

The denominator of equation 3.19 is represented here as g, and it captures twice the number of boxes used in the pallet minus $XrYc$; the first row and column of boxes across the X and Y axis. We multiply the number of boxes in the pallet by two as it is known that each box can obtain a maximum number of two changes if a change is seen in the vertical and horizontal direction except for the first row and column of pallet($XrYc$), as these boxes have an upper limit of one change since they are the starting boxes and no other boxes appear before. Equation 3.36 captures the above:

$$g = (2 \cdot \sum_i BO_i) - XrYc \tag{3.36}$$

From the above, the complexity index equation 3.19 is reformulated using the new variable g as seen in equation 3.37:

$$\zeta = \frac{HVC}{g} \tag{3.37}$$

As the above complexity index is in a non-linear form, and it involves an integer variable g that could be expressed by its binary representation, equation 3.38 is introduced. Where u_h indicates whether the n^{th} digit of the binary representation of variable g is equal to 1, and $H = log_2(maxg)$.

$$g = \sum_{h=1}^{H} 2^{h-1} \cdot u_h \tag{3.38}$$

Equation 3.37 can now be linearised by introducing an auxiliary variable $Zh_h \equiv g \cdot u_h$ and the following constraints as in equations 3.39- 3.42, where the upper bound $U3$ here is the maximum number g can hold, which as explained above is equal to maximum number of changes a box can hold in the pallet.

$$HVC = \sum_{h=1}^{H} 2^{h-1} * Zh_h \tag{3.39}$$

$$Zh_h \leq U3 * u_h \qquad \forall h \tag{3.40}$$

$$Zh_h \leq \zeta \qquad \forall h \tag{3.41}$$

$$Zh_h \geq \zeta - U3(1-u_h) \qquad \forall h \tag{3.42}$$

noindent To demonstrate the proposed method for the HVC; the horizontal and vertical changes, figure 3.5 is introduced, where; the calculations for the horizontal and vertical changes are made simultaneously; at the first stage, it is assumed that all boxes on the first row and column starting from the (0,0) point hold a value of zero, due to the fact that they are the starting points in the pallet and no rows or columns of boxes exist before them. As we move throughout the pallet horizontally and vertically, we check whether a change in the block size has been detected; if so, a value of 1 is assigned to all the boxes on the first row/column of the block, and if no change is seen the box is given a value of zero.

3.3.4 Objective Function

From the above, a multi-objective optimisation model is formulated, where the objective is to maximise the number of boxes loaded in the pallet BO_i, while minimising the complexity index ζ to obtain less complex graphical layouts. To solve the model, the ε-constraint method (Chankong and Haimes, 2008) is adapted , where only one of the objective terms is optimised, and the other is converted as a constraint using an appropriate upper-bound. The number of boxes BO_i is kept in the objective function, and the complexity index ζ is converted as a constraint bounded

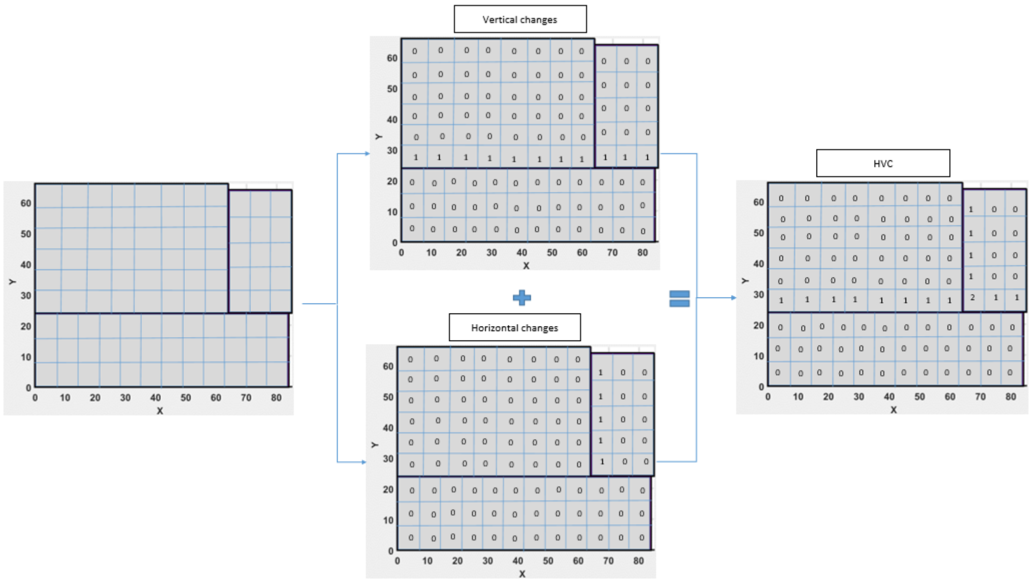


Figure 3.5: Vertical and Horizontal changes calculation method

by an appropriate upper value. Thus a single-objective model is obtained as follows:

$$max \qquad \sum_i BO_i \tag{3.43}$$

$$s.t. \qquad \zeta \leq \varepsilon \tag{3.44}$$

Along with the block constraints (3.1-3.18), the complexity measure constraints (3.20-3.25), (3.28-3.36) and (3.38-3.42). This constitutes model m_MPLP.

Knowing that ζ is between 0 and 1 the model is solved over several iterations, where at the first run ε is set to the upper-bound 1 and is updated after each run with the new ζ value obtained until ζ reaches a value of 0. The above model will ensure that the layouts generated are as simple as possible while reaching the optimal solution for the number of boxes loaded in the pallet. Moreover, to test the validity of the proposed model, the following section will present the Pareto curves and compare the layouts generated with different literature layouts.

3.4 Computational Results

3.4.1 Benchmark Instances

The model was implemented using GAMS modeling system and with CPLEX Mixed Integer Linear Programming optimisation package, on an Intel®Xeon®E5-1620 CPU with 16GB RAM PC and with a relative termination tolerance of 1 %. The proposed model has been tested against 2 data sets from literature; the first is the same data used in Chapter 2, and it is displayed in table 3.1. The second data set is displayed in table 3.2, and represents actual data from the Brazilian ports (Ribeiro and Lorena, 2008). In both tables, z is the total number of boxes loaded from literature, *m_MPLP* is the solution obtained from the MILP model proposed in this chapter. The pallet dimensions are presented as L and W, the box dimensions are α and β, and finally, the CPU time required to solve each instance using the proposed approach is reported as CPU and measured in seconds. For table 3.2 a comparison between literature and the *m_MPLP* model run times has been presented, where an additional column named *LiteratureCPU* has been added. For table 3.1 the computational times with literature could not be performed due to the lack of data.

It should be noted that the number of blocks i, j for all tested instances as in model *s_MPLP*, has been set to 15 blocks, where the proposed algorithm can freely choose the number of blocks required. Also, the max value for $r, \bar{r}$ has been set to the max number of boxes that could fit along the X or Y axis if the smallest dimension is used as the following: max(L,W)/min(α,β), and this number is different depending on the problem size.

The proposed approach for the data set (a) in table 3.1 has provided a better complexity index value in addition to the reduction of the number of blocks used while loading the same number of boxes as reported in the literature with a reasonable computational time average of less than 140 seconds. For data set (b) 3.2, a considerable reduction in times could be seen, where literature times have averaged 8910.6 seconds, the *m_MPLP* model has averaged about 300 seconds while having the same number of blocks loaded per pallet. The detailed results are presented in the following section.

Table 3.1: Literature Data-Set (a) with *m_MPLP* model results (Silva et al., 2016)

Instance	L	W	α	β	z	*m_MPLP*	CPU (s)
1	1000	1000	205	159	30	30	3.7
2	1000	1000	200	150	33	33	433.5
3	14	10	3	2	23	23	0.1
4	16	11	3	2	29	29	0.2
5	86	82	15	11	42	42	22.4
6	30	22	7	4	23	23	28.1
7	46	34	11	6	23	23	29.8
8	50	36	11	7	23	23	35.5
9	53	51	9	7	42	42	301
10	63	60	11	8	42	42	222.9
11	76	73	13	10	42	42	247.3
12	87	47	7	6	97	97	231.6
13	57	44	12	5	41	41	265.9
14	40	33	7	4	46	46	4.2
15	3750	3063	646	375	46	46	13.7
16	1200	800	176	135	38	38	9.1
17	34	23	5	4	38	38	10.8
18	300	200	21	19	149	149	432.7
19	40	25	7	3	47	47	283.5
20	52	33	9	4	47	47	203.2
21	85	66	8	7	99	99	231.8
22	43	26	7	3	53	53	155.9
23	153	100	24	7	90	90	231.5
24	42	39	9	4	45	45	148.4
25	124	81	21	10	47	47	158.3
26	100	64	17	10	36	36	23.7
27	100	82	22	8	45	45	3.7
28	100	83	22	8	45	45	15.9
29	32	22	5	4	34	34	5.1
30	32	27	5	4	42	42	4.2
31	40	26	7	4	36	36	8.3
32	53	26	7	4	48	48	45.9
33	37	30	8	3	45	45	68.7
34	81	39	9	7	49	49	18.2
35	61	38	6	5	77	77	173.4
36	63	44	8	5	69	69	204.3
37	61	35	10	3	71	71	387.3
38	61	38	10	3	77	77	238..4
39	93	46	13	4	82	82	218.9
40	106	59	13	5	96	96	136.8
41	141	71	13	8	96	96	153.8
42	108	65	10	7	100	100	207.5
43	86	52	9	5	99	99	326.3
44	74	46	7	5	97	97	217.8

Continued

Instance	L	W	α	β	z	m_MPLP	CPU (s)
45	67	44	6	5	97	97	273.6
46	49	28	8	3	57	57	152.7
47	57	34	7	4	69	69	112.6
48	67	37	11	3	75	75	75.8
49	67	40	11	3	81	81	85.8
50	74	49	11	4	82	82	69.3
51	1600	1230	137	95	147	147	123.8
			Average time				138.4

Table 3.2: Literature Data-Set (b) with m_MPLP model results (Ribeiro and Lorena, 2008)

Instance	L	W	α	β	z	m_MPLP	CPU (s)	Literature CPU (s)
L1	2296	1230	136	94	219	219	51.5	4768
L2	2536	1312	144	84	273	273	205.4	24637
L3	2252	1470	144	84	271	271	285.3	9634
L4	1470	1458	144	84	175	175	156.3	2889
L5	2296	1230	135	92	226	226	122.5	6268
L6	1804	1230	137	95	168	168	232.8	907
L7	2466	1230	137	95	231	231	832.4	7460
L8	1804	1750	137	95	240	240	232.5	16249
L9	2426	1230	137	95	227	227	612.6	7384
			Average time				303.4	8910.6

3.4.2 m_MPLP Model Layouts and Complexity Comparison

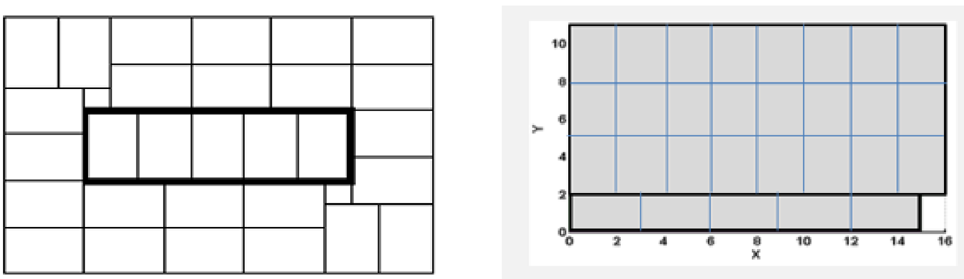
The proposed model in this chapter and as previously mentioned, is a similar structure compared to heuristic block methods, such as the 4-block pattern, the G4-structure, and the 5-block structure. The similarity arises in the idea of grouping boxes that share the same orientation in one block. It differs from heuristic approaches, apart from being an optimization-based model. It has the functionality of being a multi-block approach where the boxes are not forced in a certain predefined maximum number of blocks. The boxes are freely grouped into blocks on the pallet, depending on the problem size.

In the following section, we will be looking into the comparisons between the proposed approach layouts and Literature layouts. A selection of instances from table 3.1 have been tested and compared; instance number 4 followed by instance

5, 14, 17, 18, 21 and 51 as they cover different sizes of complexity ranging from small, medium to large.

3.4.3 Graphical Layout Comparison

Starting by instance number 4 from table 3.1, the layout from both literature (Young-Gun and Kang, 2001) and the *m_MPLP* model is presented in figure 3.6. The Pareto curve for several iterations for different values of ε is presented in figure 3.7. As seen when the ε value is reduced, the complexity index ζ is reduced, and the number of boxes loaded in the pallet is also reduced until only one block is selected having a ζ value of zero, where all boxes in that block share the same orientation. So a trade-off between the complexity of loading and the number of boxes to load is something the user can select depending on the products being packed and if certain packing restrictions apply, such as single orientation packing. For this instance, 29 boxes are loaded with a complexity index of 0.10 and 25 boxes could be loaded with a complexity index of 0.



(a)Literature Layout (b) *m_MPLP* Layout

Figure 3.6: Layouts of Instance 4: (a)-(Young-Gun and Kang, 2001)

Table 3.3: Instance 4 Complexity Index

Approach	VC	HC	$\sum B_i$	Xr	Yc	ζ	Number of Blocks
m_MPLP	5	0	29	8	4	**0.10**	**2**
Literature	11	6	29	6	5	**0.36**	**7**

From figure 3.6 the number of blocks used in the proposed approach are only

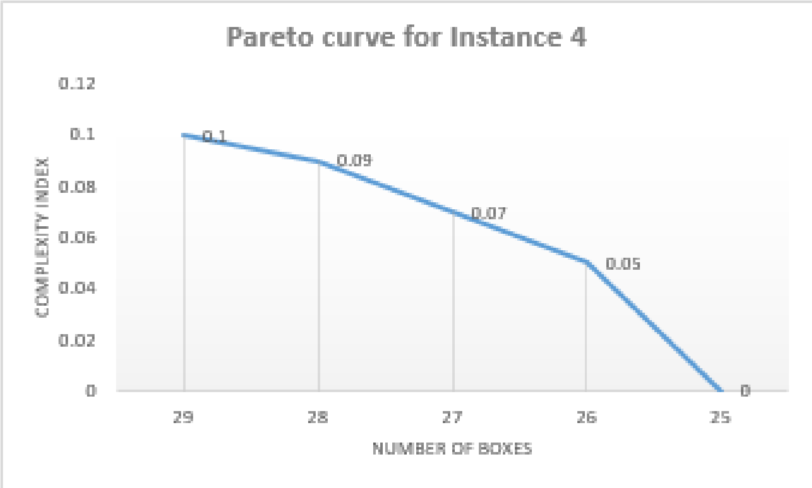


Figure 3.7: Pareto curve for Instance 4

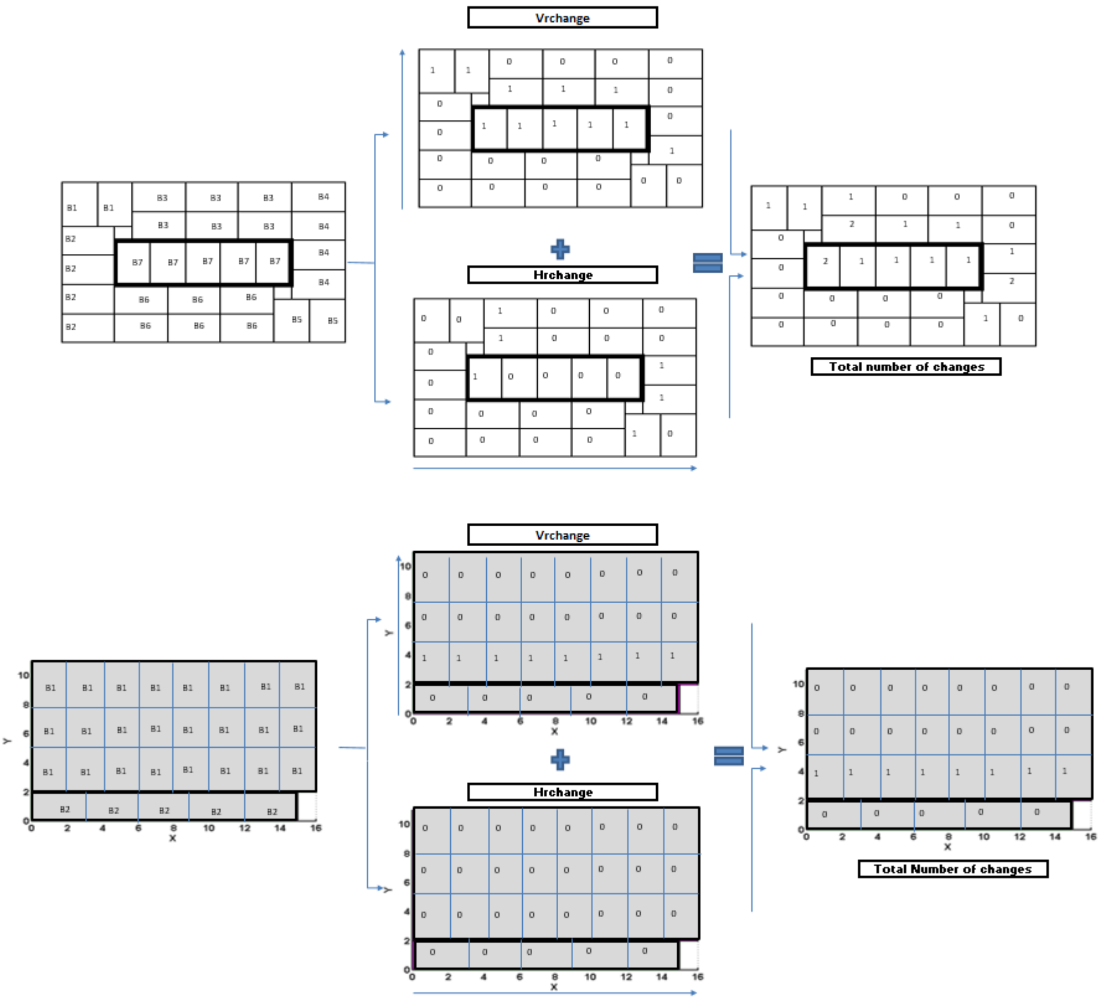


Figure 3.8: Calculation of Complexity Index for Instance 4

2 blocks, while in literature 7 blocks have been used. This significant improvement of reducing the number of blocks from 7 to only 2 blocks is an advantage for the supply chain operations; the detailed steps of calculations for the HVC have been presented in figure 3.8. As seen, a box is assigned a value of 1 once a change of block size is detected in the horizontal and vertical space, and a value of 0 if no changes are detected. The number of changes is summed up, and the number of boxes on the first row and column Xr and Yc are calculated, then the complexity index ζ is found. The results are displayed in table 3.3 where it could be seen that the complexity index ζ for the *m_MPLP* model is 0.10, very close to zero; which results in a less complex graphical layout compared to literature complexity index ζ of 0.36, i.e., a more complex graphical layout. Looking at both the complexity index ζ value and the number of blocks used on the pallet, the proposed model clearly outperforms the graphical literature layout in terms of complexity while loading the same number of boxes.

A comparison between the layouts generated from the *m_MPLP* model against literature layouts for instances 5, 14, 17, 18, 21 and 51 are presented in figure 3.10, the Pareto curves are displayed in figure 3.9 and the complexity index calculation is presented in table 3.4. It could be seen that the layouts generated for all the instances for the proposed approach have more simple graphical layouts than those presented in the literature. The simplicity arises in both the number of blocks used and the complexity index ζ found by the formula presented in equation 3.19.

For instance 5, compared with literature layout (Arenales and Morabito, 1995), it could be noticed that the layout generated from the *m_MPLP* model has a ζ value of 0.28 with 42 boxes loaded compared to literature ζ value of 0.33. If less complex layouts are required, a $zeta$ value of 0 will load a total of 35 boxes, all sharing the same ordination in only one block as in 3.9. The number of blocks needed to construct the layout are 7 blocks for the *m_MPLP* model compared to 8 blocks for literature. It might seem that the number of blocks or complexity measure is not far from the literature, but when looking into daily operations of hundreds or even thousands of pallets to handle, such reduction in complexity would make a

Table 3.4: Complexity Index detailed calculations

Instance	Approach	VC	HC	$\sum_i BO_i$	Xr	Yc	ζ	Number of Blocks
	m_MPLP	13	7	42	6	7	**0.28**	7
5	Literature	12	12	42	6	6	**0.33**	8
	m_MPLP	6	1	46	10	6	**0.09**	3
14	Literature	26	5	46	9	5	**0.40**	9
	m_MPLP	7	5	38	8	5	**0.19**	4
17	Literature	21	8	38	6	5	**0.45**	14
	m_MPLP	15	10	149	12	9	**0.09**	4
18	Literature	126	140	149	15	9	**0.97**	142
	m_MPLP	11	5	99	12	9	**0.09**	3
21	Literature	14	12	99	11	9	**0.15**	7
	m_MPLP	25	10	147	15	11	**0.13**	5
51	Literature	35	17	147	11	9	**0.19**	11

difference and ease the supply chain operations.

For instances, 14 and 17 compared with literature layout (Birgin et al., 2010); again, the number of blocks have been reduced from 9 and 14 blocks in literature to 3 and 4 blocks only for the *m_MPLP* model, respectively, that is more than 50 % reduction in the number of blocks needed to construct the pallet. Also, the complexity index ζ has been reduced from 0.40 to 0.09 for instance 14, and from 0.45 to 0.19 for instance 17; resulting in less complex layouts from the *m_MPLP* model. Again the Pareto curves are displayed in figure 3.9.

For instance 18, the *m_MPLP* model layout outperforms the literature layout (Scheithauer and Terno, 1996b) with a vast difference in the complexity index ζ, being reduced from 0.97 in literature to only 0.09 for the *m_MPLP* model, resulting in a much simpler layout formed using only 4 blocks. Similarly, instances 21 (Martins and Dell, 2008) and instance 51 (Birgin et al., 2010) show a great reduction of over 50 % in the number of blocks used to construct the pallet with a lower complexity index ζ value.

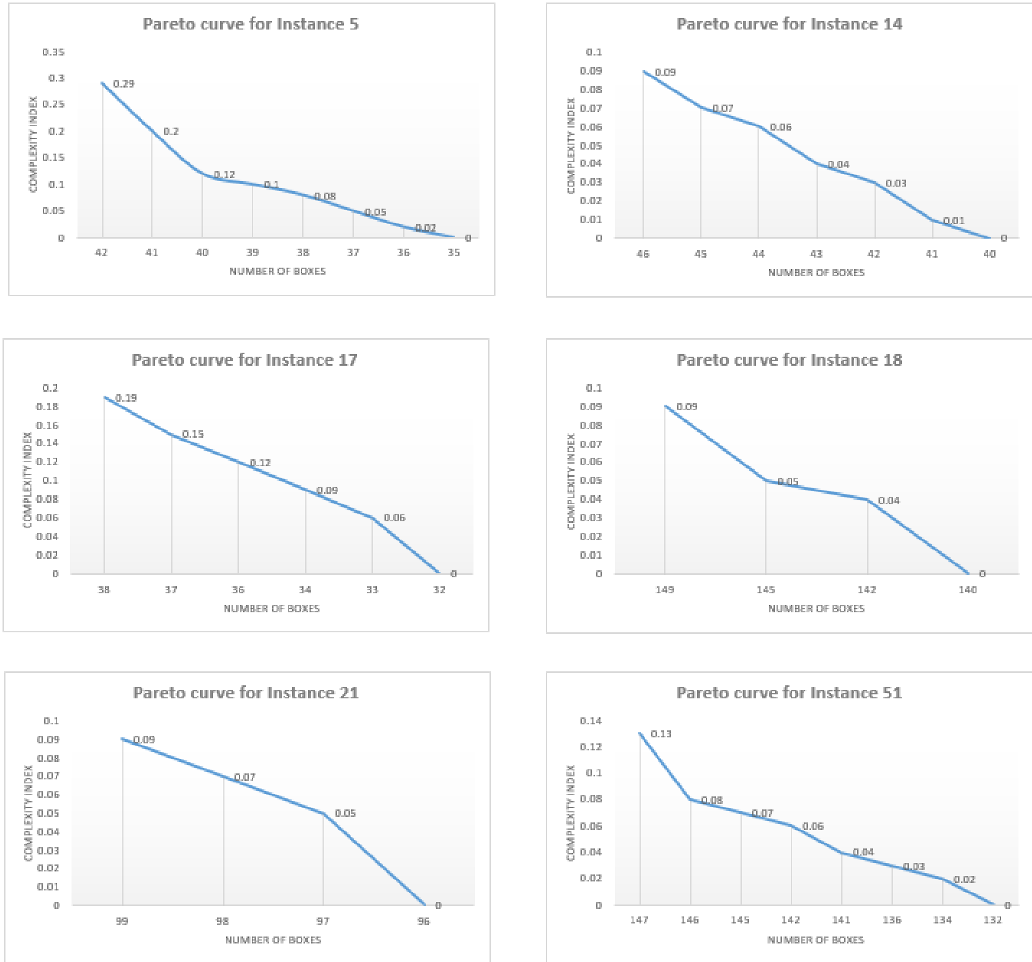


Figure 3.9: Pareto curves

From the above comparisons, the layouts generated by the *m_MPLP* model compared to those from literature show a very promising complexity index difference. Where the layouts have a reduced number of blocks and lower Complexity indexes ζ, while reaching the same optimum solutions for the number of boxes loaded per pallet as in literature. The following section compares the single and multi-objective models; *s_MPLP* and *m_MPLP*, respectively.

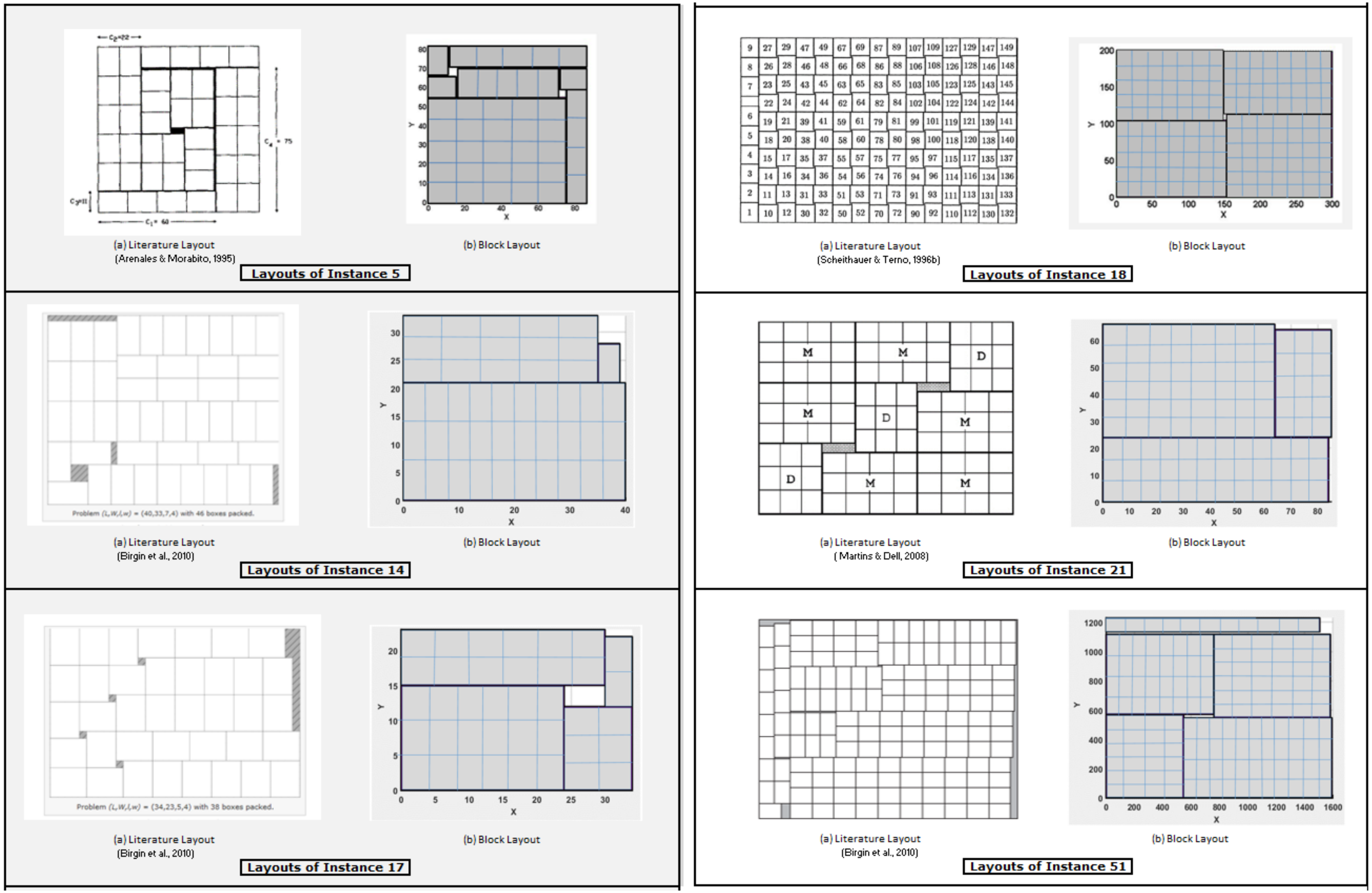


Figure 3.10: Graphical layouts comparisons

3.4.4 Models *s_MPLP* and *m_MPLP* Comparison

Both models, *s_MPLP* and *m_MPLP* aim to maximise the number of boxes loaded onto the pallet; the main difference arises in how the complexity index ζ is found. In the *s_MPLP* model, ζ is found by a post-processing step, where it is calculated manually for each instance and only used as a comparison method for the complexity of pallet loading. In contrast, for the *m_MPLP* model, the complexity index is integrated within the model, offering a unique feature of a trade-off between the complexity of loading and the desired number of boxes to be loaded by tuning the value of ζ based on user or specific managerial/customer decisions.

Below in tables 3.5 and 3.6 a comparison between the single- and multi-objective models; *s_MPLP* and *m_MPLP*, respectively, is displayed. The computational effort is presented using the same data-sets from the previous section; data-sets (a) and (b). Where three additional columns have been added; *s_MPLP*, *m_MPLP*, and % of change, they represent the single-objective model computational time, the multi-objective model computational time; both measured in seconds and the % of time improvement between the *s_MPLP* and the *m_MPLP* model.

It could be seen from table 3.5 that both models *s_MPLP* and *m_MPLP* were capable of loading the same number of boxes as those reported in literature and previously displayed in tables 2.4 and 3.1. The average computational time for the *s_MPLP* model is just under 400 seconds, whereas for *m_MPLP* model, the average time is under 140 seconds. The reduction in the computational effort from 400 seconds to about 140 seconds from the *s_MPLP* model to *m_MPLP*, respectively, shows around 185% time effort improvement.

Now, when exploring Data-set (b), presented in table 3.6 it could be seen that the sizes of the instances tested here are considered very large, having an average pallet size of 2150X1348 cm and an average box size of 139X91 cm. Furthermore, when looking into the computational times, the *s_MPLP* model averages about 600 seconds, and the *m_MPLP* model has an average of about 300 seconds. Again, considering the NP-hard nature of the problem and the sizes of the instances, the computational effort reduction of about 50% is considered very good.

Table 3.5: Comparison between *s_MPLP* and *m_MPLP* computational times for data-set (a)

#	L	W	α	β	Boxes loaded	*s_MPLP* (s)	*m_MPLP* (s)	% of change
1	1000	1000	205	159	30	16.64	3.7	350%
2	1000	1000	200	150	33	615.8	433.5	42%
3	14	10	3	2	23	4.3	0.1	4200%
4	16	11	3	2	29	10.68	0.2	5240%
5	86	82	15	11	42	863.1	22.4	3753%
6	30	22	7	4	23	32.2	28.1	15%
7	46	34	11	6	23	63.2	29.8	112%
8	50	36	11	7	23	53.87	35.5	52%
9	53	51	9	7	42	617.8	301	105%
10	63	60	11	8	42	331.2	222.9	49%
11	76	73	13	10	42	1000	247.3	304%
12	87	47	7	6	97	1000	231.6	332%
13	57	44	12	5	41	1000	265.9	276%
14	40	33	7	4	46	193.2	4.2	4500%
15	3750	3063	646	375	46	175.6	13.7	1182%
16	1200	800	176	135	38	114.7	9.1	1161%
17	34	23	5	4	38	123.8	10.8	1046%
18	300	200	21	19	149	535.2	432.7	24%
19	40	25	7	3	47	623.2	283.5	120%
20	52	33	9	4	47	531.1	203.2	161%
21	85	66	8	7	99	422.6	231.8	82%
22	43	26	7	3	53	389.5	155.9	150%
23	153	100	24	7	90	470.1	231.5	103%
24	42	39	9	4	45	348.1	148.4	135%
25	124	81	21	10	47	692.2	158.3	337%
26	100	64	17	10	36	247.3	23.7	943%
27	100	82	22	8	45	498.5	3.7	13372%
28	100	83	22	8	45	503.7	15.9	3068%
29	32	22	5	4	34	313.6	5.1	6049%
30	32	27	5	4	42	395.3	4.2	9312%
31	40	26	7	4	36	335.2	8.3	3939%
32	53	26	7	4	48	168.4	45.9	267%
33	37	30	8	3	45	207.5	68.7	202%
34	81	39	9	7	49	195.5	18.2	974%
35	61	38	6	5	77	597.7	173.4	245%
36	63	44	8	5	69	263.8	204.3	29%
37	61	35	10	3	71	426.3	387.3	10%
38	61	38	10	3	77	658.3	238..4	176%
39	93	46	13	4	82	432.4	218.9	98%
40	106	59	13	5	96	545.9	136.8	299%
41	141	71	13	8	96	164.5	153.8	7%

Continued

#	L	W	α	β	Boxes loaded	*s_MPLP* (s)	*m_MPLP* (s)	% of change
42	108	65	10	7	100	212.4	207.5	2%
43	86	52	9	5	99	657.1	326.3	101%
44	74	46	7	5	97	543.4	217.8	149%
45	67	44	6	5	97	231.7	273.6	113%
46	49	28	8	3	57	117.8	152.7	206%
47	57	34	7	4	69	124.8	112.6	366%
48	67	37	11	3	75	62.4	75.8	114%
49	67	40	11	3	81	80.2	85.8	227%
50	74	49	11	4	82	61.9	69.3	162%
Average						395.9	138.7	185%

Table 3.6: Comparison between *s_MPLP* and *m_MPLP* computational times for data-set (b)

#	L	W	α	β	Boxes loaded	*s_MPLP* (s)	*m_MPLP* (s)	% of change
L1	2296	1230	136	94	219	321.2	51.5	84%
L2	2536	1312	144	84	273	515.3	205.4	60%
L3	2252	1470	144	84	271	510.7	285.3	44%
L4	1470	1458	144	84	175	689.4	156.3	77%
L5	2296	1230	135	92	226	350.4	122.5	65%
L6	1804	1230	137	95	168	618.4	232.8	62%
L7	2466	1230	137	95	231	974.2	832.4	14%
L8	1804	1750	137	95	240	492.7	232.5	60%
L9	2426	1230	137	95	227	873.9	612.6	29%
Average time						605.13	303.47	49.8%

The model statistics vary when testing different instances due to the fact that the values for the sets $r, \bar{r}$ are different from one problem to the other depending on the magnitude of the pallet/boxes tested. Where the value for the set is calculated using the following equation: $\max(L,W)/\min(\alpha,\beta)$, indicating the max number of boxes that could fit along the X or Y axis if the smallest dimension is selected. Based on the above and to show how both models, the *s_MPLP* and the *m_MPLP* compare, Instance 1 from table 3.1 has been selected. The model statistics are displayed in table 3.7. And as seen, the *m_MPLP* model continuous and discrete variables are higher than those reported for the *s_MPLP* model, but due to the ε

constraint method used, where the model is constrained, resulting in a reduced search space; therefore, the computational times reported are better.

Table 3.7: Model statistics comparison

	Model	
	s_MPLP	*m_MPLP*
Number of discrete variables	435	653
Number of continuous variables	706	1258
Number of equations	28	44

3.5 Concluding remarks

In this chapter, a novel multi-objective Mixed Integer Linear Programming (MILP) model using the ε constraint method has been proposed to address the Manufacturer's Pallet Loading Problem (MPLP) problem, in which the number of boxes loaded is the single-objective to be optimised and the complexity index ζ is transferred into a constraint. The model proposed maximises the number of boxes loaded on the pallet while grouping boxes that share the same orientation into blocks. The number of blocks required can vary from one problem to the other depending on the problem size. The algorithm has been tested against a wide range of data sets, and the same optimum results have been obtained. Also, the algorithm provides the exact geometrical location of the blocks and provides less complex graphical layouts that outperform those from the literature. Finally, a comparison between the single- and multi-objective models (*s_MPLP*, *m_MPLP*) computational effort was presented, where the multi-objective model has a significant improvement overall in all instances tested.

The main contribution lies in using a multi-objective linear approach that groups boxes of the same orientation across the X and Y axis into blocks, reducing the number of orientation changes within the pallet using a novel Complexity measure. The Complexity measure here offers an excellent advantage for the user, offering a set of layouts ranging from one block to multi blocks, where a trade-off between the complexity of loading and the desired number of boxes to be loaded is user-tailored.

Chapter 4

Automated Design of Assortments

This Chapter addresses Automated Design of Assortments under the 11th AIMMS-MOPTA optimisation Modelling Competition. A Mixed Integer Linear Programming (MILP) algorithm that can assist in providing the user with a variety of item assortments has been developed. An Integer cut approach has been applied to provide a diverse set of solutions, not allowing the same solution to be repeated amongst the set of iterations. The model has been tested against the data sets provided by finding the maximum number of items to be packed into rectangular containers with their respective exact geometric locations. A user-friendly GUI has been developed in AIMMS platform to facilitate user interaction with an option to tune the near-optimality parameter and a graphical display of the solution assortments.

4.1 Introduction

The problem defined as ”Automated Design of Assortments” was presented in the 11th AIMMS-MOPTA optimisation Competition 2019, and up to our knowledge, no prior study has been done on such problems. As a consequence, we instead offer a summary of the literature on its sub-problem in the Cutting and Packing (CP) area.

4.2 Literature Review

As discussed in the introduction, the Cutting and Packing (CP) problems have gained wide attention over the past years. The topic manifests itself in several ways, with varying titles depending on the industry it impacts. Such as cutting stock, trim loss, bin packing, strip packing, vehicle loading, pallet loading, container loading, assortment, design, layout, capital budgeting, memory allocation, and multiprocessor scheduling problems as described by Dyckhoff (1990) and the complete phenomenology of C&P problems can be seen in figure 4.1.

Such problems, in general, contain two items, small and large, either cutting small lengths of items from a large sheet or area or loading predefined sized items into a larger container or bin, subject to the objective function defined. An example of the typical structure of the C&P problem can be seen in figure 4.2. The objective can either be minimising the trim or space loss, minimising the cost of materials or transportation, maximising the number of items loaded, or the total profit. The first mathematical definition of the problem could be traced back to the 1930s (Kantorovich, 1960), but no attention was given to such problems until the 1960s when the paper was translated to English according to Dowsland and Dowsland (1992).

Such problems can be viewed from a 1, 2, or multi-dimensional aspect, and the shape of items can vary between rectangular to irregular shapes for both the small and large items. The first typology of Cutting and Packing (CP) problems was defined by Dyckhoff (1990), where they have classified the problem according to its dimensionality, the type of assignment, assortment of large items, and the assortments of small items. The drawback of Dyckhoff's typology has been discussed in detail by Wäscher et al. (2007) indicating the weaknesses and shortcoming of the classification system, where the authors have introduced an improved typology to define a more consistent system for this area looking into all sub-problems of the C&P problems based on nearly 300 papers published in the area over the last decade.

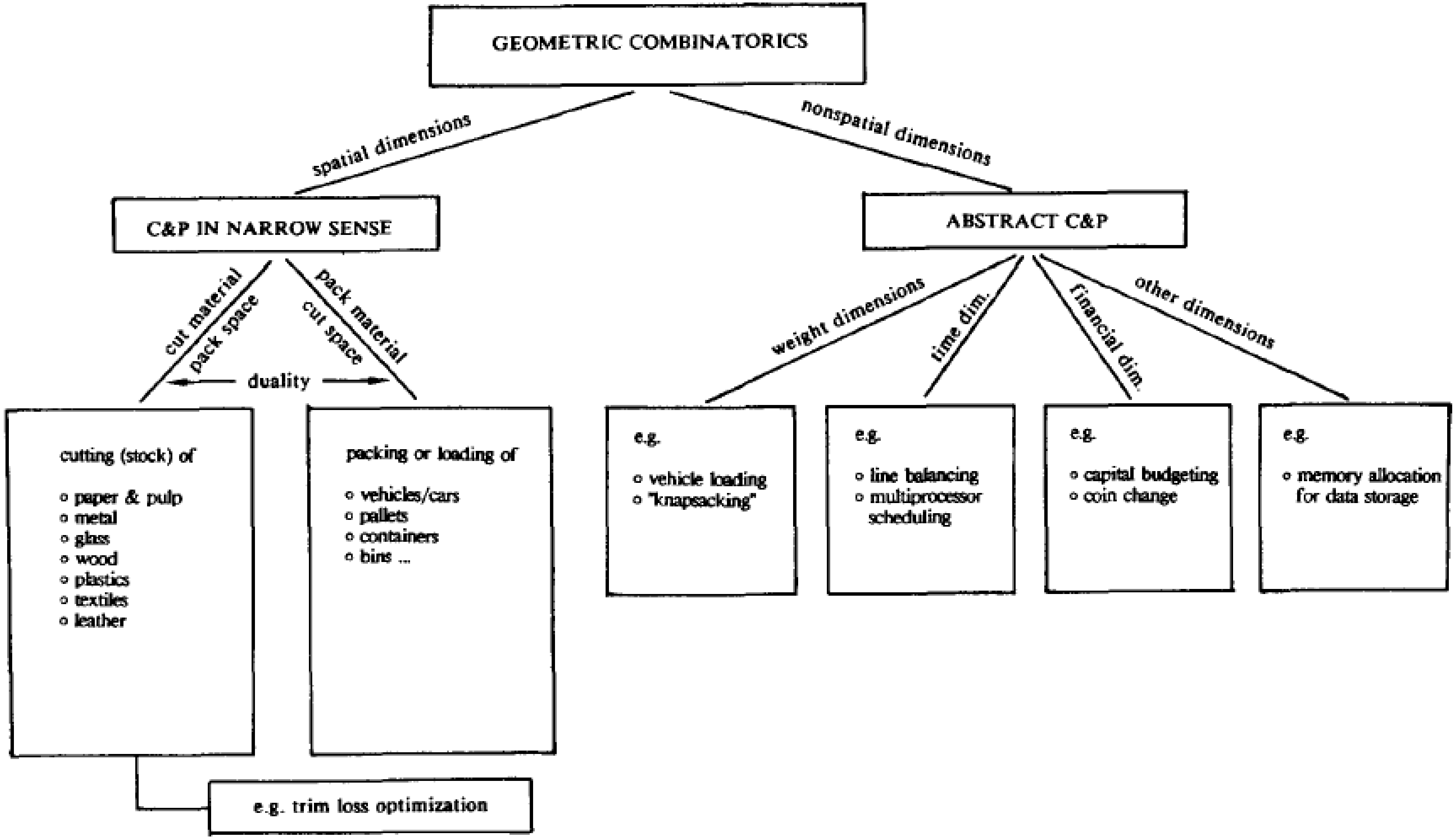


Figure 4.1: Phenomenology of C&P problems (Dyckhoff, 1990)

Figure 4.2: Typical Cutting and Packing problem structure

Also, Lodi et al. (1999) have defined another typology for the two-dimensional Cutting and Packing (CP) problems based on the orientation and cuts of the items, where they have defined four types of problems, below is the typology proposed:

- 2BP |O |G: where the items are Oriented (O); cannot rotate, and the guillotine (G) cut is necessary
- 2BP |R |G: where the items may Rotate (R) by 90°and the guillotine (G) cut is necessary
- 2BP |O |F: where the items are Oriented (O) and the cutting is Free (F)
- 2BP |R |F: where the items may Rotate (R), and the cutting is Free (F)

In literature, many solving techniques for the problem have been proposed, ranging from heuristics to mathematical modeling. Coffman et al. (1984) offers a concise description of the different approximation algorithms proposed for the Bin-Packing Problem, as well as those that are relevant to the Cutting Stock Problem. The authors analyse the algorithms' worst-case and average behaviours. The fact that such problems are well known to be NP-hard motivates the substantial investment in heuristic approaches. One of the exact methods proposed for the Bin-Packing Problem is Marcotte (1986), where a branch-and-bound method using appropriate lower bounds for the problem in combination with heuristic approaches at each node was considered. Scholl et al. (1997) method combining the branch and bound method with reduction methods, lower limits, and heuristics, as well as a tabu search procedure, yielded better results. As seen in these methods, quick computing of high lower bounds is crucial; an alternative approach to computing lower bounds for the problem were suggested by Fekete and Schepers (2001) where the authors reduce the CPU time required to verify the viability of subsets of rectangles that is if they can be loaded in the container and thereby form a feasible assortment.

The C&P problems have been solved with considerable advantages using linear programming-based algorithms; Gilmore and Gomory (1963) were the leaders, merging column generation with rounding to achieve near-optimal solutions. Cutting planes or merging column generation with branch-and-bound have been used in more recent attempts (De Carvalho, 1999; Degraeve and Peeters, 2003; Scheithauer et al., 2001; Vanderbeck, 1999). Other Mixed Integer Linear Programming (MILP) models have been developed to reduce the number of variables and constraints as in Belov et al. (2009); where the model constraints identify if items overlap on the X and Y axis and only one solution is allowed in the assortment. Egeblad and Pisinger (2009) use binary variables to determine the position of each pair of items, indicating if the pairs are placed over, under, right, or left of one another.

4.3 Problem Definition

According to the Mopta 2019 competition, the problem statement is to develop a model that can assist the user in creating a variety of good assortments of items. Where a set of items I that range from $i = 1, \ldots, N$ is given with its dimensions in width w_i and height h_i. An assortment is a subset S which belongs to I, and an assortment S is feasible if the items can be packed in a rectangular container of width W and height H without overlapping; that is, each item s belongs to S can be assigned a left-bottom corner coordinate (x_s, y_s) such that the rectangles R_s is included in the container, and the items cannot be rotated. The value v_S of a feasible assortment S is the total surface occupied by the items, and the user is interested in assortments that maximise v_S over feasible assortments or achieve near optimality.

It is requested to test the model proposed using two data sets and two different container sizes as the following:

- Container size of 300X400 and 200 different item sizes
- Container size of 500X500 and 40 different item sizes

12 different layouts/assortments for the first data set and 20 different assortments for the second data set have to be presented. A diversity measure has to be established to measure the differences in layouts.

4.4 Mathematical Model

The assortment problem can be defined as individual items i of given dimensions in width w_i and h_i, where these individual items are packed into containers with sizes defined as X^{max} and Y^{max} for the container width and height. The detailed Mixed Integer Linear Programming (MILP) mathematical formulation of the model is presented below.

The formulation is based on the following notations:

Indices

i, j	Items
k, kk	Loops
p	Number of desired solutions

Parameters

X^{max}, Y^{max}	X,Y Container Dimensions
w_i	Width of i
h_i	Height of item i
v_i	Volume of item i
M	Upper-bound value
ε	Near-optimality parameter
$zmax$	Max number of solutions obtained over all iterations
Upp_{ik}	Integer cuts parameter
Low_{ik}	Integer cuts parameter
$FinalZ_p$	All solutions satisfying the ε condition

The following variables are associated with the model:

Positive variables

X_i	X geometrical coordinate of item i
Y_i	Y geometrical coordinate of item i
TIU	Total number of items used in the container
CU	Container utilisation percentage

Binary variables

q_i	Selection of items, 1 if item i is used, 0 otherwise
$E1_{ij}$, $E2_{ij}$	Non overlapping binary variables

4.4.1 Objective Function

The objective is to maximise the volume of the items i packed in the container as stated in equation 4.1.

$$Max \sum_i v_i \cdot q_i \tag{4.1}$$

4.4.2 Model Constraints

Lower-bound:

Lower bound constraints on the coordinates of the geometrical center of each item i and j have been considered to avoid intersection of items with the origin of axis as in the equations 4.2 and 5.2.

$$X_i \geq \frac{w_i}{2} \quad \forall i \tag{4.2}$$

$$Y_i \geq \frac{h_i}{2} \quad \forall i \tag{4.3}$$

Upper-bound:

In a similar way, upper bound constraint force the items to be allocated within the container dimensions, and the rectangular space is defined by the corners (0,0) of the container dimensions X^{max}, Y^{max} as the following:

$$X_i + \frac{w_i}{2} \leq X^{max} \quad \forall i \tag{4.4}$$

$$Y_i + \frac{h_i}{2} \leq Y^{max} \quad \forall i \tag{4.5}$$

Non-overlapping Constraints:

To avoid items i and j overlapping or occupying the same location in the x and y axis we have introduced $E1_{i,j}$ and $E2_{i,j}$ binary variables as in Papageorgiou and Rotstein (1998). Where M represents an upper bound value equal to the maximum pallet volume of $X^{max} \cdot Y^{max}$.

Non overlapping in the X direction

$$X_i - X_j + M(2 - q_i - q_j + E1_{i,j} + E2_{i,j}) \geq \frac{w_i + w_j}{2} \quad \forall j \geq i \tag{4.6}$$

$$X_j - X_i + M(3 - q_i - q_j - E1_{i,j} + E2_{i,j}) \geq \frac{w_i + w_j}{2} \quad \forall j \geq i \tag{4.7}$$

Non-overlapping in the Y direction

$$Y_i - Y_j + M(3 - q_i - q_j + E1_{i,j} - E2_{i,j}) \geq \frac{h_i + h_j}{2} \quad \forall j \geq i \tag{4.8}$$

$$Y_j - Y_i + M(4 - q_i - q_j - E1_{i,j} - E2_{i,j}) \geq \frac{h_i + h_j}{2} \quad \forall j \geq i \tag{4.9}$$

4.4.3 Additional Diversity Design Constraints

Total items used (TIU)

In addition to the container utilisation percentage, we have considered the total number of items selected in an assortment as a diversity measure as in equation 4.10.

$$TIU = \sum_i q_i \tag{4.10}$$

Integer cuts

Integer cuts have been applied to the MILP model to exclude duplicated solutions over the number of iterations required to find optimal solutions. Two additional sets have been created to include the items selected over each loop k as; $u_{i,k}$ and $l_{i,k}$. The integer cut equation is demonstrated in equation 4.11.

$$\sum_{i \in u_{i,k}} q_i - \sum_{i \in l_{i,k}} q_i \leq |u| - 1 \quad \forall \quad k \tag{4.11}$$

Thus, the model consists of Objective function equation 4.1 subject to constraints 4.2-4.11.

Near optimality tuning parameter

As stated in the problem description above; for a set of assortments to be considered good, they have to be within a percentage difference. So a near optimality tuning parameter known as ε has been added and allows the user to choose the maximum desired percentage of the difference between solutions. To allow this, two loops have been created; an outer loop and an inner loop known as k and kk where the outer loop k represents the total number of iterations taking into consideration that the number of iterations (30 in this model) must exceed the number of desired solutions set by the user (12 and 20 in this model). We consider the above to ensure the model can always satisfy the ε condition, and the number of different assortments is always found. Moreover, the number of iterations was controlled by a break statement.

The model can be defined in more detail as the following; Stage 1 represents the outer loop; in this stage, we solve the model with all the constraints applied. Stage 2 represents the conditional statement within the outer loop; the main purpose of this stage is to identify the maximum objective value for all solutions and save it into a new parameter known as *zmax*. Stage 3 represents the inner loop known as *kk*, in this loop, we apply the ε condition statement across all solutions obtained so far; if the solution satisfies the ε condition, it is then saved into a new parameter known as $FinalZ_p$. Stage 4 is a break statement for the outer loop k that only applies when the number of solutions satisfying the ε condition has been met. In other words, the number of iterations in the outer loop depends on the number of solutions that satisfy the ε condition in the inner loop. To obtain this solution, the following conditions have been added:

```
STAGE 1: solving model with all constraints:
    for (k) {
     do "OuterLoop"
     solve model

STAGE 2: Finding zmax (maximum objective value over all
iterations k):
    {if (counter = 0) {then
```

```
     zmax := z.l
};
    elseif (counter >0 ) and (z.L > zmax) {then
     zmax := z.l
};
    endif
};
```

STAGE 3: for every iteration kk we apply the epsilon condition using the new zmax for all solutions in zn(kk), if the condition is satisfied for any zn(kk) it is then saved into parameter FinalZ(p+counter3). For illustration, the indices (k−counter1+counter2) and (p+counter3) are an incremental process to cover all previous solutions and to fill the results that satisfy the epsilon condition respec −tively:

```
    counter1 :=0;
    counter2 :=0;
    counter3 :=0;
    for (kk) {do
    zn(kk) := objective_value(k−counter1+counter2);
     if (zn(kk) >= ((1−epsilon)* zmax)) {then
     (FinalZ(p+counter3) := zn(kk))
};
     counter3 := counter1+1;
     endif
};
    counter2 :=counter2+1;
    endfor
  };
```

```
STAGE 4: Break outer loop when the number of assortments
 that meet the epsilon condition are satisfied:
break {"OuterLoop"
     when (counter1 = card(p)+1)
     };
     counter1 := counter1+1;
Endfor
};
```

4.5 Results

To test the validity of the proposed model ,it has been tested against 2 data sets as in table 4.1; the data sets were solved using AIMMS software and Cplex solver on a PC with a 1620 CPU and 16GB RAM and a time limit of 200 sec/iteration detailed results are presented below:

Table 4.1: Data-Sets Information

Data set	**Container size**		**Epsilon**	**Iterations**
	Width	Height		
1	300	400	0.025	12
2	500	500	0.025	20

For the first data set of 200 different items, the results are in table 4.2 displaying the Objective function solution of the Total Volume TV including the diversity measures of the Container utilisation CU, and the Total Items Used in the assortments TIU.

As it can be seen from the table above, table 4.2 the highest container utilisation percentage CU over the set of solutions is 97.147% with a container total volume

Table 4.2: Results of Data set 1

Iteration	TV	CU	TIU
1	115225	96.02083	4
2	115477	96.23083	4
3	115781	96.48417	5
4	115019	95.84917	4
5	115622	96.35167	4
6	115271	96.05917	4
7	114363	95.3025	4
8	116577	97.1475	5
9	115335	96.1125	4
10	114831	95.6925	4
11	114343	95.28583	4
12	116215	96.84583	5

TV of 116577 and the lowest is 95.285% with a total volume of 114343 not exceeding the near-optimality parameter (ε) of 0.025 between the highest and the lowest solution in the set. The number of items chosen in the assortment TIU has a range between 4 and 5 items. The visualisation of the results is displayed in Figure 4.3 using an integrated tool for visualisation in AIMMS.

For the second data set of 40 different items, the results are in table 4.3, again showing the total volume TV, container utilisation CU and the total number of items used in the assortment TIU. From table 4.3 we can see that the max utilisation percentage is 97.9064 % with a Container total volume TV of 244766, and the lowest is 95.5008 with a total volume TV of 238752 % not exceeding the 0.025 ε parameter limit between the highest and lowest solution in the set. The number of items selected in the assortment ranges between 9 and 15 items. The visualisation of the results can be seen in figure 4.4 for all the 20 different solutions generated.

To analyse the results further, we have performed several runs over a range of time periods (50,100,200,500 and 1000 seconds). We have noticed that the container utilisation percentage as the time increases was within a maximum of 1.96% difference between the highest(1000 seconds) and lowest(50 seconds) time limit for data set 1 and within a maximum of 0.963% difference for data set 2.

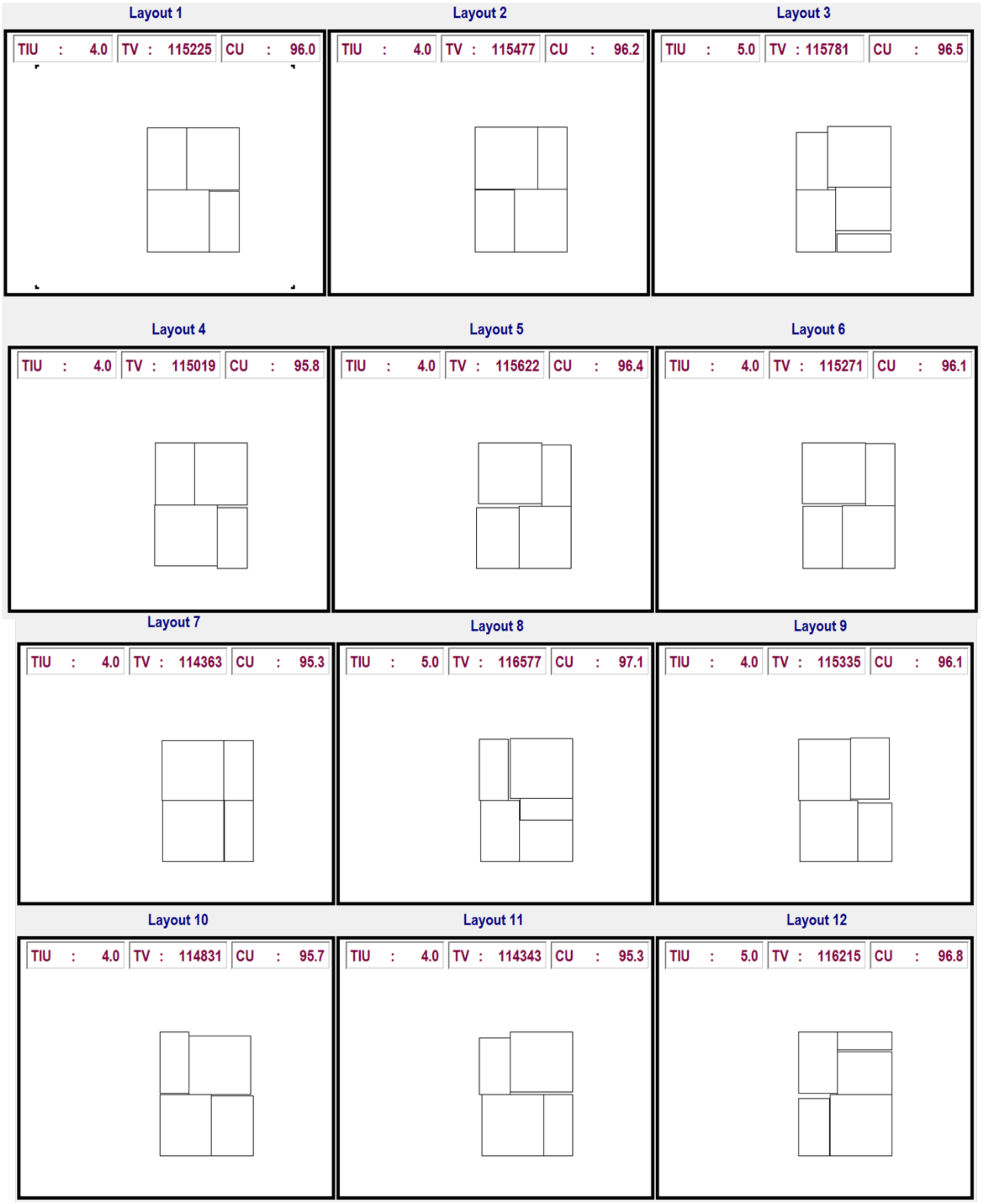


Figure 4.3: Results of Data set 1 for 12 iterations

We have then tested the difference in container utilisation percentage between the highest time limit of 1000 seconds and all the other time limits. In more depth; when looking into the difference for the time limit of 200 seconds, the maximum difference is 0.596% for data set 1 and 0.52% for data set 2, so on that basis, we have decided to set our time limit to 200 seconds as the results are still very ac-

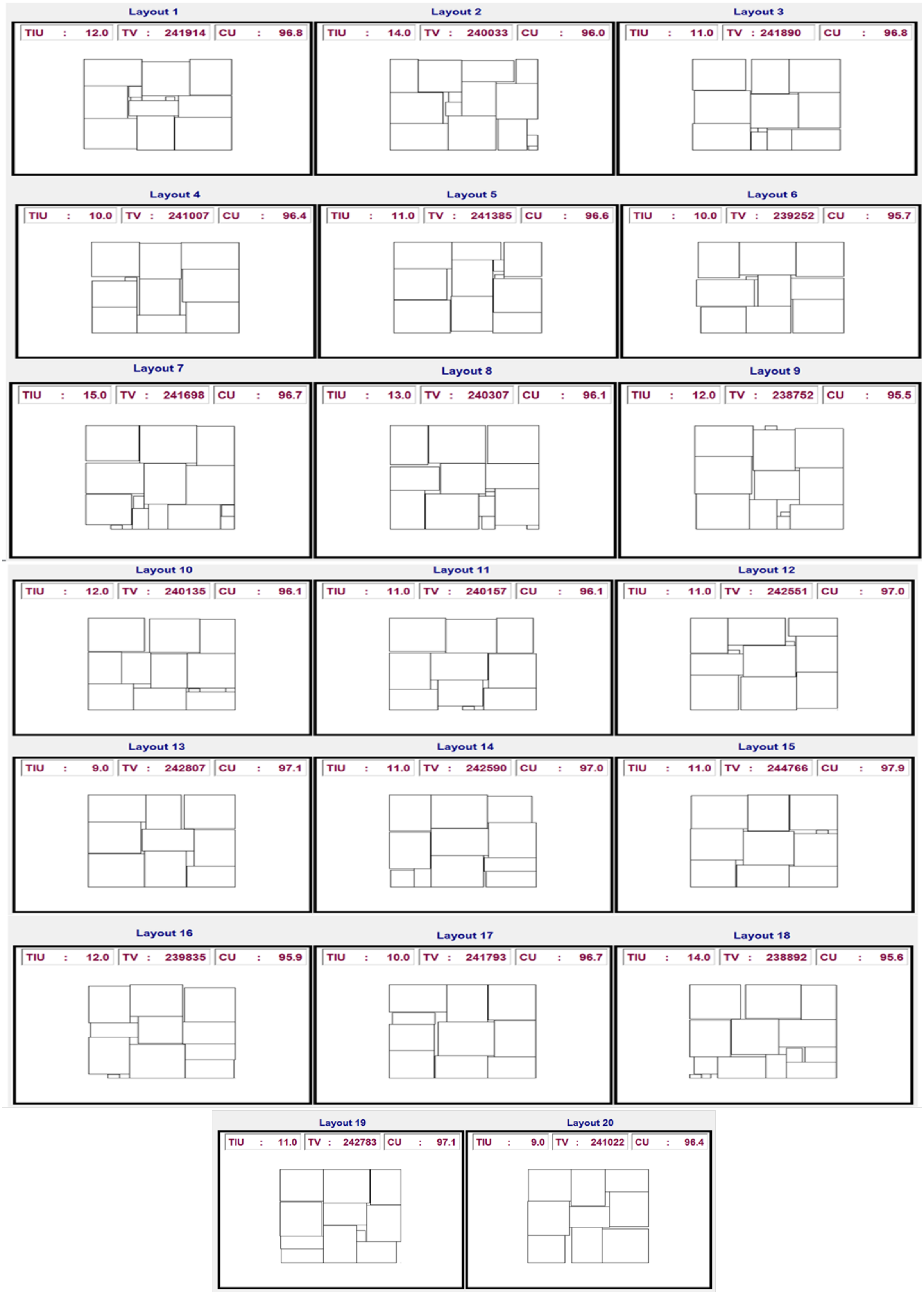


Figure 4.4: Results of Data set 2 for 20 solutions

Table 4.3: Results of Data set 2

Iteration	TV	CU	TIU
1	241914	96.7656	12
2	240033	96.0132	14
3	241890	96.756	11
4	241007	96.4028	1
5	241385	96.554	11
6	239252	95.7008	10
7	241698	96.6792	15
8	240307	96.1228	13
9	238752	95.5008	12
10	240135	96.054	12
11	240157	96.0628	11
12	242551	97.0204	11
13	242807	97.1228	9
14	242590	97.036	11
15	244766	97.9064	11
16	239835	95.934	12
17	241793	96.7172	10
18	238892	95.5568	14
19	242783	97.1132	11
20	241022	96.4088	9

ceptable comparing to the reduction of 80% in terms of CPU time. We have also calculated the standard deviation for both data sets, knowing the range of maximum and minimum utilisation percentages we would be able to reach using our model if several other iterations are performed. Figure 4.5 illustrates both data sets utilisation percentage over the different time periods and the standard deviation for each run.

4.6 Graphical User Interface (GUI)

A GUI interface has been created in AIMMS to aid the user in selecting the ε tuning parameter desired, and a button for the problem description and for running the model is presented. The output results of the Container utilisation CU, the Total Volume of the items in the container TV, the Total number of items used in the assortment TIU are easily displayed on the screen for all the iterations, finally

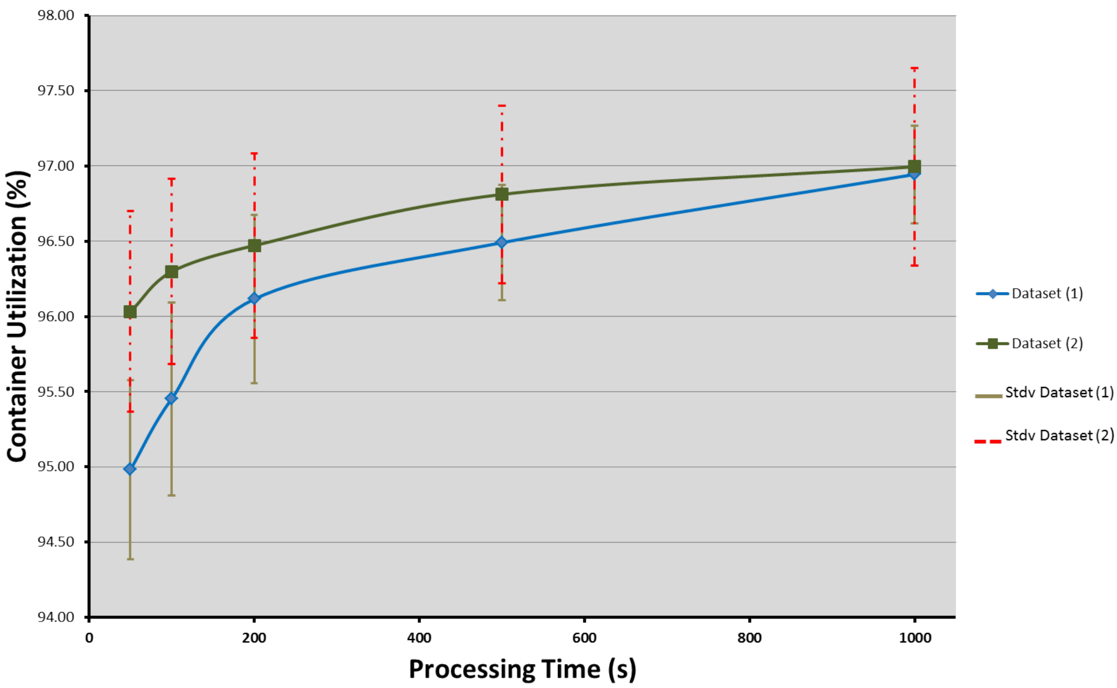


Figure 4.5: Comparison of CU over a variety of time periods

a button leading to a new page representing the visualisation of the results. The detailed visualisation interface created can be found in Appendix B

4.7 Conclusions

In this chapter, a MILP model is presented offering a solution for the Automated Design of Assortment Problem, presented in the 11th AIMMS-MOPTA optimisation Modelling Competition. The suggested linear model uses an integer cut strategy across several iterations to generate various item assortments. The model capabilities and performance were demonstrated through some data sets supplied by the competition committee. Where a variety of item assortments revealing each item's specific geometrical placement has been presented. A Graphical User Interface (GUI) has been built as an extra component to allow user interaction with a graphical presentation of the assortment solutions.

The suggested model was able to determine the total volume and number of items loaded and the container utilisation percentage. The objective was to increase the volume of loaded items while providing alternative solutions that do not exceed the ε near optimality tuning parameter. This parameter is user-defined, and it specifies a percentage of change between all derived solutions, i.e. the highest and lowest container utilisation percentage. For the data sets given, the ε was set to 0.025 %, with a maximum computational time of 200 seconds per iteration. The results obtained have demonstrated a diverse set of assortments, where for data set 1, 12 layouts have been obtained with a container utilisation percentage between 97.147 % and 95.285 %, not exceeding the ε parameter percentage of 0.025 %. And for data set 2, 20 different assortments have been obtained with a container utilisation percentage between 97.90 % and 95.5 %, again not exceeding the ε near-optimality parameter.

As an additional step, several computational time limits have been set to see what effect they could have on the quality of the solutions obtained. Numerous runs with various computational times (50, 100, 200, 500, and 1000 seconds) have been specified for each iteration. When the time limit was changed to 1000 seconds instead of 50 seconds, the maximum container utilisation percentage increased by 1.96 % for data set 1. For data set 2, the percentage of container utilisation improvement has increased by only 0.963 %. All of the other time limits have been examined in a similar manner, and when looking at the percentage of improvement, it was decided that a 200-second run is a reasonable time limit when compared to a 1000 second run, where data set 1 has a difference of just 0.596 % and data set 2 has a difference of 0.52 percent %. In comparison to an 80% reduction in computational effort, the percentage difference is considered relatively low.

Chapter 5

Shelf Space Allocation

In this Chapter, a dynamic framework using Time series linear methods; Linear Regression (LR), Support Vector Regression (SVR), Auto-regressive Integrated Moving Average (ARIMA), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models are proposed to forecast the space elasticity utilising historical data for ten categories over a duration of 5 years. The comparison between approaches is presented, showing the most effective methods for single and multi-step ahead models. Finally, the forecasted space elasticities using the single-step ahead model were used as an input parameter for the Shelf Space Allocation Problems (SSAP) model to compute the number of facing allocated for each category.

5.1 Introduction

The Shelf Space Allocation Problems (SSAP) is defined by the number of products that are displayed on shelves so as to; maximise the sales or gross margins subject to several constraints, such as limited purchasing budget, limited product display shelving space, or seasonality of sales. Clearly, the SSAP management has tremendous effects on the sales, and consequently, retailers, consultants, academia, and software developers have given significant emphasis to such problems.

All the activities associated with the SSAP can be classified under Master category planning, which includes the retailers short to mid-term planning activities from Category sales planning, Assortment Planning, Shelf Space Planning to In-store logistics planning. This study concentrates on Category Assortment and Shelf Space Planning, where the Category Assortment Planning defines what products to include in the store based on consumers' behaviours and substitution effects. Shelf Space Planning defines how and where products should be placed on the shelves taking into account space elasticity and the limited space availability. Figure 5.1 provides a complete overlook on the Master Category Planning.

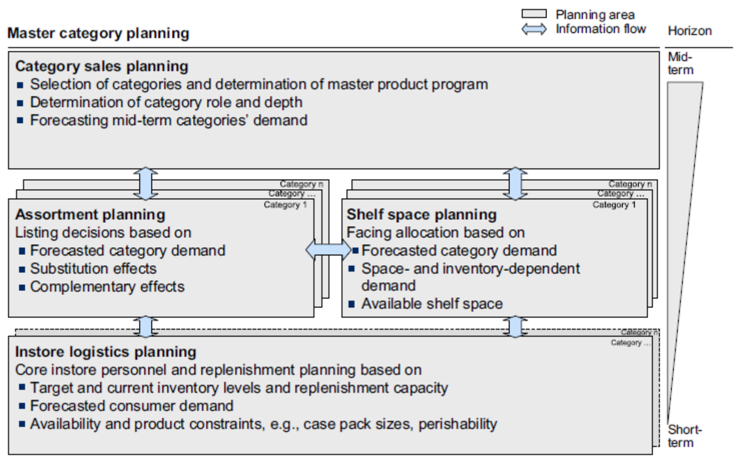


Figure 5.1: Master Category Planning

Since shopping is a daily occurrence in our lives, people tend to spend a lot on daily shopping activities. Retail businesses such as supermarkets usually use this phenomenon in order to make the most of their profit. They often conduct research to influence the customers' purchasing decisions. These retailers' need to maximise the profits compels them to create a design for modelling customer's behaviour and set simulation optimised frameworks. Choosing which items to stock on the shelf and how much space to assign for each product, for example, is a key and critical decision that impacts consumer loyalty and the retailer's profit. Also, when studying consumer shopping behaviors, it has been discovered that certain in-store variables affect buying decisions, especially when unplanned purchasing is made or some loyal items are out of stock. Such behaviours have raised retailers' awareness in smartly displaying the products to achieve better product visibility and ensure the right products at the right time to enhance customer shopping experience (Chandon et al., 2009).

Marketing experts and researchers have defined such management problems as Shelf Space Allocation Problems (SSAP), where the first published research on SSAP, for example, dates from the 1970s (Galai et al., 2016). However, since most optimisation methods developed have practical drawbacks, the research findings are unlikely to be used in practice. The limitations are believed to be caused due to their simplicity as well as a lack of essential functionalities. Furthermore, they are linked to several difficult-to-approximate parameters. Consequently, there has been a misalignment between business practices, software applications, and research. According to the US retailers survey (Keltz and Sterneckert, 2009), improving the overall probability and sales, reducing the stock levels, improving product availability, and enhancing the customer shopping experience are the main drivers for shelf space planning activities. However, the survey concluded that the benefits achieved are not yet meeting the expectations due to many challenges in that area.

The retailers embrace the use of planograms to plan for product placement. A planogram represents a particular illustration of a shop or store. It displays exactly where every product ought to be physically positioned, and the number of facing each product should have; an example of the planogram could be seen in figure 1.3. Today's available commercial planogram tools are used mainly for visual and handling purposes and have excessive human interference and manual adjustment due to the limited existence of mathematical optimisation (Desmet and Renaudin, 1998; Dreze et al., 1994; Hansen et al., 2010; Hubner and Kuhn, 2012; Irion et al., 2011). Also, such software's fail to include demand effects while planning the shelf allocation of the products.

Space elasticity is one of the key features usually included in SSAP models since it determines the relationship between the demand and the space allocated to each product. Furthermore, the relationship has often been described as a concave function because the retailer has to show various items, and displaying one item over a large space does not necessarily increase the sales (only up to a certain limit). Spatial space elasticity estimation techniques have gained research interest but are not in a position to deal with real-life problems due to their difficulty and excessive estimates of parameters. In this study, a dynamic framework is proposed using Time series linear methods; Linear Regression (LR),Auto-regressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models to forecast the space elasticity utilising historical data for ten categories over a period of 5 years.

The remainder of this chapter is structured as follows; a literature review on the issue is given in Section 5.2. In Section 5.3, the problem statement is presented. A display of the category SSAP mathematical model is presented in Section 5.4. Section 5.5 presents the methodology and the types of time series used. A discussion on the data used is explained in Section 5.6, The results and findings for the analysis are summarised in Section 5.7, and finally, Section 5.8 provides a concluding for the chapter with a summary.

5.2 Literature Review

One of the first studies in the area of shelf space allocation problems goes back to Corstjens and Doyle (1981), where the model they presented seeks to maximise the retailer's profit, including a capacity constraint on the shelf space available. They define the demand for products as a function of own and cross-space elasticities. Later, the authors extended the model to a dynamic one (Corstjens and Doyle, 1983), where product growth potential has been taken into account; also, product prices were included in the demand function. Bultez and Naert (1988) extended

Corstjens and Doyle (1981) model, where they incorporated several additional elasticities for the products and classes. The solutions obtained by their model lacks real-life implementation, as they allow non-integer solutions, meaning products can be allocated a space in decimal points. Bookbinder and Zarour (2001), also extended Corstjens & Doyle model; where they incorporated profit contribution to individual products. Zufryden (1986) implemented a model that guarantees integer-valued solutions by dividing the shelf into slot sizes, and the product is a multiple slot size; in their proposed model, they do not allow any orientation of the display.

Borin et al. (1994) also proposed a model for the shelf space allocation by characterising the demand for products as unmodified, modified, acquired, and stock-out demand. The unmodified demand represents the consumer's favourability of a particular product, and the modified demand is the unmodified demand plus the demand lost by price, space, and other retail effects. Acquired demand defines the outcome of the sales from products that have not been included in the assortment. Finally, the stock-out demand captures the sales from other products in the same category being out of stock or unavailable at a specific time. A study by Yang and Chen (1999) simplified Corstjens and Doyle (1981) model by setting upper and lower bounds on the product facing and assuming that the demand function is linear in this case. Their model is the first model to incorporate location effects, although no cross-effects are considered. Later, Lim et al. (2004) extended Yang and Chen (1999) model in two approaches; one deals with a linear profit function while the other considers a non-linear profit function.

Bai and Kendall (2008) proposed a non-linear model ignoring cross-product demand and shelf location effects. Murray et al. (2010) developed a model that jointly optimises a retailer's decisions for product prices, display facing areas, display orientations, and shelf-space locations in a product category. Irion et al. (2012) proposed a piecewise linearisation technique to approximate the non-linear shelf space allocation model; the approach proposed can solve single category-shelf space allocation problems with multiple products and integrated with cost and profit elements.

Although the above existing models have contributed to the area of shelf space allocation, they tend to have several limitations in defining the problem statement and including all the necessary constraints and elasticities. In research focusing on space elasticity, Curhan (1972) attempted to use multiple regression analysis to estimate space elasticity from 11 product characteristics. However, the coefficient of determination was .032, and there is little power for this regression analysis to predict space elasticity. Furthermore, under actual operating conditions, Curhan observed almost 500 grocery goods in 4 stores in a time frame of 5 to 12 weeks before and after changes in the shelf space and the average space elasticity for all items was .212, which indicates a positive relationship between shelf space and unit sales. Heinsbroek (1977) obtained an average shelf space elasticity of 0.15 for 20 items, with a minimum of 0.05 and a maximum of 0.50. Later, Bultez and Naert (1988) created a Sh.ARP (Shelf Allocation for Retailer Profit) optimisation method and the average shelf space elasticity for a diary product was estimated at around 0.30. Finally, 31 traditional studies conducted from 1960 to summer 2012 were surveyed by Eisend (2014) where 1,268 space elasticities were recorded; Eisend concluded that the shelf space elasticity average was 0.17.

5.3 Problem Description

Concerning Space allocation, three decisions are usually considered; the number of facings, which is the space that each item occupies (Space), the allocation of which items to include on the shelves (Allocation), and finally, the geometrical positioning of the items on the shelves (Location). Along with these decisions, the space elasticity β_i; which is defined as the relationship between the facing area of each product and the demand, is often considered taking a value between 0 and 1 (Hoch et al., 1994), and it is usually estimated using regression analysis.

In this chapter, we propose dynamic forecasting for the space elasticity to allow retail stores to increase the total profit taking into account seasonal demands using Time series methods; Linear Regression (LR), Auto-regressive Integrated

Moving Average (ARIMA), Support Vector Regression (SVR), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models to forecast the space elasticity, where the space elasticity β_i is considered as a function of time $\beta_{i,t}$ utilising historical data for ten categories over a period of 5 years.

In the single-step ahead, the sliding window is updated each time using the historical data real value. However, in the Recursive Multiple-step ahead, the current time step's predicted space elasticity $\hat{\beta}_{i,t+1}$ value is used to update the sliding window. Figure 5.2 and 5.3 illustrate the single-step and the Recursive multi-step ahead forecasting strategies, respectively. Two well-used performance measures from the literature were used to compare the models' performance: the Mean Square Error (MSE) and the Mean Absolute Percentage Error (MAPE). Finally, the predicted $\hat{\beta}_{i,t}$ is then used in the optimisation model to optimise the shelf space used for each category i, where a comparison between the shelf space currently used by the retailer and the optimised shelf space is presented along with the current and future profit.

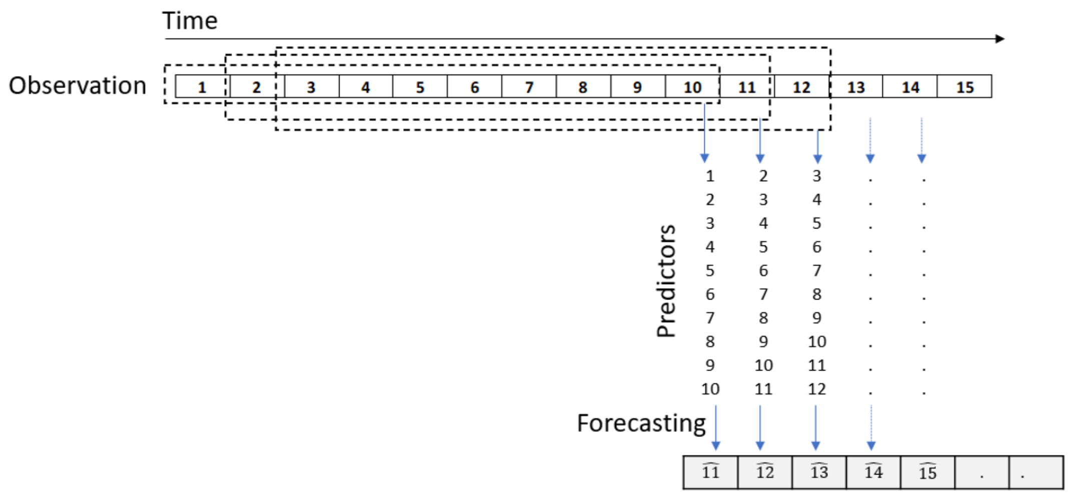


Figure 5.2: Single-step ahead Forecasting Strategy

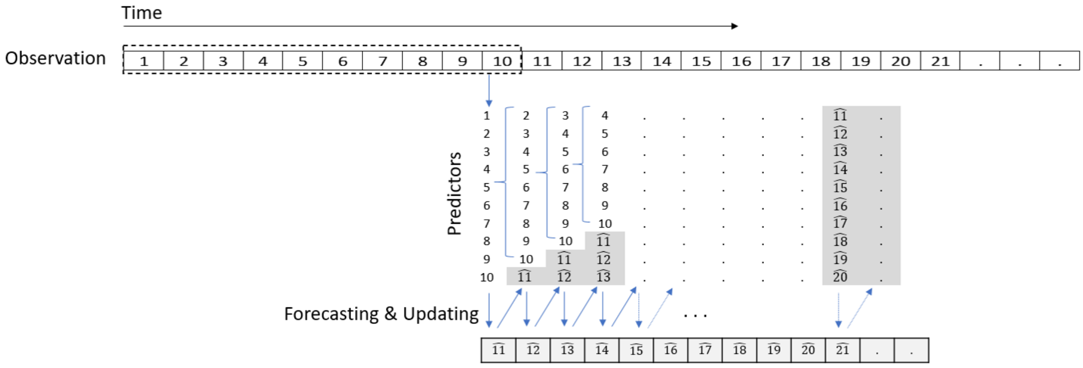


Figure 5.3: Recursive multi-step ahead Forecasting Strategy

5.4 Category SSAP Mathematical Model

As previously mentioned, the retailer's main objective is to maximise the total profit by displaying the right amount of items on the shelves subject to the forecasted demand. The category SSAP problem transformed into a time series model taking into account the space elasticity is formulated as the following, where figure 5.4 illustrates the main notations used:

Indices

i	Categories

Parameters

c_i	Retailers unit cost for category i
p_i	Retailers unit price for category i
l_i	Length of category item i
α_i	Scaling parameter for category i
$\hat{\beta}_{i,t}$	Space elasticity for category i at time period t
S	Length of shelf
LB_{it}	Lower-bound on the amount of facing assigned for category i at time period t
UB_{it}	Upper-bound on the amount of facing assigned for category i at time period t

The formulation is based on the following variables:

Integer Variables

$\hat{X}_{it}$ Number of facing units of category i placed on the shelf in time t

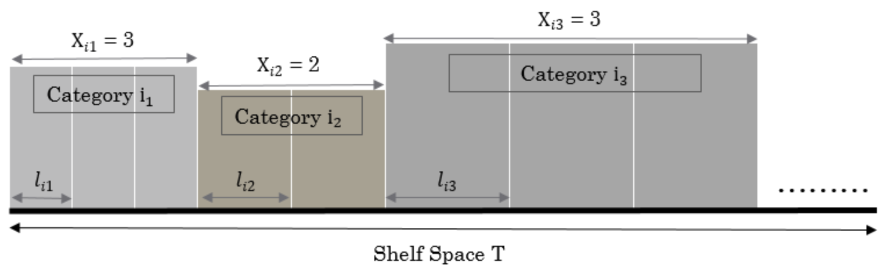


Figure 5.4: Illustration of the notations used for the SSAP model

5.4.1 Objective Function

In SSAP, the relationship between the demand and the space allocated for each category can be described as a polynomial functional form as introduced by Baker and Urban (1988). The relationship is presented in equation 5.1, where, D_i is the demand of category i, α_i is the scaling parameter, X_i is the number of facing for each category i and β_i is the space elasticity of category i. The beneficial features of this model include diminishing returns, shelf space elasticity, and intrinsic linearity. The diminishing returns capture the decreased growth in the demand as the space allocated to the shelf increases. The shelf space elasticity, and as explained before, defines the relationship between the facing area of each product and the demand rate. Finally, the model's intrinsic linearity allows it to be conveniently converted by logarithmic transformation to a linear demand model, after which the parameters can be calculated using any regression technique.

$$D_i = \sum_i \alpha_i (X_i)^{\beta_i} \tag{5.1}$$

where; $\alpha_i > 0, \quad 0 < \beta_i < 1$

The objective of the model is to maximise the total retailer's profit where the profit; the (price - cost) is included, and the function is transformed as a time series model; where $\hat{X}_{it}$ is the number of facing for each category i for each time period t and $\hat{\beta}_{it}$ is the space elasticity of category i for each time period t as the following in equation 5.2:

$$Max \quad \sum_{it}(p_i - c_i)\alpha_i(\hat{X}_{it})^{\hat{\beta}_{it}} \tag{5.2}$$

5.4.2 Constraints

To ensure that the number of categories i placed on the shelf do not exceed the maximum shelf length given S at each time period t equation 5.3 is introduced.

$$\sum_{it} l_i \hat{X}_{it} \leq S \tag{5.3}$$

Equation 5.4 sets the lower and upper bound on the number of facing for each category i if some facing must be limited depending on the retailer's environment decisions.

$$LB_{it} \leq \hat{X}_{it} \leq UB_{it} \qquad \forall \quad i,t \tag{5.4}$$

5.5 Methodology

Forecasting is the practice of predicting future events of a particular phenomenon using its historical facts. In general, Time series forecasting can be divided into two main categories; conventional statistical models and machine learning models (Yoo and Oh, 2020). Moreover, some studies combined both categories into hybrid models. The forecasted horizon varies in length depending on the data used and the outcome desired; it could be short-term, ranging from one hour to one week, medium-term, ranging from one month to one year, or long-term, more than one year. In general, the short-term horizon is often used as in most studies it performs better by achieving better precision than the other forms of time horizons (Yoo and Oh, 2020).

Conventional statistical methods include Linear Regression (LR), Autoregressive Moving Average (ARMA), Auto-regressive Integrated Moving Average (ARIMA), and other general exponential techniques. In the simplest case, the linear regression models allow for a linear relationship between the past and the future variables. The ARIMA models developed by Box and Jenkins are the most commonly used for linear models in univariate time series. Machine learning models are now being used in various applications because of their exceptional capacity to handle complex input and output connections. Machine learning models include the Support Vector Machine (SVM) model, Recurrent Neural Network (RNN) model, and Long Short Term Memory (LSTM) method. The support vector machine (SVM) model has been presented in various applications due to its efficient generalisation capability and margin maximisation mechanisms. In recent years, RNN extends the basic concept of feed-forward neural networks to accommodate sequential data by providing the model with internal memory. Because of its strong representation learning capacity, RNN has emerged as a practical approach; however, such models suffer from exploding and vanishing gradients. Thus, the LSTM network has been introduced, and they intend to learn temporal data, time-stamped data series, and long-term dependencies more effectively than RNN.

In this study, the forecasting of the space elasticity was carried using different methods of Time Series forecasting Models; a Linear Regression (LR) model, an Auto-regressive Integrated Moving Average (ARIMA) model, a Support Vector Regression (SVR) model, and a machine learning model using a Long Short Term Memory (LSTM) network. Each method is a form of a regression problem in which historical space elasticity is considered as an observation indexed by time, and the historical space elasticity of each category is observed as a time series. The input vector is a moving window $\beta_{i,t}$ = $[\beta_{i,t-n+1}$, $\beta_{t-n+2}, \dots$ $\beta_{i,t-1}]$ where n is number of the recent weekly space elasticity for i category, and the size of n used in this study is $n = 10$. This is followed by updating the input vector using the latest predicted space elasticity $\hat{B}_{i,t+1}$. A comparison between the methods is presented to select the most promising approach that generates space elasticities with a lower Mean Absolute Percentage Error (MAPE). Later the forecasted space elasticity $\hat{\beta}_{i,t+1}$ is used in the category SSAP mathematical model to provide optimised shelf space for the retailer to increase their total profit. Figure 5.5 illustrates a schematic description of the proposed forecasting technique.

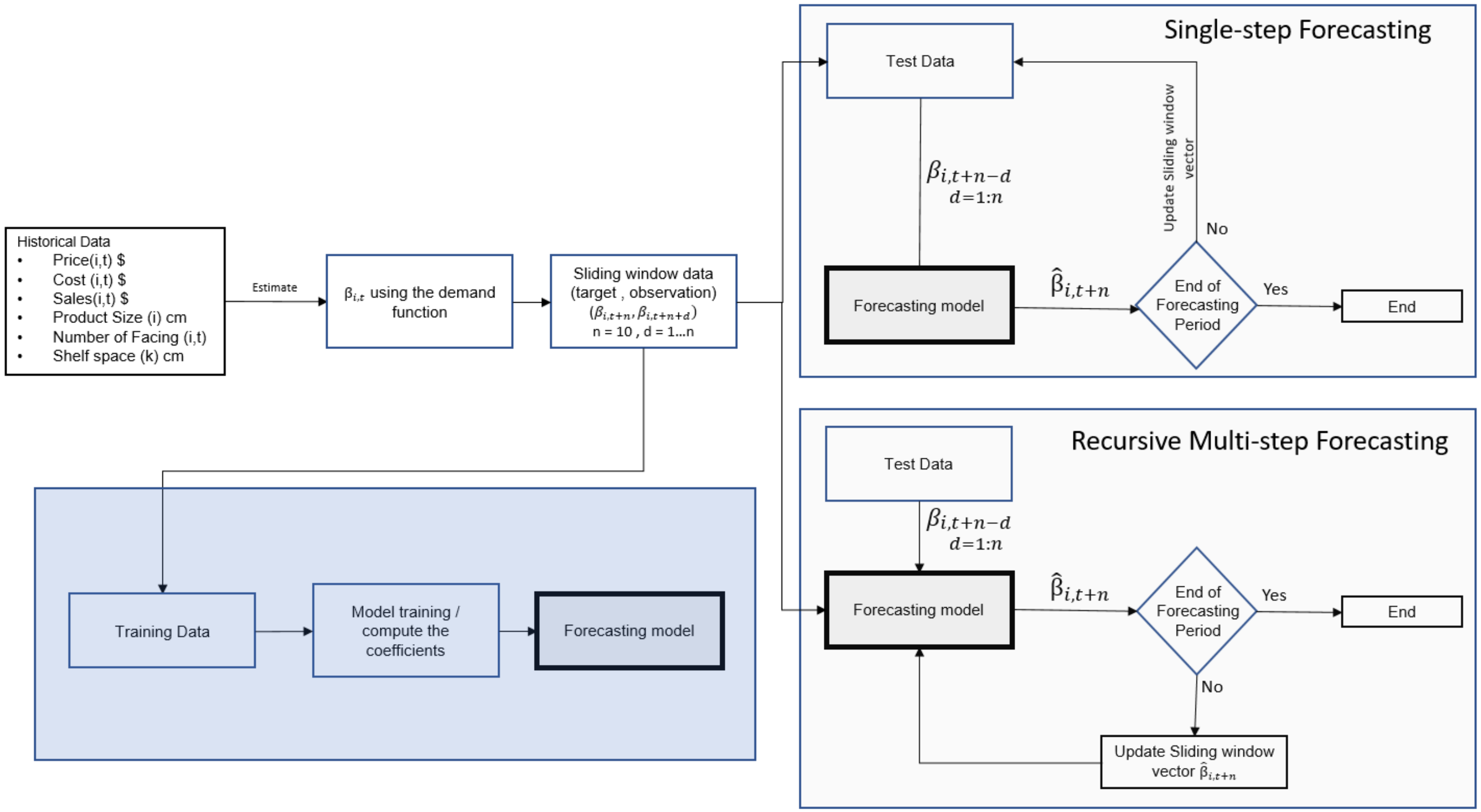


Figure 5.5: A schematic description of the proposed forecasting technique

5.5.1 Linear Regression

Linear regression is considered a classical type of regression analysis. It is an approach to model the relationship between two or more variables linearly to fit a mathematical function describing how the variation in the predictors changes the response's value. Based on the number of input variables, the regression could be either simple or multiple; for a single variable input, the method is known as simple linear regression, and for multiple variable inputs, the method is known as multiple linear regression. This study considers a single linear regression by examining a set of training data and tuning its parameters to ensure that the model accurately predicts new untrained data while minimising the absolute fitting error. The indices, parameters, and variables associated are stated as the following:

Indices

i	Categories
n	Sliding window size
t	Time periods

Parameters

β_{it}	Space elasticity of category i in time t

Variables

a_{in}	Linear regression coefficient
$\hat{\delta}_{it}$	Absolute error $\lvert\hat{\beta}_{it} - \beta_{it}\rvert$

Positive Variables

$\hat{\beta}_{it}$	Predicted Space elasticity of category i in time t

The objective is to minimise the summation of the absolute error presented here by the $\hat{\delta}_{it}$ which is expressed as $\lvert\hat{\beta}_{it} - \beta_{it}\rvert$ as presented in equations 5.5-5.8

$$Min \sum_{it} \hat{\delta}_{it} \tag{5.5}$$

$$\hat{\delta}_{it} \geq \hat{\beta}_{it} - \beta_{it} \qquad \forall \quad i = 1, ...N, \quad t \geq n \tag{5.6}$$

$$\hat{\delta}_{it} \geq \beta_{it} - \hat{\beta}_{it} \qquad \forall \quad i = 1, ...N, \quad t \geq n \tag{5.7}$$

$$\hat{\beta}_{it} = a_{in}\beta_{i,t-n} \qquad \forall \quad i = 1, ...N, \quad t \geq n \tag{5.8}$$

5.5.2 Autoregressive Integrated Moving Average (ARIMA)

Box and Jenkins developed the ARIMA model in 1970, and it is also known as the Box-Jenkins technique, which consists of a collection of practises for identifying, estimating, and diagnosing ARIMA models with time-series data. In financial forecasting, the model is the most commonly used tool (Merh et al., 2010; Nochai and Nochai, 2006; Pai and Lin, 2005). ARIMA models have outperformed complex structural models in short-term prediction, demonstrating an effective capability to produce short-term forecasts (Meyler et al., 1998). ARIMA stands for Autoregressive models, while MA stands for Moving Average models. The I in ARIMA denotes the number of lags used in data differencing. The concept behind autoregressive models is that the current value of a time series can be explained as a function of p previous values. Appendix C.0.1 demonstrates an example of an autoregressive model of order p.

To build an ARIMA forecast model, there are three main steps to take. The first step is the model specification, which entails determining three numbers, p, d, and q, that define the ARIMA model. The value of d is first calculated by determining the data's stationarity, d=0 when the data is stationary. If the data is trendy, we take the first difference and double-check for stationarity. We keep taking differences until we reach a point where the output is stationary. The number d represents the requisite number of differences. Stationarity checks are used to assess whether or not a sequence is stationary. Statistical hypothesis tests, such as the unit root test, are one way to decide more scientifically whether differencing is needed. The Dickey-Fuller test and its augmented variant, the augmented Dickey-Fuller test (ADF), were used in our analysis; for more information on ADF, see Appendix C.0.2.

There are several methods for deciding the best ARIMA order for a given time series. Examining the autocorrelation function (ACF) and partial autocorrelation (PACF), for example, will assist in deciding the number of AR and/or MA terms required. This can be accomplished by looking into the lag after which the autocorrelation or partial autocorrelation becomes zero or approaches zero, indicating the values of q and p, respectively. Another approach requires manually defining a grid of p and q ARIMA parameters to iterate for ARIMA model tuning for a one-step rolling prediction. For each parameter, a model is developed, and its output is measured using the MAPE function.

The autoARIMA function in EViews was used to derive an initial estimation in this study; EViews is a computational package for Windows that is primarily used for time-series-based econometric analysis. A small grid of p and q around the original estimates was used in the grid search. A prediction is made, and its accuracy is tested at each iteration. After determining the ARIMA model's order, the model parameters must be estimated. The maximum likelihood estimation (MLE) technique is used to find the parameter values that increase the likelihood of obtaining the observed data. It is essential to highlight that, in the grid search approach mentioned above, the ARIMA parameters were estimated for each iteration, and its performance is evaluated using the test data.

5.5.3 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a regression-specific adaptation of SVM (Support Vector Machine). It is a common regression to solve problems in a non-linear form (Vapnik, 2013). A kernel function maps the input data into a higher-dimensional space without increasing the computational cost. Theoretically, a linear function exists in the high-dimensional feature space to build a relationship between the input mapped features and the output data.

An optimisation problem is needed to solve the function by minimising the coefficients, finding the narrowest tube centered around the surface while minimising the regression error ε, i.e., the difference between the predicted and ideal outputs and maximising the margin of the hyperplane. Figure 5.6 shows an example plot of the SVR.

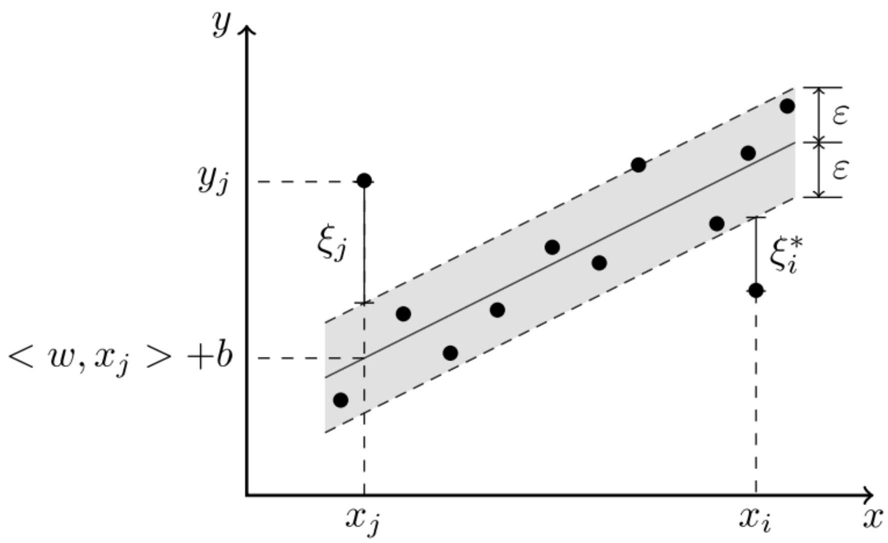


Figure 5.6: An example plot of a support vector regression (Moustapha et al., 2018)

Selecting the SVR parameters C and ε is one of the most important factors in SVR accuracy. For more details about the SVR please see Appendix C.0.3. In this study MATLAB Bayesian optimisation (Snoek et al., 2012) is used to extract the optimal SVR hyper-parameters.

5.5.4 Long Short-Term Memory (LSTM) Networks

Staudemeyer and Morris (2019) proposed the Long Short Term Memory (LSTM) network, where it is a form of artificial Recurrent Neural Networks (RNN), see Appendix C.0.4 for more information about the RNN networks. In RNN, ”Recurrent” refers to the fact that the network’s input and output are looped. At each time step, the network’s output is copied and returned to the network as input in the next step.

The LSTM network is designed to learn temporal data, time-stamped sequences of data, and their long-term dependencies more accurately than recurrent neural networks (RNN); this is helpful to avoid the problem of gradient disappearance and explosion in RNN.

The LSTM network is widely used in many applications such as stock market prediction, handwriting recognition, speech recognition, natural language processing, and others and for more details about LSTM see (Hochreiter and Schmidhuber, 1997; Staudemeyer and Morris, 2019). According to the LSTM applications, the architecture varies, from many-to-one model (sequence to one), many-to-many model (sequence to sequence), and several other variations. Figure 5.7 shows the architecture for the basic LSTM Network model and LSTM neuron structure, where the LSTM contains a memory cell structure and three types of gates input, forget, and output gate. The working procedure of the LSTM cell is defined mathematically and can be found in Appendix C.0.5.

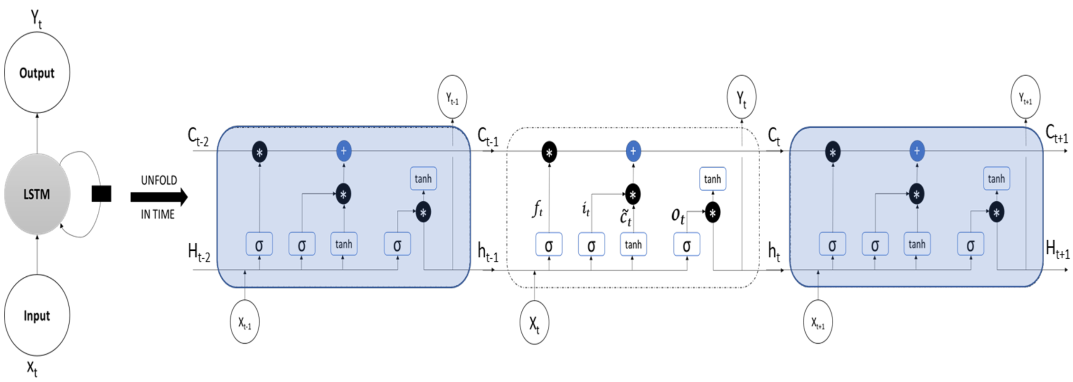


Figure 5.7: The structure of LSTM

In this study: a forecasting model is built with two layers of stacked LSTM before being forwarded to a dropout layer; a layer to prevent over-fitting, with 128 and 64 neurons for the first and second LSTM layer, respectively and 100 epochs and a dropout rate of 0.4. Then, a fully connected dense layer is then connected to the regression layer to predict the output. The stacked LSTM is a more powerful way to extract features and maximise model capacity than the conventional LSTM (Yu et al., 2019). Matlab software (MATLAB, 2018) with the optimisation algorithm (adam) (Kingma and Ba, 2014) were used to train the model with an initial learning rate of 0.005.

To train the LSTM model the historical space elasticity $\beta_{i,t}$ data is used. The data was randomly partitioned into a 70% training set and a 30% testing set. Historical data is converted to a time series data set of observation and target to prepare the input data. The observation set is a moving window of $\beta_{i,t+n-d}$ and the corresponding target is $\beta_{i,t+n}$, where n is the sliding window size, set to $n = 10$ and d is the length of the historical data, $d = 1 \ldots n$.

5.6 Data

In this study, cross-sectional data were collected from a local store in Jeddah, Saudi Arabia. The data covers a period of 260 weeks from 2014 to 2018. It contains a weekly space allocation for ten different categories along the Cereal Aisle. In addition, the sales and number of facing per category were included in the data, which vary from one week to the other.

Table 5.1: Average and standard deviations of the Data

Category	Facing (item)		Space (cm)		Sales (item)		Price ($)		cost ($)	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
1	69	5	1073.61	83.46	147	37	19.27	0.52	13.40	0.03
2	79	1	672.32	12.57	108	5	24.23	1.09	20.60	0.92
3	74	4	764.30	46.21	369	78	34.75	2.23	26.53	1.90
4	27	6	647.63	136.62	27	4	36.37	2.17	19.02	1.78
5	78	1	570.50	6.75	233	10	18.30	0.67	16.34	0.59
6	22	3	314.46	47.27	54	5	10.39	0.36	8.87	0.31
7	15	6	160.05	65.81	27	6	39.43	1.92	35.24	1.72
8	75	3	716.74	26.74	318	46	16.02	0.15	13.88	0.15
9	76	3	756.46	32.96	205	27	18.54	1.93	15.49	1.63
10	24	6	323.42	75.03	31	3	29.13	1.89	24.36	1.52

Table 5.1 represents the mean and standard deviation of data. The total shelf space available for all categories was set to 60 meters based on managerial agreements in the store. The average size for each category was calculated using the space allocated divided by the number of facing. The complete data is presented in the supplementary material attached to this thesis.

We have followed the ordinary least squares regression (OLS) method presented in (Van Dijk et al., 2004) to estimate the scaling parameter α_i afterward the space elasticities for each time period $\beta_{i,t}$ were calculated using the following demand function:

$$D_{it} = \alpha_i (X_{it})^{\beta_{it}} \qquad \forall \quad i,t \tag{5.9}$$

Where:

D_{it} = demand for category i at time t

α_i = scaling parameter for category i

X_{it} = number of facing for each category i at time t

β_{it} = space elasticity for category i at time t

As stated above, one of the main features of the Demand function is its ability to be transformed by logarithmic transformation to a linear function; ie. taking the log of both sides. Therefore, the demand function of equation 5.9, is transformed as the following:

$$ln\, D_{it} = ln\, \alpha_i + \hat{\beta}_i\, ln\, X_{it} \qquad \forall \quad i,t \tag{5.10}$$

where, $\hat{\beta}_i$ is the approximate average value of β_{it}. We apply the OLS regression to equation 5.10 to estimate α_i and $\hat{\beta}_i$, where only the values of α_i are kept and the following equation is used to find the space elasticity $\beta_{i,t}$ over time t:

$$\beta_{it} = ln[\frac{D_{it}}{\alpha_i}] \cdot \frac{1}{lnX_{it}} \qquad \forall \quad i,t \tag{5.11}$$

5.7 Results and Discussions

Preparing the data was carried by randomly partitioned into a 70% training set and a 30% testing set to avoid the re-substitution error. All models were given the testing and training data for a length of 4 years, i.e. (208 weeks) from 2014 to 2017. Then, it was converted to a time-series data set of observation and target. The observation set is a sliding window[$\beta_{i,t}$, $\beta_{i,t+1}$,... $\beta_{i,t+n}$] where n is the most recent weekly space elasticity for the i category, the size of the sliding window was set to $n = 10$.

The forecasting of the space elasticity was carried using different Time Series models, where the models were to forecast the year of 2018. The prediction was then compared with the actual data. The performances were measured for both single-step and Recursive Multiple-step ahead methods. In the single-step, the sliding window was updated each time using the historical data real value. However, in the Recursive Multiple-step, the current time step's predicted space elasticity $\hat{\beta}_{i,t+1}$ value is used to update the sliding window. Two well-used performance measures from the literature were used to compare the performance of both models: the Mean Square Error (MSE) and the Mean Absolute Percentage Error (MAPE) as in equations 5.12 and 5.13 respectively.

$$RMSE_i = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(\beta_{ti} - \hat{\beta}_{ti})^2} \tag{5.12}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|(\frac{\beta_{ti} - \hat{\beta}_{ti})}{\beta_{ti}}| \tag{5.13}$$

It should be noted that all models have been performed on an Intel®Xeon®E5-1620 CPU with16GB RAM. The Linear Regression and SSAP model were solved using GAMS modelling system with CPLEX Mixed Integer Linear Programming (MILP) optimisation package. For the ARIMA model, the autoARIMA function in EViews, a computational package for Windows primarily used for time-series-based econometric analysis, was used to derive an initial estimation, then MATLAB software was used for training and forecasting. For the SVR model, MATLAB Bayesian optimisation was used to extract the optimal hyper-parameters, train, and forecast. Finally, for the LSTM model, MATLAB software was used with (adam) optimisation algorithm for training and forecasting.

5.7.1 Forecasting Results

Table 5.2 summarises the results for all forecasting models using a single-step-ahead method. The results; indicate that the SVR outperforms the other methods when using a single-step ahead method, with a MAPE of 1.08±0.78. The forecasting is considered very good, and the results comparing the actual data with the forecasted data can be seen in figure 5.8 for the SVR. It could be seen from 5.8 that the forecasting values follow a very similar trend to the the historical data for all 10 Categories. The ARIMA model also provides reasonable MAPE values of 1.27±1.5. Typically SVR and ARIMA models outperform other Time series models when used on single-step ahead forecasting strategies due to their ability in predicting near-future data. LR had the highest MAPE values of 2.73±2.07, followed by the LSTM model with a MAPE of 1.72±1.73.

Whereas for the Recursive Multiple-step as in table 5.3, the results show that the LSTM generates better results compared with all other methods, with a MAPE of 2.03±1.02, proving that LSTM models have an advantage when forecasting using a multi-step ahead strategy. The results comparison of the actual data and the forecasted data for the LSTM can be seen in figure 5.9. The results show that the LSTM can follow the historical data values with an acceptable percentage of error for most of the Categories. The ARIMA model has also provided good prediction with a MAPE of 3.23±3.55, followed by LR and SVR with a MAPE of 5.49±4.49 and 6.05±5.80, respectively.

Table 5.2: Comparison for single-step Forecasting Strategy

Category	LR		ARIMA		SVR		LSTM	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
1	0.01	2.03	0.01	1.17	0.01	0.92	0.02	1.55
2	0.01	1.70	0.00	0.21	0.00	0.54	0.00	0.04
3	0.01	0.65	0.01	1.08	0.01	0.87	0.02	1.57
4	0.02	5.53	0.01	1.62	0.01	2.05	0.00	0.06
5	0.01	1.07	0.00	0.33	0.00	0.44	0.01	1.26
6	0.02	3.94	0.01	0.94	0.01	0.94	0.01	1.27
7	0.01	3.72	0.02	5.60	0.02	3.01	0.02	6.59
8	0.01	0.39	0.01	0.48	0.01	0.42	0.02	1.42
9	0.01	1.42	0.01	0.63	0.01	0.74	0.01	1.88
10	0.02	6.84	0.00	0.65	0.00	0.89	0.01	1.52
Mean	0.01	2.73	0.01	1.27	0.01	1.08	0.01	1.72
STD	0.00	2.07	0.01	1.50	0.00	0.78	0.01	1.73

Table 5.3: Comparison for multi-step Forecasting Strategy

Category	LR		ARIMA		SVR		LSTM	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
1	0.10	17.77	0.02	2.52	0.06	9.10	0.03	3.24
2	0.01	2.25	0.01	0.99	0.01	1.43	0.00	0.09
3	0.06	7.24	0.02	1.95	0.05	5.48	0.02	2.36
4	0.02	5.68	0.01	2.83	0.03	10.69	0.00	1.33
5	0.02	1.81	0.01	0.98	0.01	1.27	0.01	0.91
6	0.02	4.46	0.02	3.63	0.02	3.46	0.01	2.52
7	0.01	3.14	0.04	13.60	0.06	20.89	0.02	3.07
8	0.02	2.41	0.02	1.80	0.02	2.56	0.02	2.05
9	0.02	3.18	0.02	2.61	0.02	3.16	0.03	3.01
10	0.02	6.95	0.01	1.43	0.01	2.43	0.01	1.77
Mean	0.03	5.49	0.02	3.23	0.03	6.05	0.02	2.03
STD	0.03	4.49	0.01	3.55	0.02	5.80	0.01	1.02

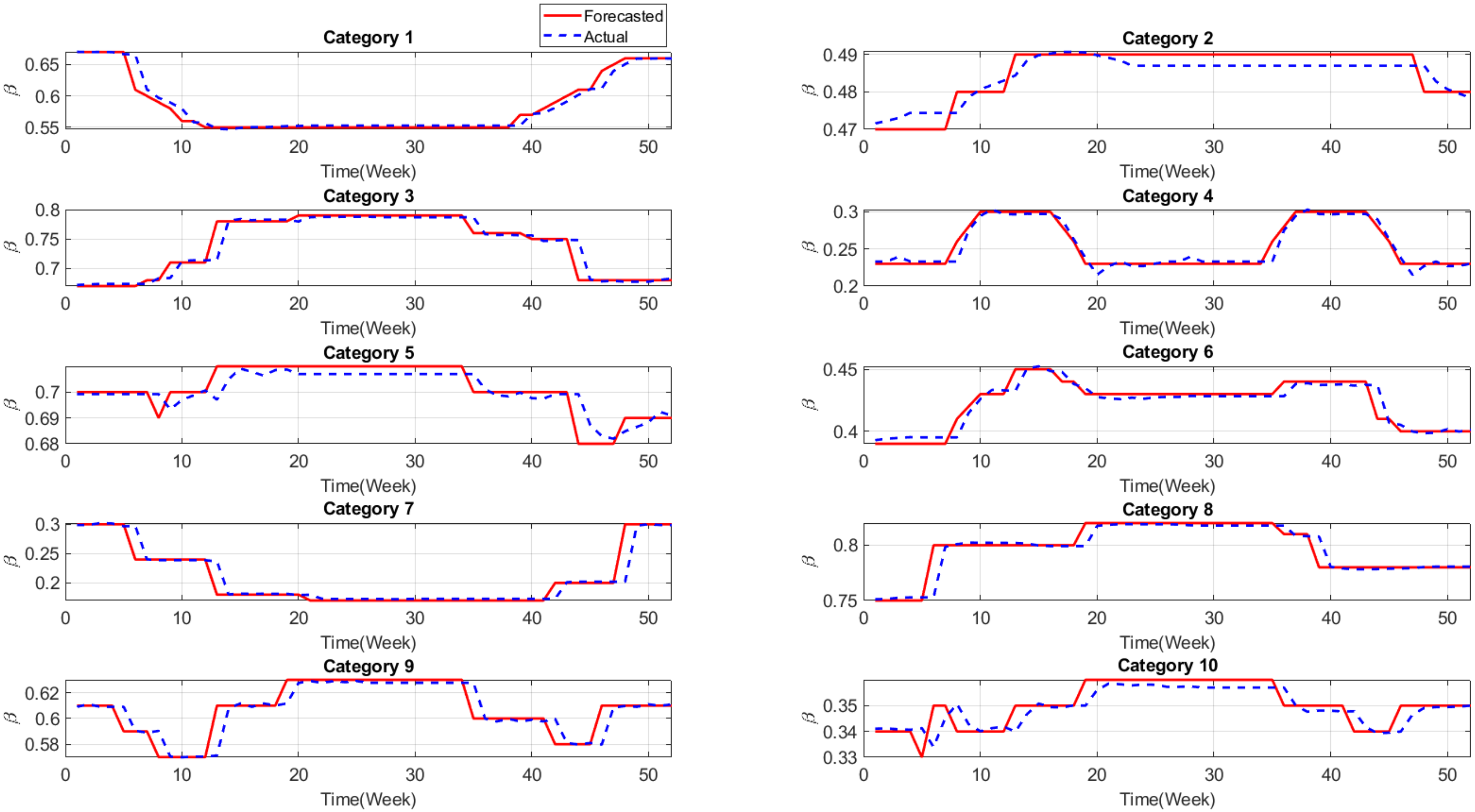


Figure 5.8: Comparison between actual and forecasted results using SVR for the single-step forecasting strategy

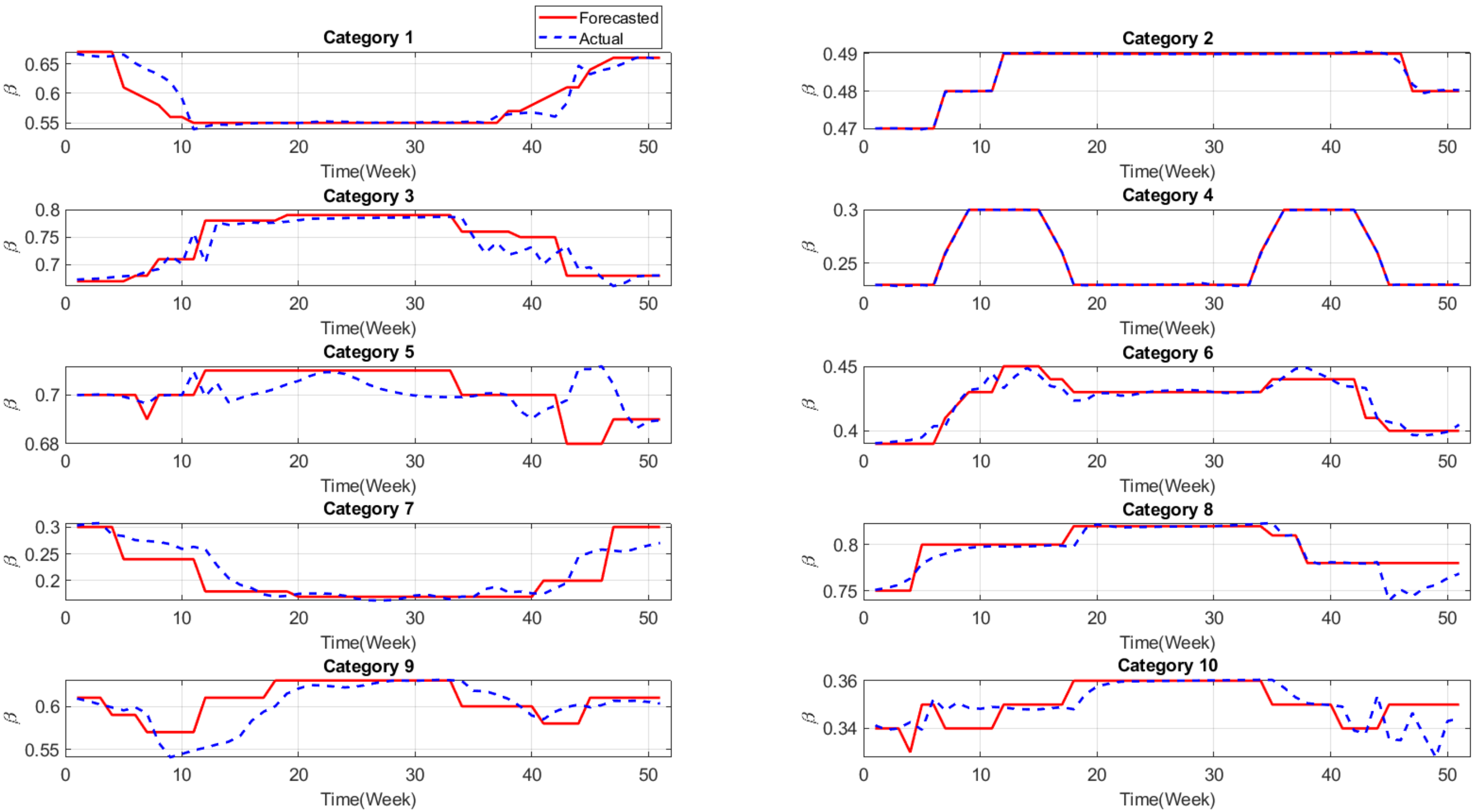


Figure 5.9: Comparison between actual and forecasted results using LSTM for the multi-step forecasting strategy

5.7.2 SSAP Results

From the results above, the LSTM outperforms the other Time series models when using the multi-step ahead forecasting strategy. So the forecasted 52 weeks space elasticity $\hat{\beta}_{i,t}$ using the LSTM multi-step ahead model was used as an input parameter to the SSAP traditional model to compute the number of facing allocated for each category, taking into account the data given. The SSAP model presented in Section 4 for equations 5.1-5.4 and the comparison between the historical data and the proposed SSAP model based on the estimated $\hat{\beta}_{i,t}$ are presented in figure 5.10:

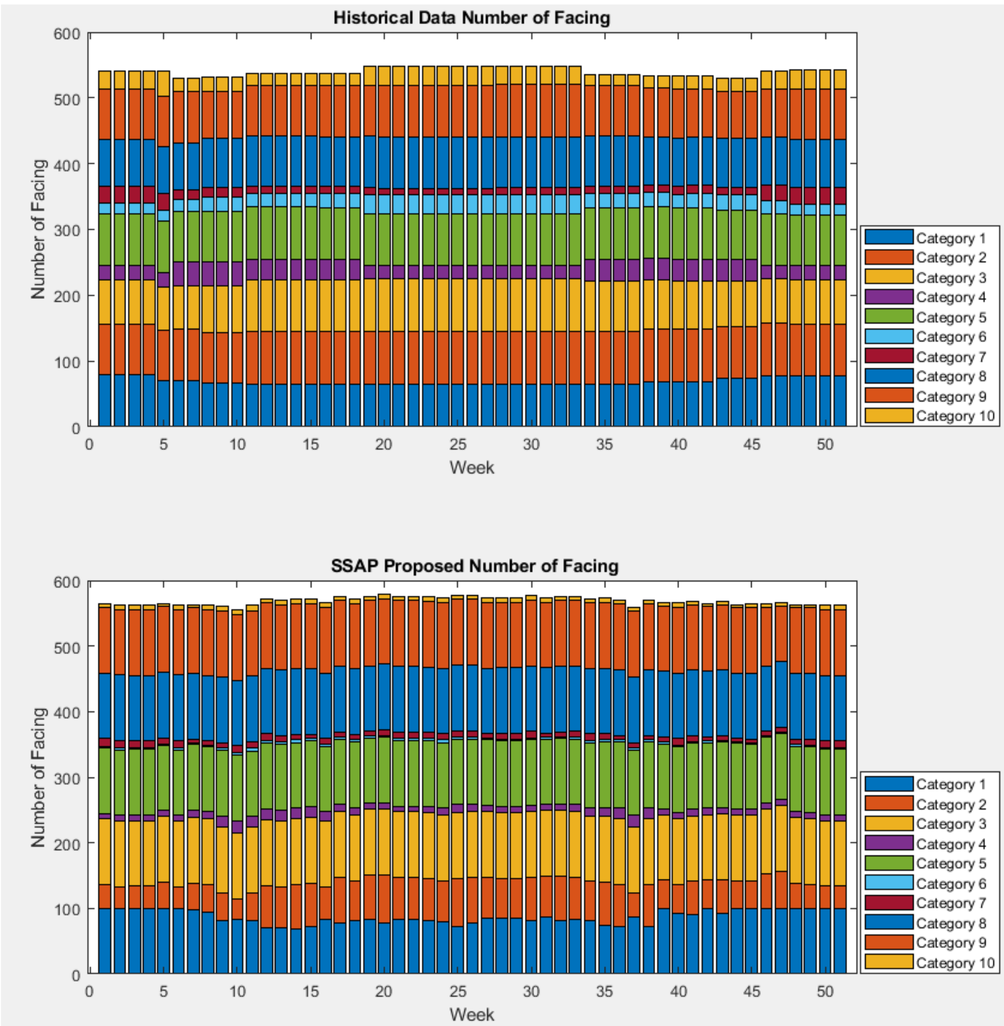


Figure 5.10: Comparison between the Number of Facing from Historical Data and the Proposed SSAP

As it can be seen from the above figure, figure 5.10, the number of facing for the categories have changed when using the new estimated $\hat{\beta}_{i,t}$ and combining it with the optimisation model the average number of items on the shelves has increased by approximately 5.3% which has a prediction of around 12% increase in Sales compared with historical data.

5.8 Conclusions

The SSAP is a representative of shelf-space management, and it has a direct positive effect on sales by managing the distribution of items across the shelves. Therefore, consumers are more likely to purchase items if a commodity is given a wide shelf space, but the amount of shelf space given affects the purchasing decisions up to a certain level, i.e., the relationship is non-linear. Conventional research on shelf-space allocation problems tries to formulate the problem using stationary demand and fixed space elasticities based on previous observations. In the approach proposed in this chapter, a dynamic framework using Time series linear methods; Linear Regression (LR), Auto-regressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models are proposed to forecast the space elasticity utilising historical data for ten categories over a period of 5 years. By comparing the approaches' outcome, the SVR in the single-step ahead model outperforms other methods with a MAPE error of 1.08±0.78 compared to 1.27±1.50, 1.72±1.73, and 2.73±2.07 MAPE errors for the ARIMA, LSTM, and Linear regression, respectively. While for the recursive multi-step ahead model, the LSTM network outperforms all other methods. On average, the MAPE errors for the LSTM model are 2.03±1.02 compared to 3.23±3.55, 5.49±4.49, and 6.05±5.80 for the ARIMA model, linear regression, and SVR, respectively. To further verify the model's completeness, the estimated space elasticities were used in the traditional SSAP model to compare the number of facing and sales against the historical data. The results show an increase of about 5.3% in the number of products displayed on the shelf which has a sales increase prediction of around 12%.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, a summary of the problems presented in this thesis is given, noting each problem's main contribution to the Supply Chain. Furthermore, future research extensions and recommendations are noted.

6.1 Summary

In this research thesis, Supply Chain real-life application problems have been investigated; the Manufacturer's Pallet Loading Problem (MPLP), the Design of Assortments Problem, and Shelf Space Allocation Problems (SSAP), where these problems heavily arise in daily activities of manufacturing, warehousing, logistics, and retailing. Therefore optimisation based, decision-making frameworks are necessary to capture such problems and offer practical solutions. All the problems presented share a similar structure in having two types of elements; small known as items and big known as pallets, bins, or shelves. Where the main objective is to maximise the number of items loaded or stacked onto the pallets, bins, or shelves to maximise the utilisation percentage or potential sales.

In the area related to the pallet loading problem, novel Mixed Integer Linear

Programming mathematical models have been implemented to represent the problem; a single-objective model and a multi-objective model. Both models look into the problem from a Manufacturers point of view, packing homogeneous products into predefined pallet sizes. The model was implemented to occupy problems in the 2-Dimensional space, ie. allowing items to rotate and to be grouped into blocks of a single orientation, generating optimal layouts of improved structures. A comprehensive comparison between the two models was provided, supported, and tested by several data sets from the literature. Both models have provided promising results surpassing layouts presented in the literature in terms of complexity. The main differences arise in how the complexity index ζ is defined and calculated, where in the single-objective model, it is a manual process used for comparison only. Whereas in the multi-objective model, the complexity index is integrated into the mathematical model, providing a more robust model. Another advantage of the multi-objective model is the computational time required to solve the data sets, where an improvement of over 100 % has been seen. Such improvement is necessary for industrial applications, where fast and robust solutions are required.

In the area of Cutting and Packing (CP), a sub-problem known as Design of Assortment has been investigated. The problem statement has been created for the 11th AIMMS-MOPTA Optimisation Modelling Competition. The main objective was to provide optimal packing from a set of given items and decide which items to pack. A Mixed Integer Linear Programming algorithm has been developed to provide a diverse set of solutions. Furthermore, a basic software prototype based on the AIMMS platform has been built with a user-friendly interface to ease user engagement with a visual representation of the obtained solutions. The model effectiveness has been tested using a set of data obtained by the competition committee, and the results show that the proposed algorithm is efficient in handling the problem.

The final problem considered is the Shelf Space Allocation Problems (SSAP), where it concerns retail businesses, from small shops, supermarkets to departmental stores. A dynamic framework has been proposed to forecast space elasticities, the relationship between the demand and the shelf space allocated to each item. The forecasting was based on historical data using multiple standard time-series methodologies. Linear Regression (LR), Support Vector Regression (SVR), Autoregressive Integrated Moving Average (ARIMA), and Deep learning Long Short Term Memory (LSTM) networks with single and recursive multi-step ahead models have been developed. A comparison between all models has been presented, indicating the most effective approach based on real store data. In addition, an optimisation mathematical model has been implemented using the forecasted space elasticities to provide the retailer with optimal shelf space, thus resulting in a closer match between the supply and demand, leading to increased profitability.

6.2 Recommendations For Future Work

The work in this thesis has covered several problems in the area of space allocation, and there are still several research directions for future work as the extension of the current study.

One of the main drawbacks of supply chain research is the lack of communication between the research community and the industrial world. Where the research focuses on the actual problems presented in the industry but without tackling key practical features, from this proposition, this research took place, where for the Manufacturer's Pallet Loading Problem (MPLP), the mathematical models developed have shown to produce optimal solutions but with layouts of reduced complexity and solved within reasonable computational times. Pallet loading complexity is a significant aspect when hundred or maybe thousands of pallets are packed every day at manufacturers/ plants warehouses, but previous research has not tackled this point. Such reduction in complexity is directly associated with pallet loading times, whether manual or robotic loading occurs, as it will affect the time required to construct each pallet. A case study of actual loading times could be made, reporting the percentage of time and cost reduction when using simplified layouts compared to the layouts presented in the literature to further justify these claims.

Another extension to the problem could be incorporating additional features/-constraints to the problem, such as cargo weight, stability of loading, or weight distribution. It may also be worth looking at different pallet loading problems such as the Distributor's Pallet Loading Problem (DPLP), where heterogeneous packing is considered and apply the complexity concept. Also, exploring other effective solution strategies such as combining heuristics and mathematical programming is another way of tackling the problem, especially if computational time reduction is desired.

For the problem of assortments, the proposed mathematical model effectively provides a diverse set of solutions for the user based on a set of heterogeneous items and homogeneous pallets. An extension to the problem could be using heterogeneous sets of items and pallets, where the model would find and allocate the best combinations of items and pallets. Another challenging area would be to explore irregular item assortment problems, where the set of items can be in different shapes such as cones, cylinders, pentagons, or hexagons. Many other extensions could be applicable depending on the research area and problem at hand, such as extending the problem to a 2-D model rather than the current 1-D model. Furthermore, a multi-objective model could be considered, where the number of items loaded and the utilisation percentage is maximised, and the spaces between the items are minimised.

Finally, for the last problem explored and discussed in this thesis, a more detailed approach of including categories and sub-categories could be considered by looking into cross-space elasticises; the effect complementary or substitute products may have on the shelf space allocation. Also, another interesting approach would be using an artificial intelligence algorithm that can combine both the estimation of space elasticises and the prediction of shelf space allocation; however, building such an algorithm may require larger historical data. As discussed earlier, an alignment between the research and industry is essential, so another exciting approach would be combining mathematical models into available planogram software's. Furthermore, the replenishment and back storage cycles could be considered when designing the shelf space allocation model.

References

Aljuhani, D. M. and Papageorgiou, L. G. (2021). Improved layout structure with complexity measures for the manufacturer's pallet loading problem (mplp) using a block approach. *Journal of Industrial Engineering and Management*, 14(2):231–249.

Alvarez-Valdés, R., Parreño, F., and Tamarit, J. M. (2005a). A branch-and-cut algorithm for the pallet loading problem. *Computers & Operations Research*, 32(11):3007–3029.

Alvarez-Valdés, R., Parreño, F., and Tamarit, J. M. (2005b). A tabu search algorithm for the pallet loading problem. *Or Spectrum*, 27(1):43–61.

Amaral, A. R. and Wright, M. (2001). Experiments with a strategic oscillation algorithm for the pallet loading problem. *International Journal of Production Research*, 39(11):2341–2351.

Arenales, M. and Morabito, R. (1995). An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, 84(3):599–617.

Arghavani, J. and Abdou, G. (1996). 3D volumetric pallet-loading optimisation. *The International Journal of Advanced Manufacturing Technology*, 11(6):425–429.

Bai, R. and Kendall, G. (2008). A model for fresh produce shelf-space allocation and inventory management with freshness-condition-dependent demand. *INFORMS Journal on Computing*, 20(1):78–85.

Baker, R. A. and Urban, T. L. (1988). A deterministic inventory system with an inventory-level-dependent demand rate. *Journal of the Operational Research

Society, 39(9):823–831.

Beasley, J. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations research*, 33(1):49–64.

Bechtel, C. and Jayaram, J. (1997). Supply chain management: a strategic perspective. *The international journal of logistics management*, 8(1):15–34.

Belov, G., Kartak, V., Rohling, H., and Scheithauer, G. (2009). One-dimensional relaxations and lp bounds for orthogonal packing. *International Transactions in Operational Research*, 16(6):745–766.

Bhattacharya, S., Roy, R., and Bhattacharya, S. (1998). An exact depth-first algorithm for the pallet loading problem. *European Journal of Operational Research*, 110(3):610–625.

Birgin, E. G., Lobato, R. D., and Morabito, R. (2010). An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. *J. Oper. Res. Soc*, 61(2):306–320.

Birgin, E. G., Morabito, R., and Nishihara, F. H. (2005). A note on an l-approach for solving the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 56(12):1448–1451.

Bischoff, E. and Dowsland, W. (1982). An application of the micro to product design and distribution. *Journal of the Operational Research Society*, 33(3):271–280.

Bookbinder, J. H. and Zarour, F. H. (2001). Direct product profitability and retail shelf-space allocation models. *Journal of Business Logistics*, 22(2):183–208.

Borin, N., Farris, P. W., and Freeland, J. R. (1994). A model for determining retail product category assortment and shelf space allocation. *Decision sciences*, 25(3):359–384.

Bultez, A. and Naert, P. (1988). Sh. arp: Shelf allocation for retailers' profit. *Marketing science*, 7(3):211–231.

Chandon, P., Hutchinson, J. W., Bradlow, E. T., and Young, S. H. (2009). Does in-store marketing work? effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase. *Journal of marketing*, 73(6):1–17.

Chankong, V. and Haimes, Y. Y. (2008). *Multiobjective decision making: theory and methodology*. Courier Dover Publications.

Coffman, E. G., Garey, M. R., and Johnson, D. S. (1984). *Approximation Algorithms for Bin-Packing — An Updated Survey*, pages 49–106. Springer Vienna, Vienna.

Cooper, M. C., Lambert, D. M., and Pagh, J. D. (1997). Supply chain management: more than a new name for logistics. *The international journal of logistics management*, 8(1):1–14.

Corstjens, M. and Doyle, P. (1981). A model for optimizing retail space allocations. *Management Science*, 27(7):822–833.

Corstjens, M. and Doyle, P. (1983). A dynamic model for strategically allocating retail space. *Journal of the Operational Research Society*, 34(10):943–951.

Curhan, R. C. (1972). The relationship between shelf space and unit sales in supermarkets. *Journal of Marketing Research*, 9(4):406–412.

De Carvalho, J. V. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, 86:629–659.

Degraeve, Z. and Peeters, M. (2003). Optimal integer solutions to industrial cutting-stock problems: Part 2, benchmark results. *INFORMS Journal on Computing*, 15(1):58–81.

Desmet, P. and Renaudin, V. (1998). Estimation of product category sales responsiveness to allocated shelf space. *International Journal of Research in Marketing*, 15(5):443–457.

Dowsland, K. A. (1984). The three-dimensional pallet chart: an analysis of the factors affecting the set of feasible layouts for a class of two-dimensional packing problems. *Journal of the Operational Research Society*, 35(10):895–905.

Dowsland, K. A. (1987a). An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, 31(1):78–84.

Dowsland, K. A. (1987b). A combined data-base and algorithmic approach to the pallet-loading problem. *Journal of the Operational Research Society*, 38(4):341–345.

Dowsland, K. A. and Dowsland, W. B. (1992). Packing problems. *European journal of operational research*, 56(1):2–14.

Dreze, X., Hoch, S. J., and Purk, M. E. (1994). Shelf management and space elasticity. *Journal of retailing*, 70(4):301–326.

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

Egeblad, J. and Pisinger, D. (2009). Heuristic approaches for the two- and three-dimensional knapsack packing problem. *Computers & Operations Research*, 36(4):1026–1049.

Eisend, M. (2014). Shelf space elasticity: A meta-analysis. *Journal of Retailing*, 90(2):168–181.

Farago, R. and Morabito, R. (2000). Um método heurístico baseado em relaxação lagrangiana para o problema de carregamento de paletes do produtor. *Pesquisa Operacional*, 20(2):197–212.

Fekete, S. P. and Schepers, J. (2001). New classes of fast lower bounds for bin packing problems. *Mathematical programming*, 91(1):11–31.

Galai, D., Hillel, L., and Wiener, D. (2016). *How to create a successful business plan: for entrepreneurs, scientists, managers and students*. World Scientific.

Gilmore, P. C. and Gomory, R. E. (1963). A linear programming approach to the cutting stock problem—part ii. *Operations research*, 11(6):863–888.

Hansen, J. M., Raut, S., and Swami, S. (2010). Retail shelf allocation: a comparative analysis of heuristic and meta-heuristic approaches. *Journal of Retailing*, 86(1):94–105.

Heinsbroek, H. (1977). De maximalisatie van de schapruimtetoewijzing in de supermarkt. *Unpublished Ph. D. thesis, Free University of Brussels, Belgium*.

Hinxman, A. (1980). The trim-loss and assortment problems: A survey. *European Journal of Operational Research*, 5(1):8–18.

Hoch, S. J., Drèze, X., and Purk, M. E. (1994). Edlp, hi-lo, and margin arithmetic. *Journal of Marketing*, 58(4):16–27.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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

Hubner, A. H. and Kuhn, H. (2012). Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. *Omega*, 40(2):199–209.

Irion, J., Lu, J.-C., Al-Khayyal, F., and Tsao, Y.-C. (2012). A piecewise linearization framework for retail shelf space management models. *European Journal of Operational Research*, 222(1):122–136.

Irion, J., Lu, J.-C., Al-Khayyal, F. A., and Tsao, Y.-C. (2011). A hierarchical decomposition approach to retail shelf space management and assortment decisions. *Journal of the Operational Research Society*, 62(10):1861–1870.

Ji, Y.-G. and Jin, G.-W. (2009). A New Exact Algorithm Using the Stair Structure for the Pallet Loading Problem. *Journal of the Korean Operations Research and Management Science Society*, 34(3):43–53.

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Keltz, H. and Sterneckert, K. (2009). The trend toward consumer–centric merchandising requires assortment management and space planning investments. *Tech. rep., AMR Research*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *https://arxiv.org/abs/1412.6980*.

Kocjan, W. and Holmström, K. (2008). Generating stable loading patterns for pallet loading problems. In *The Fifth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*.

Koul, S., Sinha, P. K., and Mishra, H. G. (2016). Retracted: Antecedents to customer dependency in buyer–seller relationship: A bop retailer investigation.

Lambert, D. M. and Cooper, M. C. (2000). Issues in supply chain management. *Industrial marketing management*, 29(1):65–83.

Lau, H. C., Chan, T., Tsui, W., Ho, G. T., and Choy, K. L. (2009). An AI approach for optimizing multi-pallet loading operations. *Expert Systems with Applications*, 36(3):4296–4312.

Leao, A. A., Toledo, F. M., Oliveira, J. F., Carravilla, M. A., and Alvarez-Valdés, R. (2020). Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803–822.

Lim, A., Ma, H., Xu, J., and Zhang, X. (2012). An iterated construction approach with dynamic prioritization for solving the container loading problems. *Expert Systems with Applications*, 39(4):4292–4305.

Lim, A., Rodrigues, B., and Zhang, X. (2004). Metaheuristics with local search techniques for retail shelf-space optimization. *Management science*, 50(1):117–131.

Lins, L., Lins, S., and Morabito, R. (2003). An L-approach for packing (l, w)-rectangles into rectangular and L-shaped pieces. *Journal of the Operational Research Society*, 54(7):777–789.

Lins L, Lins S, . M. R. (2003). An L-approach for packing (l, w)-rectangles into rectangular and L-shaped pieces. *Journal of the Operational Research Society, 54, 777–789*.

Lodi, A., Martello, S., and Vigo, D. (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11(4):345–357.

Marcotte, O. (1986). An instance of the cutting stock problem for which the rounding property does not hold. *Operations Research Letters*, 4(5):239–243.

Martello, S. and Toth, P. (1990). An exact algorithm for large unbounded knapsack problems. *Operations research letters*, 9(1):15–20.

Martins, G. H. and Dell, R. F. (2007). The minimum size instance of a pallet loading problem equivalence class. *European Journal of Operational Research*, 179(1):17–26.

Martins, G. H. and Dell, R. F. (2008). Solving the pallet loading problem. *European Journal of Operational Research*, 184(2):429–440.

MATLAB (2018). *9.7.0.1190202 (R2019b)*. The MathWorks Inc., Natick, Massachusetts.

Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., and Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business logistics*, 22(2):1–25.

Merh, N., Saxena, V. P., and Pardasani, K. R. (2010). A comparison between hybrid approaches of ann and arima for indian stock trend forecasting. *Business Intelligence Journal*, 3(2):23–43.

Meyler, A., Kenny, G., and Quinn, T. (1998). Forecasting irish inflation using arima models.

Min, S., Zacharia, Z. G., and Smith, C. D. (2019). Defining supply chain management: in the past, present, and future. *Journal of Business Logistics*, 40(1):44–55.

Morabito, R. and Morales, S. (1998). A simple and effective recursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 49(8):819–828.

Moustapha, M., Bourinet, J.-M., Guillaume, B., and Sudret, B. (2018). Comparative study of kriging and support vector regression for structural engineering applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 4(2):04018005.

Murray, C. C., Talukdar, D., and Gosavi, A. (2010). Joint optimization of product price, display orientation and shelf-space allocation in retail category management. *Journal of Retailing*, 86(2):125 – 136.

Neliβen, J. (1995). How to use structural constraints to compute an upper bound for the pallet loading problem. *European Journal of Operational Research*, 84(3):662–680.

Nochai, R. and Nochai, T. (2006). Arima model for forecasting oil palm price. In

Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and applications, pages 13–15.

Oliver, R. K., Webber, M. D., et al. (1982). Supply-chain management: logistics catches up with strategy. *Outlook*, 5(1):42–47.

Pai, P.-F. and Lin, C.-S. (2005). A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505.

Papageorgiou, L. G. and Rotstein, G. E. (1998). Continuous-domain mathematical models for optimal process plant layout. *Industrial & engineering chemistry research*, 37(9):3631–3639.

Pureza, V. and Morabito, R. (2006). Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. *Computers & Operations Research*, 33(3):804–819.

Ribeiro, G. M. and Lorena, L. A. N. (2007). Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. *Computers & Operations Research*, 34(9):2695 – 2708.

Ribeiro, G. M. and Lorena, L. A. N. (2008). Optimizing the woodpulp stowage using lagrangean relaxation with clusters. *J. Oper. Res. Soc*, 59(5):600–606.

Scheithauer, G. and Terno, J. (1996a). A heuristic approach for solving the multi-pallet packing problem. *Decision making under conditions of uncertainty (cutting and packing problems)*, pages 140–154.

Scheithauer, G. and Terno, J. (1996b). The G4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, 47(4):511–522.

Scheithauer, G., Terno, J., Müller, A., and Belov, G. (2001). Solving one-dimensional cutting stock problems exactly with a cutting plane algorithm. *Journal of the Operational Research Society*, 52(12):1390–1401.

Scholl, A., Klein, R., and Jürgens, C. (1997). Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, 24(7):627–645.

Shumway, R. H. and Stoffer, D. S. (2006). Time series regression and exploratory data analysis. *Time Series Analysis and Its Applications: With R Examples*, pages 48–83.

Silva, E., Oliveira, J. F., and Waescher, G. (2016). The pallet loading problem: a review of solution methods and computational experiments. *International Transactions in Operational Research*, 23(1-2):147–172.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *https://arxiv.org/abs/1206.2944*.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm– a tutorial into long short-term memory recurrent neural networks. *https://arxiv.org/abs/1909.09586*.

Tarnowski, A. G., Terno, J., and Scheithauer, G. (1994). A polynomial time algorithm for the guillotine pallet loading problem. *INFOR: Information Systems and Operational Research*, 32(4):275–287.

Tsai, R. D., Malstrom, E. M., and Kuo, W. (1993). Three dimensional palletization of mixed box sizes. *IIE transactions*, 25(4):64–75.

Van Dijk, A., Van Heerde, H. J., Leeflang, P. S., and Wittink, D. R. (2004). Similarity-based spatial methods to estimate shelf space elasticities. *Quantitative Marketing and Economics*, 2(3):257–277.

Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86(3):565–594.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vargas-Osorio, S. and Zuniga, C. (2016). A literature review on the pallet loading problem. *REVISTA DIGITAL LAMPSAKOS*, pages 69–80.

Vitasek, K. (2016). Supply chain management terms and glossary. supply chain visions. *Strategic Outsourcing: An International Journal*, pages 126–138.

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

Westerlund, J. and Papageorgiou, L. G. (2004). Improved performance in process plant layout problems using symmetry-breaking constraints. *Discovery through Product and Process Design, Proceedings of FOCAPD*, pages 485–488.

Wu, K. and Ting, C. (2007). A Two-phase algorithm for the manufacturer's pallet loading problem. In *Industrial Engineering and Engineering Management, 2007 IEEE International Conference*, pages 1574–1578.

Yamassaki, C. A. and Pureza, V. (2003). A refinement of dowsland's tabu search algorithm for the manufacturer's pallet loading problem. *Production*, 13(3):6–17.

Yang, M.-H. and Chen, W.-C. (1999). A study on shelf space allocation and management. *International journal of production economics*, 60:309–317.

Yi, J., Chen, X.-G., and Zhou, J. (2009). The pinwheel pattern and its application to the manufacturer's pallet-loading problem. *International Transactions in Operational Research*, 16(6):809–828.

Yoo, T.-W. and Oh, I.-S. (2020). Time series forecasting of agricultural products' sales volumes based on seasonal long short-term memory. *Applied Sciences*, 10(22):8169.

Young-Gun, G. and Kang, M.-K. (2001). A fast algorithm for two-dimensional pallet loading problems of large size. *European Journal of Operational Research*, 134(1):193–202.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270.

Zufryden, F. S. (1986). A dynamic programming approach for product selection and supermarket shelf-space allocation. *Journal of the operational research society*, pages 413–422.

Appendix A

Publications

- Aljuhani, D. M. and Papageorgiou, L. G. (2021). Improved layout structure with complexity measures for the manufacturer's pallet loading problem (mplp) using a block approach. *Journal of Industrial Engineering and Management*, 14(2):231–249

- Aljuhani, D. M. and Papageorgiou, L. G. Dynamic framework for the shelf space allocation problem (SSAP). Final stage before submission.

- Aljuhani, D. M. and Papageorgiou, L. G. A Multi-objective optimisation based approach for the manufacturer's pallet loading problem. Final stage before sub-mission

- D. Aljuhani and L.G. Papageorgiou, "Automated Design of Assortments". 11th AIMMS-MOPTA Optimization Modeling Competition (https://coral.ise.lehigh.edu/ mopta2019/) at Lehigh University, Bethlehem PA, USA (14-16 August 2019), Finalist award received.

- D. Aljuhani and L.G. Papageorgiou, "Improved Layout structure with Complexity measure for the Manufacturers pallet-loading problem (MPLP) using a Mega-box Approach", 15th ESICUP Meeting: EURO Special Interest Group on Cutting and Packing workshop/conference. Zoetermeer, The Netherlands (23-25 May 2018).(https://paginas.fe.up.pt/ esicup/extern/esicup-15thMeeting/) .

- D. Aljuhani and L.G. Papageorgiou, “Improved layout for the pallet-loading problem using mega-box Representation”, 2nd IEOM European International Conference on Industrial Engineering and Operations Management (https://www.ieseg.fr/en/events/2nd-ieom-europeaninternational-conference-on-industrialengineering-and-operations-management/), Paris, France (26-27 July 2018).

Appendix B

AIMMS Graphical User Interface (GUI)

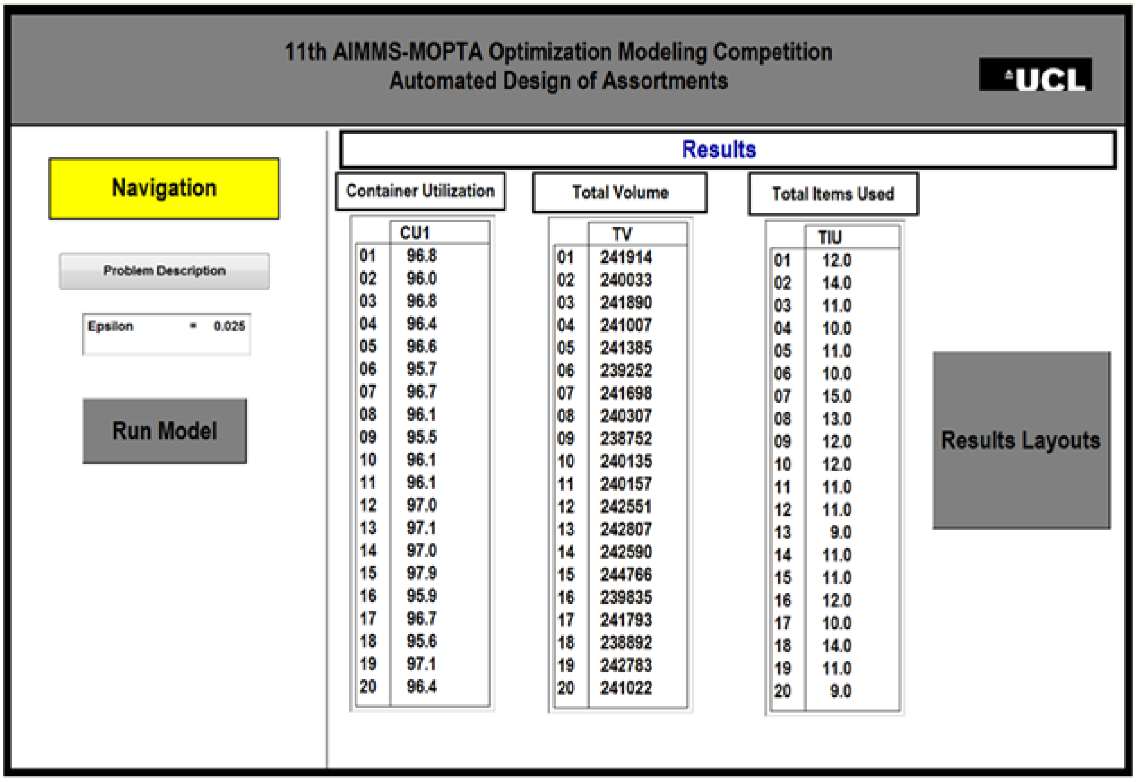


Figure B.1: GUI Interface

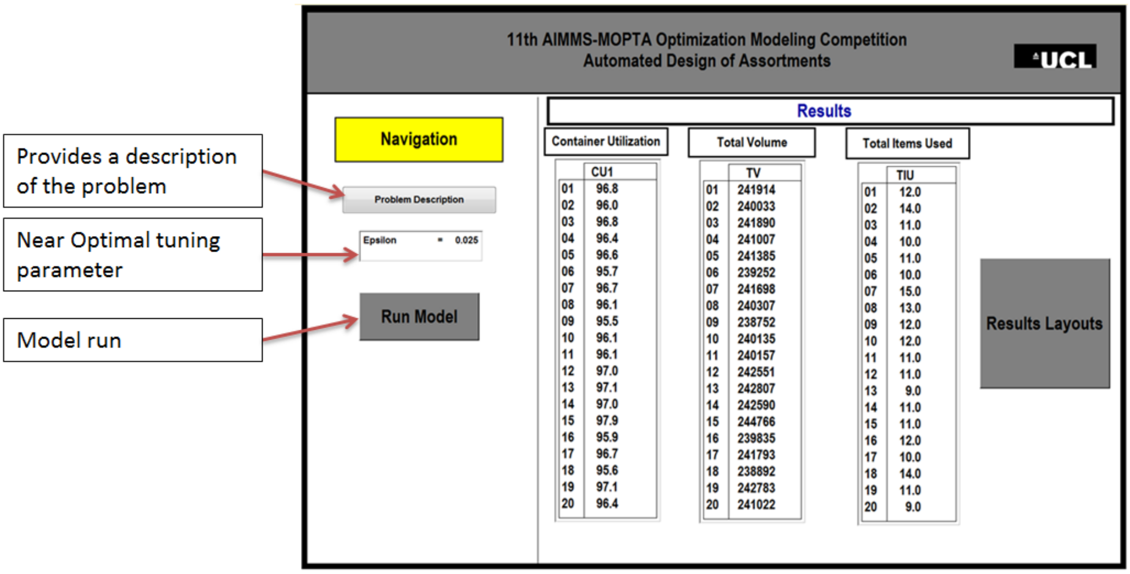


Figure B.2: GUI Interface

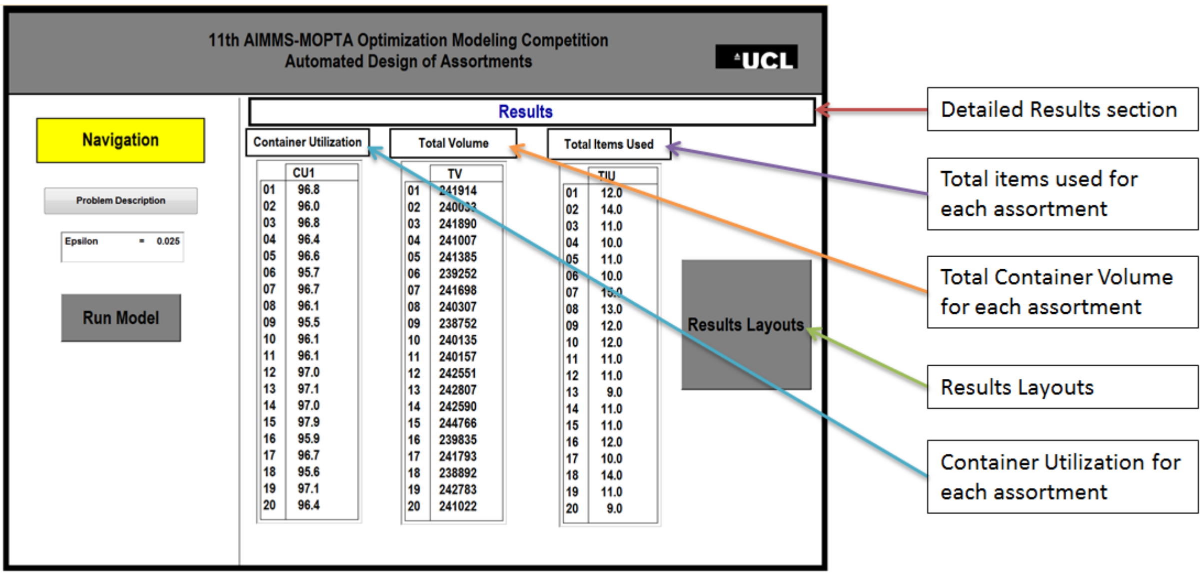


Figure B.3: GUI Interface

Appendix C

Shelf Space Allocation

C.0.1 Autoregressive Integrated Moving Average (ARIMA)

A example of an autoregressive model of order p as defined in Shumway and Stoffer (2006) can be explained as the following:

$$\mathrm{Y_t} = \zeta + \phi_1 \mathrm{Y_{t-1}} + \phi_2 \mathrm{Y_{t-2}} + \ldots + \phi_\mathrm{p} \mathrm{Y_{t-p}} + \mathrm{e}_t \tag{C.1}$$

Where $\phi_1, \ldots, \phi_p$ is autoregressive parameter and ζ is constant. A back-shift operator could be used to overcome the difficulties of the models random regressors ($\mathrm{Y_{t-1}}$.... $\mathrm{Y}_{\mathrm{t}-p}$) as the following for the $AR(p)$:

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) Y_t = e_t \tag{C.2}$$

The autoregressive operator $\phi_p(B)Y_t$ is a polynomial of degree p as the following:

$$\phi_p(B)Y_t = e_t \tag{C.3}$$

The moving average model of order q, abbreviated as MA(q), assumes the white noise e_t on the right-hand side of the equation are combined linearly to form the observed data as an alternative to the autoregressive representation in which the Y_t is on the left-hand side of the equation. An order q moving average model is as follows (Shumway and Stoffer, 2006):

$$\mathrm{Y_t} = \mathrm{e}_t + \theta_1 \mathrm{e_{t-1}} + \theta_2 \mathrm{e_{t-2}} \ldots + \theta_q \mathrm{e_{t-q}} \tag{C.4}$$

Where $\theta_1, \ldots, \theta_q$ is moving average parameter. An equivalent representation of MA(q) process can be written as the following:

$$Y_t = \theta(B) e_t \tag{C.5}$$

The moving average operator is:

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q. \tag{C.6}$$

In general, time series data must be stationary in order for lagged products to be averaged over time. So we can calculate the variations between successive measurements to make a non-stationary time series stationary. The ARIMA model of Box and Jenkins relies heavily on the differencing strategy. The right amount of differencing is usually the lowest order of differencing that produces a time series that fluctuates around a well-defined mean value and whose autocorrelation function (ACF) plot decays very quickly to zero from above or below. If the series still has a long-term trend or does not appear to revert to its mean value or its autocorrelations are positively affected by a high number of lags (i.e., 10 or longer), then a greater order of separation is needed. An Autoregressive Integrated Moving Average (ARIMA) is a broader class of the ARMA models, including differencing. The differenced data model ARMA(p,q) is similar to the ARIMA(p,d,q) model.

$$\mathrm{Y_t} = \zeta + \phi_1 \mathrm{Y_{t-1}} + \phi_2 \mathrm{Y_{t-2}} + \ldots + \phi_\mathrm{p} \mathrm{Y_{t-p}} + \mathrm{e_t} - \theta_1 \mathrm{e_{t-1}} - \theta_2 \mathrm{e_{t-2}} \ldots - \theta_q \mathrm{e_{t-q}} \tag{C.7}$$

In general, ARIMA model can be define as:

$$\phi(B)(1-B)^d Y_t = \alpha + \theta(B) e_t \tag{C.8}$$

C.0.2 Augmented Dickey Fuller Test (ADF)

The ADF test is one of the tests used to determine if the data used shows any stationary; it is also known as the ”Unit Root Test”. A time series’ unit root is a function that renders it as non-stationary. In the equation below, a unit root exists when $alpha = 1$ in a time series.

$$Y_t = \alpha Y_{t-1} + e_t \tag{C.9}$$

If $\alpha < 1$ then the process is stationary or $\alpha = 1$ we get non-stationary.

The above equation can be alternatively written as

$$Y_t - Y_{t-1} = \Delta Y_t = \delta Y_{t-1} + e_t \tag{C.10}$$

where $\delta = \alpha - 1$. For non-stationarity, the condition now becomes $\delta = 0$ the alternative hypothesis being $\delta < 0$. The null and alternate hypothesis are:

$$H_0 : \delta = 0$$
$$H_1 : \delta < 0$$

In case, the null hypothesis is that $\delta = 0$, i.e., there is a unit root—the time series is non-stationary. As non-stationarity can exist in three ways, the dickey fuller test is estimated in three different forms

1. Y_t is a random walk: $\Delta Y_t = \delta Y_{t-1} + e_t$

2. Y_t is a random walk with drift: $\Delta Y_t = \beta_1 + \delta Y_{t-1} + e_t$

3. Y_t is a random walk with drift around a deterministic trend : $\Delta Y_t = \beta_1 + \beta_2 t + \delta Y_{t-1} + e_t$

Dickey and Fuller have developed a test, known as the augmented Dickey–Fuller (ADF) test. This test is conducted by “augmenting” the three equations by adding the lagged values of the dependent variable. The null hypothesis and alternative hypothesis are given by

$$H_0 : \gamma = 0 \ (the\ time\ series\ is\ non-stationary)$$
$$H_1 : \gamma < 0 \ (the\ time\ series\ is\ stationary)$$

where
$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} + \delta_2 \Delta Y_{t-2} + \dots$

C.0.3 Support Vector Regression (SVR)

The SVR linear function can be expressed as follows:

$$f(\mathbf{X}) = \langle \varphi(\mathbf{X}), \mathbf{w} \rangle + b \tag{C.11}$$

Where;$f(\mathbf{X})$ are the forecasted values, $\varphi(\mathbf{X})$ is the function of mapping the input data into the higher-dimensional space, $\mathbf{w}$ is the weight factor/autoregressive coefficient, **b** is the adjustable factor (error value)and $\langle .,. \rangle$ is the dot product.

An optimisation problem is needed to solve the function $f(\mathbf{X})$ by minimising the coefficients; finding the narrowest tube centred around the surface while minimising the regression error ε,i.e, the difference between the predicted and ideal outputs and maximising the margin of the hyperplane. Thus the constrained optimisation problem can be written as the following:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \tag{C.12}$$

subject to:

$$y_i - f(x_i) \leq \varepsilon \tag{C.13}$$
$$f(x_i) - y_i \leq \varepsilon \tag{C.14}$$

In order to take into consideration the probability of errors that are larger than

ε, a slack variable ξ is assigned to each instance. The new optimisation problem can be written as the following:

$$\min_{\mathbf{w},\xi,b}\left\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right\} \tag{C.15}$$

subject to:

$$y_i - f(x_i) \leq (\varepsilon + \xi_i) \tag{C.16}$$

$$f(x_i) - y_i \leq (\varepsilon + \xi_i^*) \tag{C.17}$$

$$\xi_i, \xi_i^* \geq 0 \tag{C.18}$$

Where C is a penalty for observations that fall outside the epsilon margin ε constant value. Figure C.1 shows an example plot of an SVR process. Equation C.15 is solved using Lagrange multipliers, which transform the optimisation problem into its dual formulation:

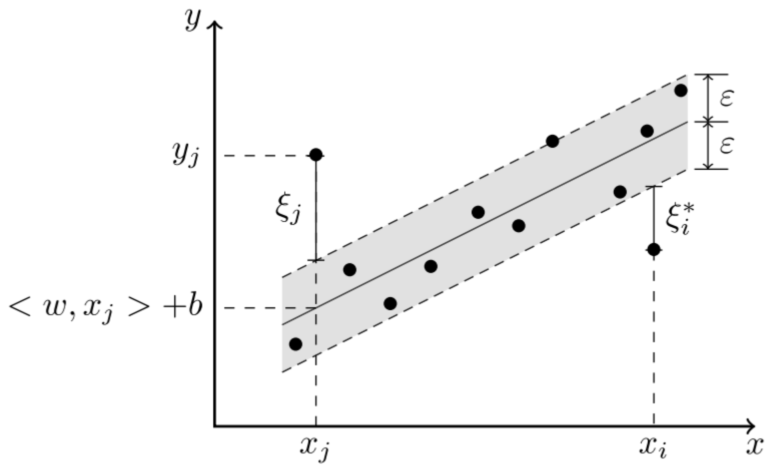


Figure C.1: Support vector regression example (Moustapha et al., 2018)

$$maximise\left\{\begin{array}{l}\frac{-1}{2}\sum\limits_{i,j=1}^{l}(\alpha_i+\alpha_i^*)(\alpha_j+\alpha_j^*)k(x_i,x_j)\\ -\varepsilon\sum\limits_{i,j=1}^{l}(\alpha_i+\alpha_i^*)+\sum\limits_{i,j=1}^{l}y_i(\alpha_i+\alpha_i^*)\end{array}\right. \tag{C.19}$$

subject to:

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) = 0. \tag{C.20}$$

Where $\alpha_i, \alpha_i^* \in [0, C]$ are Lagrange multipliers and $k(x_i, x_j)$ is kernel function. From Equation C.12:

$$\mathbf{w} = \sum_{i=1}^{l}(\alpha_i + \alpha_i^*)\Phi(\mathbf{x_i}) = 0. \tag{C.21}$$

By solving the optimisation model, the SVR model can is written as:

$$f(\mathbf{X}) = \sum_{i=1}^{n}(\alpha_i + \alpha_i^*)\, k(x_i, x) + b \tag{C.22}$$

C.0.4 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks take the general principle of feed-forward neural networks and enable them to handle sequential data by giving the model an internal memory. Figure C.2 shows the folded and unfolded versions of a single layer RNN, with the unfolded structure of the RNN showing the calculation done at each time step t. In the figure, X_t and Y_t are the input and corresponding output vector respectively, h_t is the hidden layer, and W the weight matrix between the input layer and the hidden layer, U is the weight matrix between the hidden layer and the hidden layer at (t-1) and V is the weight matrix between the hidden layer and the output layer. The parameters b_h and b_v are bias vectors. The hidden layer h_t serves as memory and is calculated using the previous hidden state h_{t-1} and the input X_t. The mathematical model of RNN is expressed as follows:

$$\begin{aligned} h_t &= q(WX_t + Uh_{t-1} + b_h) \\ Y_t &= r(Wh_t + b_v) \end{aligned} \tag{C.23}$$

q = tanh and r = sigmoid are the activation functions of the hidden layer and the output layer, respectively. one of the main drawback of RNN is vanishing gradient. This Recurrent means that long-term data must travel through all cells before reaching the current unit. This means that it is easily weakened by being multiplied

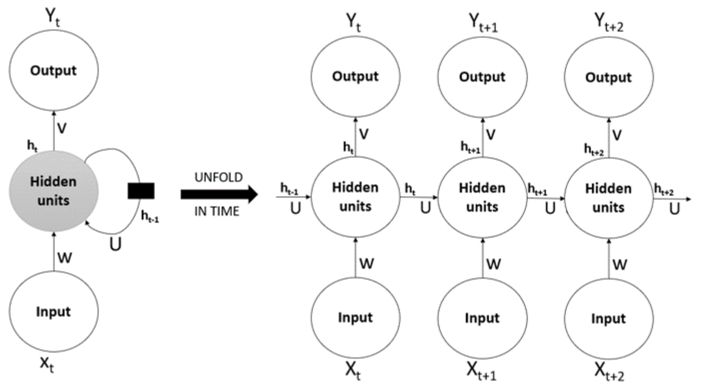


Figure C.2: Recurrent neural network (RNN) structure

several times by small numbers < 0.

C.0.5 Long Short Term Memory (LSTM)

1. Forget gate: receive the previous hidden state h_{t-1} and the current input x_t and decided which information to keep or forget. It uses a sigmoid function convert the output value into the range 0 to 1 for forget and keep respectively.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{C.24}$$

2. Input gate: receive the previous hidden state h_{t-1} and the current input x_t and decided which information to be used to update the cell state C_t by using a sigmoid to update and $tanh$ to create a candidate cell state $\tilde{c_t}$, ranging from -1 to 1, to determine what information to add or subtract from these entries.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ \tilde{c_t} &= \tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \end{aligned} \tag{C.25}$$

3. At this stage the values of the forget gate, the input gate and the candidate cell state $\tilde{c_t}$ are used to compute the current cell state c_t. $\circ$ is the scalar product of the two vectors.

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c_t} \tag{C.26}$$

4. The output gate: compute the output information o_t and the current hidden

state h_t which carried to the next time step and passes to next layer or prediction layer.

$$\begin{aligned} o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ h_t &= \tanh(c_t) \circ o_t \end{aligned} \tag{C.27}$$

where the W terms denote the different weight matrices, b term with a subscript is the bias vector for each gate. The output sequence Y_t can be obtained by

$$Y_t = W_{hy}h_t + b_y \tag{C.28}$$