UNIVERSITY OF BAHRAIN

College of Information Technology

A Novel Algorithm to solve a Multi-objective 3D bin-packing problem

Thesis submitted in partial fulfillment of the requirements for the
M.Sc Degree in **Information Technology**

Submitted by:

Jasim Hasan Ebrahim

19990385

Supervised by:

Dr. Jihene Kaabi

(Assistant Professor)

Dr. Youssef Harrath

(Assistant Professor)

University of Bahrain

Kingdom of Bahrain

February, 2019

UNIVERSITY OF BAHRAIN

College of Information Technology



A Novel Algorithm to solve a Multi-objective 3D bin-packing problem

Thesis submitted in partial fulfillment of the requirements for the **M.Sc** Degree in **Information Technology**

Submitted by:

Jasim Hasan Ebrahim

19990385

Supervised by:

Dr. Jihene Kaabi

(Assistant Professor)

Dr. Youssef Harrath

(Assistant Professor)

University of Bahrain

Kingdom of Bahrain

February, 2019

University of Bahrain
Deanship of scientific Research



جامعة البحرين
عمادة البحث العلمي

Dissertation Deposit Form

Author: Jasim Hasan Ebrahim

Thesis Title: A Novel Algorithm to solve a Multi-objective 3D bin-packing problem

Year: 2019

College: Information Technology Email: jasim.bh@gmail.com

Department: Information Technology Contact No: 39947977

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author: ---------- Date: 08/10/2019

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed: ---------- Date: 22/4/2019

Jasim Hassan Ebrahim (Candidate)

The Thesis defense committee considers the thesis titled:

Multi-objective 3D bin-packing problem

Submitted by Jasim Hassan Ebrahim to the College of Information Technology is satisfactory and acceptable for the Master of Science in Information Technology.

The Defense Committee:

Dr. Jihene Kaabi
(Supervisor)

Department of Information Systems
University of Bahrain

Dr. Youssef Harrath
(Co-Supervisor)

Department of Computer Science
University of Bahrain

Prof. Adel Khelifi
(Member)

American University at UAE

Prof. Nabil Hewahi
(Member)

Department of Computer Science
University of Bahrain

Abstract

The Bin-packing problem is a widely researched subject, and still being researched till present to reach solutions which are near optimum. There are many variants of the problem, depending on the objective of the research, each has multiple solutions proposed by researchers. In this research, we investigate a multi-objective 3D bin-packing problem and propose a novel algorithm where we consider the volume and weight as constraints, and use the minimum number of bins to pack different-sized boxes. The resulting bins should be balanced as much as possible. The proposed solution is a layer-based approach, the proposed algorithm packs all boxes using horizontal layers, it has shown promising results in terms of number of bins and execution time, and provides flexibility to be applied on different variations of the bin-packing problem. The results have been compared and successively validated with a theoretical lower bound and a previously published algorithm that tackled the same problem.

Dedication

To:

My wife, for her unconditional support

My daughter Hawra and my son Ali, for showing their nonstop love, while coping with the long time my study took me away from them.

Acknowledgment

I would like to thank Dr. Jihene Kaabi and Dr. Youssef Harrath, not only for their time and extreme patience, but also for their intellectual contributions to my development as a researcher.

Table of contents

Topic	Page Number
Approval	A
Abstract	B
Dedication	C
Acknowledgement	D
Table of Contents	E
List of Tables	G
List of Figures	H
Chapter One: Introduction	
1.1. Overview	2
1.2. Bin-packing problems overview	3
1.3. Problem Statement	5
1.4. Research Objectives	5
1.5. Research Scope and Limitations	5
1.6. Contribution	7
1.7 Methodology	7
1.8. Thesis Outlines	8
Chapter Two: Literature Review	
2.1. Introduction	10
2.2. Exact algorithms approach	10
2.3. Evolutionary and nature-based algorithms approach	11
2.4. Heuristic algorithms approach	15

2.5. Conclusion	21
Chapter Three: The proposed Approach	
3.1 Introduction	23
3.2 Problem Description	23
3.3 Bounds and Problem complexity	25
3.4 The Proposed Algorithm (LBA-3DBPP)	26
3.5 Conclusion	30
Chapter Four: Experimental Results	
4.1 Introduction	32
4.2 Implementation Parameters	32
4.3 Implementation	32
4.4 Experimental Results	33
4.5 Discussion	37
4.6 Conclusion	38
Chapter Five: Conclusion	
5.1 Conclusion	40
5.2 Future work	40
References	42
Appendix 1 – Finding Layers solutions source code	49
Appendix 2 – bin-packing source code	51
Appendix 3 – AddBin SQL stored procedure	53

List of Tables

No	Topic	Page Number
1	Table 3.1: Bin-packing problem notations	25
2	Table 3.2: LBA-3DBPP algorithm notations	30
3	Table 4.1: LBA-3DBPP implementation parameters	32
4	Table 4.2: LBA-3DBPP algorithm box types results	33
5	Table 4.3: LBA-3DBPP algorithm Layers solutions	34
6	Table 4.4: LBA-3DBPP experiment results	35

List of Figures

No	Topic	Page Number
1	Figure 1.1: Containers standard sizes (20ft and 40ft)	3
2	Figure 1.2: Fedex 10kg box	7
3	Figure 2.1: Final packing patterns, bin number, its volume, and weight percentage. Saraiva et al (2015)	20
4	Figure 3.1: Bin and box dimensions	24
5	Figure 3.2: orientation option 0	26
6	Figure 3.3: orientation option 1	27
7	Figure 3.4: layers layout in a Bin (3x25kg box +2x10kg box)	27
8	Figure 4.1: LBA-3DBPP volume and weight utilization results for 2,500 boxes	36
9	Figure 4.2: LBA-3DBPP 2,500 boxes solution bin's weight	37
10	Figure 4.3: Comparison for number of Bins versus number of Boxes	38

Chapter One

Introduction

1.1. Overview
1.2. Bin-packing problems overview
1.3. Problem Statement
1.4. Research Objective
1.5. Research Scope and limitations
1.6. Contribution
1.7. Methodology
1.8. Thesis Outlines

Chapter One: Introduction

1.1 Overview

In the courier industry, the general bin-packing problem consists on filling container(s) with all boxes to be shipped, by leaving the minimum empty space in the container as possible. A variation of the aforementioned problem, namely 3D bin-packing problem, takes into consideration the height, width, and depth of each box. In addition, the box weight as well as the container's maximum allowed weight are considered.

The multi-objective 3D bin-packing problem is an active research topic in the literature. Many research papers (Martello et al. (2000), Jangha and Sungsoo (2003), Mongeau and Bes (2003), and Elhedhli et al. (2010)) were written to find the optimum solution using various algorithms. In this research, a novel algorithm is proposed to fill the bins layer by layer, the way the layers will be formed is first constructed, with any number of box items, and then a bin-packing procedure will construct the layers for the same box items.

The solution is applicable for any courier company with the same constraints and objectives of this research. The outcome solution is to improve and automate the couriers' bin-packing process, optimizing multiple criteria.

Researchers have examined many variations of bin-packing problems, and recently more focus on multi-objective is being studied (Sridhar and Page (2016), Kanna and Campus (2017), Piyachayawat and Mungwattana (2017), Sridhar and Page (2017), and Sheng (2017)), which make the studies more complicated, but more realistic as they are more based on real life problems, especially with courier industry.

Many courier companies use expert judgment for the task of bin-packing and they should consider the load balance of the bins, utilize the minimum number of bins, and optimize the placement of the boxes. This is not an easy task and gets complicated with huge number of received boxes.

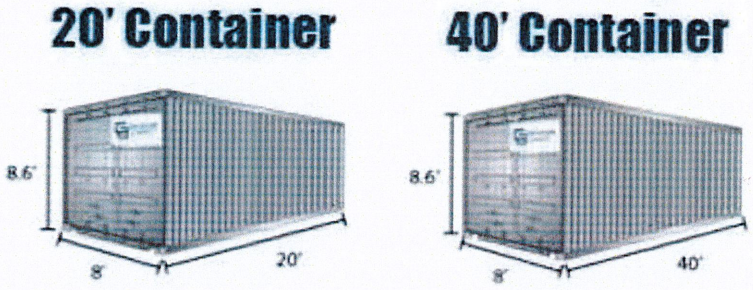


Figure 1.1: Containers standard sizes (20ft and 40ft)

In the courier industry, the containers have two standard sizes, 20ft long, and 40ft long, for both sizes, the height is 8.6ft, and the width is 8ft (Figure 1.1). The boxes have many dimensions in terms of weights, shapes, and volumes. As Couriers ships all kind of shipments, the boxes are placed in many shapes of packages, they use 3 different size of boxes (to be filled inside the container) for papers, books, etc. They accept larger items that are directly placed in the container without packaging them into boxes.

This research will refer to containers as bins, and any shipment packed or to be packed in the bin(s) as box(es).

1.2 Bin-packing problems overview

Bin-packing problems can be classified as online or offline problem, online means the boxes to be packed in containers as they arrive, while offline, packing the boxes starts once all the boxes are well known before starting solving the problem. Our study focuses on an offline bin-packing problem.

Some criteria to be considered for the bin-packing problem are the dimensions of the boxes and bins. In 1D bin-packing problem, only one property is considered for solving the problem, like the weight of boxes.

In 2D problems two properties are considered, for example, length and depth of small rectangles into bigger rectangles, or weight and volume in gas and liquid shipping. While in 3D problems, three properties are considered, like: width, height, length, weight, etc. such as packing boxes into containers, which has wide usage in courier, factories shipping, and logistics industry. This research considers a 3D bin packing problem. In addition, the dimensions of boxes or bins, could be of fixed or variable sizes. In this research, we assume that the bins are of fixed dimensions and fixed maximum weight, whereas the boxes are of different dimensions and weights, but all known before solving the problem. The boxes and bins have cuboid shapes.

The Bin-packing problems could also have a single objective, or multiple objectives to be optimized when solving the problem. Our research focuses on a multi-objective 3D bin-packing problem.

The literature has different variations for the bin packing problem, and has been classified in many ways (Dyckhoff and Finke, 1992). Problems similar to the subject of this research are referred to as 3D bin-packing problems, where all the boxes have to be packed inside bins, and the objective is minimizing the cost by minimizing the number of containers. However, other problems have tolerance of leaving out boxes not packed into containers; these problems refer to as 3D knapsack problems, and the objective is to maximize the bins' utilized volume.

Another classification is based on the type of boxes to be packed. The boxes of the same type, have the same shape and dimensions.

Therefore, the problem could have homogeneous set of boxes (one box type only), weakly heterogeneous set of boxes (few box types, many of each type), or strongly heterogeneous set of boxes (many box types, and few boxes of each type).

The subject of our research can be classified as 3D bin-packing problem, with weakly heterogeneous set of boxes, for minimizing the number of bins, and balance them.

1.3 Problem Statement

The purpose of this research is to solve a real-world problem of bin-packing in the courier industry, of packing all the boxes into minimum number of bins as possible, without exceeding neither the maximum allowed weight of bins, nor the bin's dimension.

1.4 Research Objectives

The objective of this research is to propose a novel algorithm for solving multi-objectives 3D bin-packing problem, such that:

1. Minimize the number of bins used.
2. Balance the load of all the bins, each loaded bin's weight should be as close to average bins weight as possible.

1.5 Research Scope and Limitations

The research will does not different bin's sizes, and will not consider bins rotation, as the bins are fitted only in one way while being shipped, by any shipment method (air, land, sea).

The constraints considered for this research are limited to the following inputs:

1. Bin dimension
2. Bin maximum allowed weight
3. Box dimension
4. Box weight

The functional constraints to be considered:

1. The total boxes' weight within a bin shouldn't exceed the bin maximum allowed weight.
2. The total boxes' volume within the bin shouldn't exceed the bin volume.
3. The total boxes dimensions shouldn't exceed the bin dimensions, in terms of length, width, and depth.

Box types, in terms of dimensions and weight need to be defined at the beginning of the execution of the algorithm. The boxes have to be in cuboid shape.

The rotation of boxes is allowed only horizontally, it will not be considered vertically, and the base of the boxes is always the bottom face (Figure 1.2).

There will be no limitation on what boxes to be placed on the top of other boxes, but the base of the box has always to be at the bottom, as per (Figure 1.2):



Figure 1.2: Fedex 10kg box

1.6 Contribution

This research contributes to a real world problem of 3D bin-packing problem with a novel algorithm. It is not only limited to the cargo industry, but it can be expanded to any logistic service in any industrial sector.

The novel algorithm will be evaluated and scaled up and down in future studies, in order to make it applicable to many bin-packing problems, which is an addition to the research and practical worlds.

1.7 Methodology

The literature will be scanned for approaches to solve the 3D bin-packing problem, and approaches will be compared to find the best approach for solving the multi-object 3D bin-packing problem with objectives similar to those of this research. The most suitable approach will be adapted to create a new algorithm.

The data will be collected from a courier industry. The proposed algorithm will be implemented, tested and compared with existent

algorithms. A deep experimental study will be conducted to evaluate the performance of the proposed solution approach.

1.8 Thesis Outlines

The thesis report will contain five chapters. After introducing the problem in chapter 1, a deep review of existent works related to the investigated problem will be reported in chapter 2. Chapter 3 details the proposed novel algorithm to solve a 3D bin-packing problem. In chapter 4, results of an experimental study conducted to test the propose approach as well compare it with existent algorithm will be displayed. Finally, chapter 5 will conclude the entire research, and highlights possible future about this research.

Chapter Two

Literature Review

2.1. Introduction
2.2. Exact algorithms approach
2.3. Evolutionary and nature-based approach
2.4. Heuristic algorithms approach
2.5. Conclusion

Chapter Two: Literature Review

2.1 Introduction

This chapter reviews the literature for previous researches on the bin-packing problems. The literature is reviewed based on the proposed solution type: exact algorithm, evolutionary and nature-based, and finally, heuristic approach solutions.

2.2 Exact algorithm approach

The approach used to find an optimal solution is called exact approach. The Branch-and-Bound algorithm proposed by Martello et al. (2000) to get an optimal solution for 3D rectangular bins is one of the exact approaches. The researchers used examples of up to 90 items, and have shown reasonable time, however, as the number of items, and items dimensions increase, the execution time will considerably increase.

Jangha and Sungsoo (2003) proposed a modified First-Fit and Best-Fit-Decreasing algorithms for packing variable size boxes into minimum number of bins, the proposed algorithms are: iterative-first-fit and iterative-best-fit-decreasing, where the standard algorithm is used initially to pack the boxes, then repack the packed boxes iteratively until an optimal solution is reached. However, the problem doesn't consider constraints, and will take longer time for obtaining a solution, as the number of boxes increases.

Mongeau and Bes (2003) focused on optimizing the load of boxes into aircraft container with multi-objectives of minimizing the fuel consumption, maximizing the number of loaded boxes, and satisfying the

stability criteria using integer linear programming method. The Air container was divided into two along longitudinal axis of aircraft, and the boxes must be loaded on both containers simultaneously to maintain center of gravity along the axis.

Elhedhli et al. (2010) proposed a Branch-and-Price algorithm to solve the bin-packing problem, specifically for scheduling and resource allocation. The algorithm was compared to similar approaches and the results indicate efficiency and stability for the algorithm.

Exact algorithm aims to provide optimal solutions. However, it works in reasonable time for small problems, where the number of boxes is small, and few boxes dimensions are to be considered.

The exact algorithms are usually not able to find an optimal solution in reasonable time when the number of boxes and boxes types increase.

2.3 Evolutionary and nature-based algorithms approach

Evolutionary approaches are well suited to solve large bin-packing problems; to achieve near optimal solutions. The constraints added to the problem aim to mimic the practical problems of bin-packing, have big impact on the complexity of the problem, and hence, the efficiency of the solution proposed Hopper and Turton, (1998); Tavakkoli et al, (2009); and Li, X., Zhao, Z., & Zhang, K. (2014).

Hopper and Turton (1998) studied Genetic Algorithm (GA) applications on 2D and 3D bin packing problems. They have provided classification for the bin packing problem and genetic operators, and came to the result that the GA can be used for various bin packing

optimization problems, and the complexity of the problem increases as constraints and geometric parameters increase.

Bortfeldt and Gahring (2001) proposed a hybrid Genetic Algorithm for 3D bin packing problem for a single container, considering the following constraints:

1. Box edges constraint being parallel to container wall.
2. Overlapping constraint (boxes have to be placed on the top of each other).
3. Boxes packed within container boundary.
4. Rotation constraint.
5. Stacking constraint.
6. Container maximum allowed weight.
7. Stability and balance constraints.

Boxes considered for the problem are strongly heterogeneous, and the proposed algorithm tested using procedures from other authors. The computing time is within acceptable limits. Pimpawat and Chaiyaratana (2001) proposed a heuristic co-operative co-evolutionary GA to optimize the sequence of loading boxes into container, by dividing larger sequences into smaller sequences, then loading the boxes with larger sequence into the container from bottom-left corner of the container. An Integer based encoding was used for GA, and differentiated from traditional GA by using different means of selection of the population.

Leung et al. (2008) implemented Genetic Algorithm concept to optimize the packing and distribution of carton boxes. Number of boxes to be filled in bins was decided by the Genetic Algorithm output, and also the size of the box that can fit into the bin. The concept was implemented for a Towel manufacturer, to decide the size of box to pack the towels in

it. The towels were categorized based on their types (wash towel, hand towel, bath towel), and packed based on layer-by-layer concept and as a result minimizing the box types. However, this research output is carton designs, but have common objective of maximum container utilization, similar to our research. Liu (2008) developed a multi-objective evolutionary particle swarm optimization algorithm to solve 2D multi-objective bin packing problem. Pareto's optimality concept used to evolve multiple solutions, and then numerical investigations performed to compare the solutions with other optimization algorithms, which have shown effectiveness of the proposed algorithm.

Wu et al (2010) proposed a mixed integer programming solution for 3D bin-packing problem, considering variable bin's height. The bin and boxes are both in rectangular shapes. The objective is to pack the boxes, such that, the bin's height is minimized. The boxes' rotation is allowed, and box's edges parallel to bin's edges. It has concluded that the proposed approach showed better results than manual packing, however, for larger problems with more constraint GA is better to use.

Li et al. (2014) considered that the bins are varying in size, they utilized modified Genetic algorithm, by developing a novel heuristic procedure. The heuristic procedure converts the boxes packing sequence and the bin packing sequence encoded in a chromosome into a packing solution. The purpose of using the heuristic procedure is to reduce the amount of time the algorithm will take for producing a near optimal solution.

Rudy and Żelazny (2015) considered a multi-objective 3D bin-packing problem. The objective is to maximize the number of boxes within a bin as well as the volume of boxes. They evaluated the solution of implementing two algorithms, simulated annealing, and genetic

algorithms, then the results of the implementation were compared, and showed that the genetic algorithm performs slightly better than the simulated annealing. They also observed that using less number of boxes (of same size) leads to better utilization of bin's volume. In addition, using more number of boxes will lead to worse utilization of containers. The research didn't consider different sizes and shapes of the boxes, which is an important constraint in the real world 3D bin-packing problem.

Sridhar and Page (2016) proposed an Adaptive Genetic Algorithm (AGA) to solve the 3D multi-objective bin packing problem. The constraints used are: number of bins, size, shape, weight, and container dimension. The solution proposed uses an implementation of (AGA) algorithm, which gets repeatedly applied until best solution is reached based on the input constraints. After that there could be empty spaces left in each bin, which gets filled by applying a Tuning algorithm. Finally, the solution is presented in a graphical format for the packers. The researchers stated that the proposed solution is not better nor the best, as no comparison was conducted. However, the researchers consider the results as satisfactory and effective in terms of execution time and results.

Kanna and Campus (2017) used Firefly algorithm for solving the 3D bin packing problem of arbitrary sized heterogeneous bins into standard size bin. They considered many constraints: placement, overlapping, stability, weight, load bearing, and orientation constraints. The algorithm was tested with real data and shows comparatively better results than the best fit method and the genetic algorithm. Piyachayawat and Mungwattana (2017) investigated an application of the 3D multi-objective bin-packing problem, which is a distributor's pallet loading problem with mutli-size boxes and pallets in lamp and lighting factory.

The objective was to find a loading solution that has the minimum amount of used area by pallets. A Genetic algorithm and differential evolution algorithm were used. The solution was evaluated by comparing the application's results with Human's solution from real problem, and concluded that the application's solutions are mostly better, by an average of 23% better than human expert judgment. However, the paper is about a unique application for the bin-packing, but it's similar to our research in terms of 3D multi-objective bin packing constraints and objectives. Moreover, the paper didn't use other algorithms for testing the performance of the proposed solution; rather they used human expert judgment real results for comparison.

Sridhar and Page (2017) built on (2016) proposed a hybrid genetic approach to find an optimal solution for the problem. They considered more practical constraints in the cargo industry; such as, box orientation, stack priority, bin's stability, weight, overlapping, and placement. They achieved the research goal, by optimizing the empty volume inside the bin using a genetic approach. The research lacks comparative analysis of other algorithms.

Although the literature suggested many efficient solutions for the multi-objective 3D bin-packing, still suggesting high or near optimal utilization of containers remain a challenge, and the subject is open for research.

2.4 Heuristic algorithms approach

Luitpold Babel (2004) proposed online box packing into minimum number of bins, considering the constraints: boundary crossing constraint and overlapping in the shortest time possible using two algorithms, first is modified first-fit algorithm, the second algorithm categorized the boxes

and packed them in separate bins. Since it is an online algorithm, the optimum or near to optimum solution wouldn't be reached similar to an offline algorithm, where all the boxes are known in advance of executing the algorithm.

Takahara (2006) researched a solution for the multiple containers loading problem. A new simple meta-heuristic algorithm was developed based on first-fit algorithm. To identify the best boxes arrangement within the container, a first-fit algorithm was used, then the Simulated Annealing algorithm used for reaching an optimal solution, considering load and weight distribution constraints. The boxes are allowed to be rotated to increase the effectiveness of the proposed solution. Dube and Kanavathy (2006) investigated a 3D bin-packing problem for rectangular shape item boxes and rectangular shaped Bins. The authors considered the rotation of item boxes in all directions and proposed a heuristic algorithm. The experimental results were graphically represented. The drawback of their research is that they didn't consider the weight of item boxes and didn't have a constraint for the maximum allowed bin's weight. They also mentioned that the solution was obtained within a reasonable time. However, they didn't present the execution time in the experimental results.

Wang and Levy (2008) proposed a heuristic algorithm based on a tertiary tree model for solving the problem of container loading problem (CLP). Weakly heterogeneous boxes were considered. The homogenous boxes are sorted and packed starting from lower left corner, then the remaining free container space is packed using space decomposition algorithm (optimal-fitting sequencing and an inner-right-corner occupying rules). The rotation of boxes was considered to increase solution effectiveness. The researchers compared the results with existing

algorithms, and have shown the effectiveness of the proposed algorithm. Coffman (2008) proposed an algorithm based on random order best-fit algorithm, where the boxes are packed into blocks such that the total volume of the block is almost equal to the bin's volume, and have shown that it gives the same result of next-fit algorithm. It's another way for solving the bin-packing algorithm, however, it doesn't consider the practical constraints, and wouldn't be timely efficient with large number of boxes. Cao (2008) studied two problems, truck scheduling and storage allocation. An integer programming model was proposed to minimize the queue and trucks travel time. Also heuristic and GA algorithms were experimented, and the results showed that the heuristic is best used for small problems (less than 5 bins), while GA suitable for large sized problems (more than 15 bins used).

Egeblad and Pisinger (2009) proposed a heuristic algorithm for 2D and 3D knapsack problems. The authors didn't consider any constraints, but considered boxes rotation, which makes their algorithm more theoretical than practical. They packed rectangular boxes into bins of fixed dimensions. Their novel work is an abstract representation of box placement, and they called it sequence triple. They've run comparative analysis to compare the proposed solution with various previous approaches, and obtained promising results. Huang et al (2009) developed a new algorithm for solving a 3D single container loading problem, with focus on packing outside box into a corner, such that, it was closely packed as possible. Rotation was considered, but the weight was not. The experiment was run on 800 strongly heterogeneous boxes, and has shown 87.97% average bin's utilization. Gurbuz and Akyokus (2009) proposed a new heuristic algorithm as well, which can accept different rectangular box types to be packed into rectangular shaped bin. They did not consider

any orientation for the boxes and height of the Bin. The efficiency of the proposed algorithm was measured by percentage of waste space within a bin after packing all boxes. They didn't consider the weight of boxes as a constraint, which makes the solution not applicable for the courier industry in general.

Wu et al (2010) proposed a mixed integer programming solution for a 3D bin-packing problem, considering variable bin's height. The container and boxes are both of rectangular shapes, and the objective is to pack the boxes, such that, the bin's height is to be minimized. The boxes rotation is allowed, and the box edges are parallel to the bin's edges. It was concluded that the proposed approach showed better results than the manual packing. However, for larger problems with more constraint GA is better to use. Han et al (2010) proposed a dynamic heuristic algorithm and a modified first-fit approach to minimize the number of containers required to pack all boxes. No constraints were considered. Thus, the researchers mainly focused on analysis of first-fit algorithm for problems of bin-packing type.

Kang et al. (2012) proposed an Improved Deepest Bottom Left with Fill (I-DBLF) algorithm utilizing a hybrid genetic algorithm, for the 3D-BPP to pack a single bin with both homogeneous and heterogeneous boxes. The objective was to pack maximum number of boxes with minimal space waste within the bin. The constraints considered were: 1. each box fit completely within the container. 2. Boxes are not allowed to overlap. 3. Each box is parallel to the bin's surface, and either supported by the bin's surface or other boxes. The rotation of boxes was also considered in the solution. The research didn't consider the bin's weight, and multiple bins, they left it for future work.

Li et al. (2014) considered that the bins are varying in size and developed a modified Genetic algorithm. The heuristic procedure converts the boxes packing sequence and the bin packing sequence encoded in a chromosome into a packing solution.

Öylek (2015) examined two bi-criteria bin-packing problems, both of them considered only the weights as constraints and focused on minimizing two objectives, namely the number of bins and the maximum over-deviation. The paper doesn't consider 3D bin-packing, which makes the studied problem less realistic and far away from the cargo industry bins packing. Saraiva et al. (2015) proposed a layer-based solution for the 3D multi-objective bin-packing problem. Their approach consists on building horizontal layers of identical items first, and then generating a packing pattern by greedily loading the layers using multiple selection criteria. The authors didn't consider real world constraints such as the weight of boxes, and bin maximum allowed weight. Their focus was to minimize the waste percentage, while providing speedy execution time. They've validated the proposed solution with by re-implementing wall-building algorithm by George and Robinson (1980), and their solution was showing better results. The results presented in the paper have shown huge variation in the weight of Bins, even when excluding first and last bins. The idea of using Layers speeded up the execution and finding the solution but the bins were not balanced, as shown in Figure 2.1

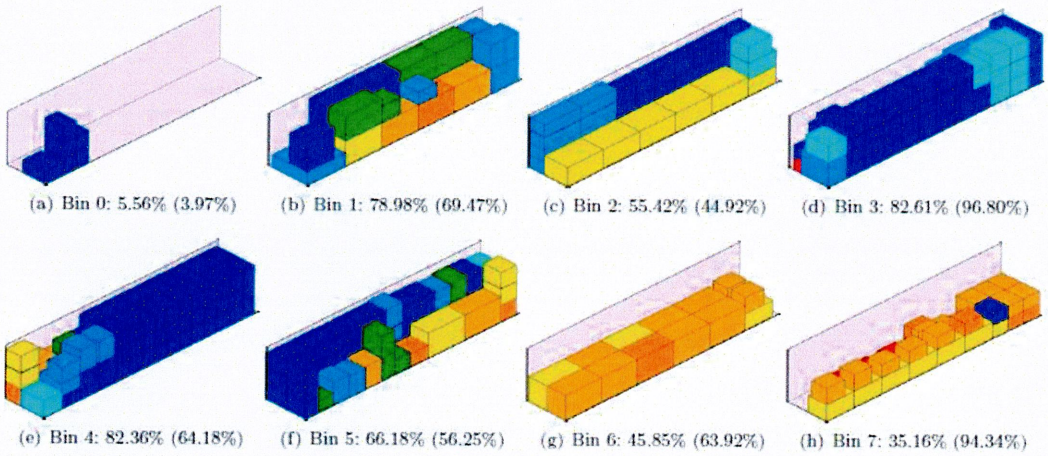


Figure 2.1: Final packing patterns, bin number, its volume, and weight percentage. Saraiva et al (2015)

Trivella and Pisinger (2016) considered load balancing of the bins, the minimum number of bins, and placement as objectives. The authors proposed a heuristic algorithm to solve the problem. The empirical study produced promising results. However, no comparative analysis was done, and the criteria used are less practical than the real world operation in the cargo industry, as it didn't consider the maximum allowed bin's weight.

Sheng (2017) addresses the problem of loading orders into containers for shipment at factories, so that boxes expiring soon should have priority in shipment, and boxes without expiry have the lowest priority. In addition, if a box of a given order is packed into a bin, the rest of the boxes of the same order must be packed within the same bin. However, if one box of an order cannot be packed in the same bin, all the boxes shouldn't be packed in that given bin. The problem is classified as 3D Single Knapsack Problem 3D-CLP, with the following constraints: orientation constraint, stability constraint, guillotine-cutting constraint, complete-shipment constraint, and the loading priority constraint. A Heuristic was proposed to solve the problem of order packing into bins. The heuristic first selects the order collections using Simulated Annealing

strategy, and then packs the collection into a bin using tree-graph search procedure. Finally, the algorithm is validated using experimental results.

The multi-objective 3D bin-packing could have multiple solutions suggested by the literature. However, the number of constraints and complexity of the requirements could impact the efficiency of the solution. According to Hopper and Turton (1998), Tavakkoli et al. (2009), Wu et al. (2010), Li et al. (2014), the Genetic Algorithm is the best approach to be used for medium to large bin-packing problems, and even with usage of the Genetic Algorithm, the way the chromosomes are encoded, could impact the time taken to reach the optimum or near to optimum solution.

2.5 Conclusion

The literature shows multiple classifications for the bin-packing problems, and have different approaches to solve the problem. The proposed solutions in the literature give guidelines on how to solve the 3D bin-packing problems, depending on the objectives, and the constraints to be considered in the solution. Despite, the 3D bin-packing problem has been discussed and investigated since long time, still the researchers are working on it with the aim of improving the solutions, and providing novel approaches with the consideration of advancements in the technology.

Chapter Three

The Proposed Approach

3.1. Introduction
3.2. Problem Description
3.3. Bounds and Problem complexity
3.4. The Proposed Algorithm (LBA-3DBPP)
3.5. Conclusion

Chapter Three: The Proposed Approach

3.1 Introduction

This chapter will illustrate the problem description, problem notations and bounds, then the proposed LBA-3DBPP (Layers Based Approach for the 3D Bin-Packing Problem) algorithm is explained, with all of its steps (3 steps). Finally, the advantages of the proposed algorithm will be discussed.

3.2 Problem Description

The 3D bin-packing problem consists of loading number of cubic boxes of different types (dimension and weight) into a minimum number of identical bins. The packed boxes must not exceed the bin's dimension and maximum allowed weight.

To illustrate the problem parameters, below are the definition of each of them, and how they will be used in the proposed algorithm:

- **Bin's dimensions**: Width, length and depth of the bin. These parameters can be set at the beginning of the algorithm execution and remain the same for the same problem instance.
- **Bin's Volume**: once the dimensions are set, the bin's volume gets calculated and stored.
- **Maximum bin's weight**: the maximum bin's weight allowed needs to be set prior to the execution of the algorithm and remains the same for the problem instance solved.

- **Box Types**: for each box type, the box dimension and maximum weight is set, they remain the same during the algorithm execution.
- **Variable parameters**: The only two parameters that may vary during the proposed algorithm execution are the box's weight and the number of each box type.

In our research we will need to formulate many constraints and objectives as mathematical expressions. Table 3.1 will show each variable definition, and Figure 3.1 illustrates its exact usage:

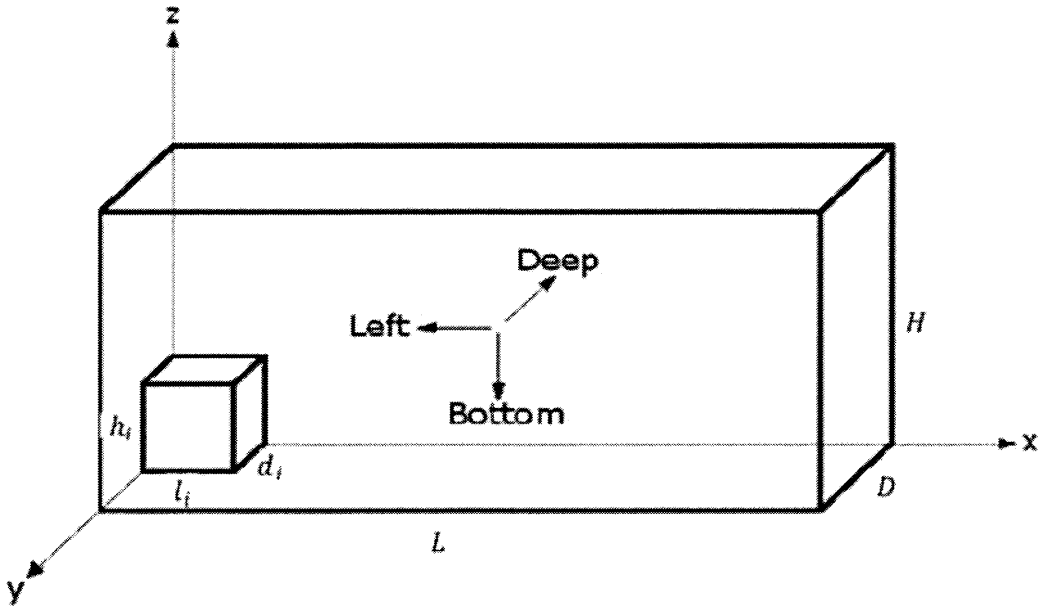


Figure 3.1: Bin and box dimensions

Table 3.1: Bin-packing problem notations.

Notation	Meaning
m	Number of bins (to be minimized)
t	Number of box types (in term of dimensions)
n	The total number of boxes of all types ($n = \sum_{i=1}^{t} n_i$)
n_i	Number of boxes of type i, $i=1, \ldots, t$
$D, L,$ and H	The depth, length and height of a bin
W	Maximum allowed weight of a bin
V	The volume of a bin ($V = D * L * H$)
d_i, l_i and h_i	The depth, length and height of a box of type i, $i=1, \ldots, t$
w_k	The weight of box k, $k=1, \ldots, n$
v_k	The volume of box k, $k=1, \ldots, n$

3.3 Bounds and Problem complexity

The problem consists on deciding for each bin $j; 1 < j < m$; the set of boxes that it will hold as well as how these boxes are to be placed. Let:

$$x_{kj} = \begin{cases} 1 \; if \; box \; k \; is \; placed \; in \; bin \; j \\ 0 \; otherwise \end{cases}$$

The objective is to minimize the number of bins used to pack all the boxes respecting the following constraints:

- The total boxes weight within a bin j shouldn't exceed the bin maximum allowed bin weight, equation (3.1) the formulation of this constraint:

$$\sum_{k}^{n} x_{kj} * w_k \leq W \qquad j = 1, \ldots, m \quad (3.1)$$

- The total boxes volume within the bin j shouldn't exceed the bin volume, equation (3.2) shows the formulation of this constraint:

$$\sum_{k=1}^{n} x_{kj} * v_k \leq V \qquad j = 1, \ldots, m \quad (3.2)$$

In order to set the lowest possible value for the optimal solution for the 3D bin-packing problem, we define the volumetric lower bound *lb*, which is formulated in equation (3.3):

$$lb = \frac{\sum_{k=1}^{n} v_k}{V} \qquad (3.3)$$

3.4 The Proposed Algorithm (LBA-3DBPP)

In order to meet the objectives of the research and satisfy all the functional constraints, a novel algorithm was designed which packs the boxes in the minimum a number of bins layer by layer. The proposed Layer Based Approach for 3D bin-packing problem, denoted by LBA-3DBPP, is composed of two parts; the first part will run only one time when setting the bin dimensions and maximum weight, and each box type's dimensions. This part consists of two steps:

Step 1: Identify the best orientation for each 3D box type considering only its base face to place the maximum number of boxes per horizontal layer. Each layer should consist only from boxes of the same type as shown in Fig 3.2 and Fig 3.3. The benefit of this step is storing box type orientation and the maximum number of each box type per layer within the horizontal dimensions; length and depth in order to speed up the next algorithm steps.

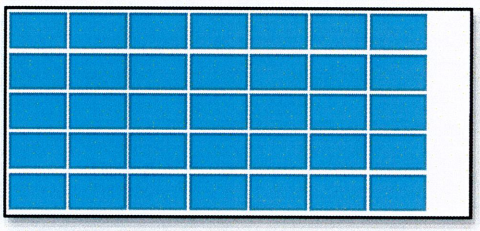


Figure 3.2: orientation option 0

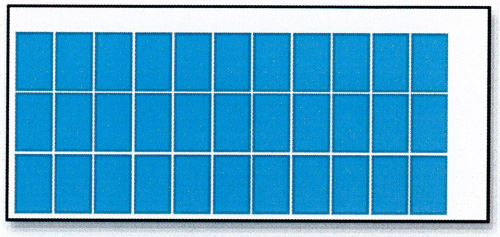


Figure 3.3: orientation option 1

The pseudo-code of this step is detailed in 2D-layers-candidates Algorithm 1:

Algorithm 1 2D-layer-candidates

Input: $D, L, d_i, l_i, i = 1 \ldots t.$

Output: hlc_i = maximum number of boxes per one horizontal layer for each box type i , $i = 1 \ldots t$

for $i = 1$ to t **do**

$$hlc_i = max\left(\left\lfloor \frac{L}{l_i} \right\rfloor \times \left\lfloor \frac{D}{d_i} \right\rfloor, \left\lfloor \frac{D}{l_i} \right\rfloor \times \left\lfloor \frac{L}{d_i} \right\rfloor\right)$$

return hlc_i.

end for

Step 2: Identify all possible solution candidates formed by combining layers of different box types that can vertically fit in a bin without exceeding its height. In this step, the total weight of boxes is not considered, neither the material within the box, and any box can fit in any bin. Therefore, some solution candidates would not be feasible and will be discarded further. The details of this step is illustrated in Figure 3.4 and summarized in Algorithm 2.

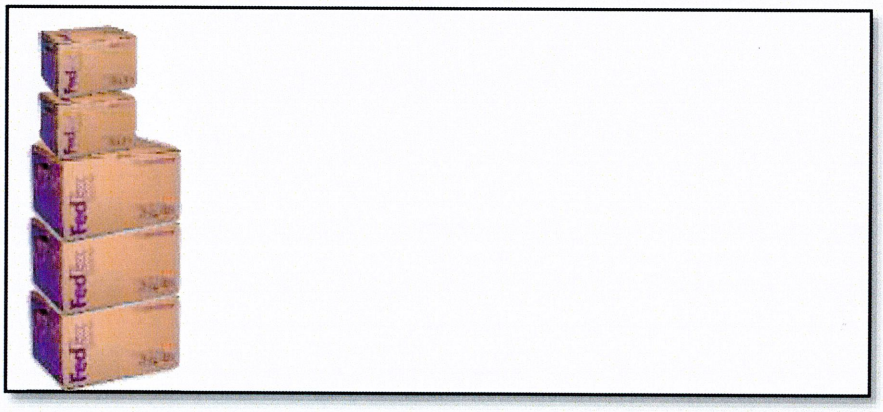


Figure 3.4: layers layout in a Bin (example: 3x25kg box +2x10kg box)

Algorithm 2 3D-solution-candidates

1: **Input:** $D, L, H, d_i, l_i, h_i,\ i = 1 \ldots t.$
2: **Output:** set of combined layers that can volumetrically fit within a bin
3: $S = \emptyset$
4: **for** $r_1 = 0$ to $\left\lfloor \frac{H}{h_1} \right\rfloor$ **do** // r_1 is the number of layers of boxes type 1
5: **for** $r_2 = 0$ to $\left\lfloor \frac{H}{h_2} \right\rfloor$ **do**
⋮
6: **for** $r_t = 0$ to $\left\lfloor \frac{H}{h_t} \right\rfloor$ **do**
7: **if** $(\sum_{i=1}^{t} r_i h_i < H)$ **then**
8: $S = S \bigcup \left(\bigcup_{i=1}^{t} r_i \text{ layers of boxes type } i \right)$
9: **end if**
10: **end for**
11: **end for**
12: **end for**
13: **return** S.

The second part of the proposed LBA-3DBPP algorithm focuses on placing the received boxes into a minimum number of bins. The algorithm will go through the unpacked boxes and check their numbers. Then, it will loop through the Layer solutions, starting with layers with highest volume wise bin utilization, to pack the maximum number of boxes within a bin respecting the bin's maximum weight. In addition, the algorithm at each iteration, will check the unpacked boxes total volume. If it is less than or equal the bins volume, the remaining unpacked boxes will be placed in the last bin. The pseudo code of the LBA-3DBPP is shown in Algorithm 3.

Note that W_{s_p} in line 17 is obtained by adding the weights of the $(n_{ip}/2)$ heaviest unpacked boxes type i and the weights of the $(n_{ip}/2)$ lightest unpacked boxes type i. The condition in this line ensures the

feasibility of the solution S_p in term of bin's maximum weight and helps in getting balanced bins in term of weights.

Algorithm 3 LBA-3DBPP

```
1:  Input: D, L, H, W, d_i, l_i, h_i, n_i, i = 1...t and w_k, k = 1...n.
2:  Output: m filled bins (m to be minimized).
3:  S = result of 3D-solution-candidates call
4:  Sort S in decreasing order of bin's volumetric utilization.
5:  m = 0
6:  accept S_p = true
7:  U: set of all received boxes not yet packed
8:  for i = 0 to t do
9:      k_i = 0 (k_i is the number of packed boxes type i)
10: end for
11: while U ≠ ∅ do
12:     for p = 1 to |S| do
13:         W_Sp = 0
14:         for i = 0 to t do
15:             n_ip = number of boxes type i in solution S_p
16:         end for
17:         if W_Sp > W then
18:             accept S_p = false
19:             check the next solution (line 12)
20:         end if
21:         for i = 0 to t do
22:             if n_ip > n_i − k_i then
23:                 accept S_p = false
24:                 check the next solution (line 12)
25:             end if
26:         end for
27:         if accept S_p = true then
28:             fill the bin B_m according to S_p
29:             U = U − {the packed boxes in B_m}
30:             m = m + 1
31:             go to line 11
32:         end if
33:     end for
34: end while
35: return m.
```

On Table 3.2 new notations used in the algorithms are explained:

Table 3.2: LBA-3DBPP algorithm notations.

Algorithm	Notation	Meaning
Algorithm 1	hlc_i	Horizontal layer candidate of i
Algorithm 2	S	Set of horizontal layers forming a solution
Algorithm 3	W_{S_p}	Total weight of boxes within a solution

If the number of box types and dimensions, and bin's dimension is the same for different number of box items, only phase 2 can be executed multiple times, to achieve the best solution, which saves the time of executing first part of the algorithm.

On another note, before proposing the LBA-3DBPP many attempts were done to propose Genetic Algorithm to solve the 3DBPP, but the execution time were reaching 7-8 hours for small problem size, which then diverted the research towards a Heuristic Algorithm.

3.5 Conclusion

The new proposed algorithm LBA-3DBPP consists of 2 phases, phase 1 consists of 2 steps. The first 2 steps of phase 1 are to be executed only 1 time for the same set of bin dimension and box types, the 2nd phase can be executed multiple times as the count of box items changes. The algorithm shows many advantages over other heuristic algorithms, specifically in the design to minimize the execution time.

Chapter Four

Experimental Results

4.1. Introduction
4.2. Implementation Parameters
4.3. Implementation
4.4. Experimental Results
4.5. Discussion
4.6. Conclusion

Chapter Four: Experimental Results

4.1 Introduction

This chapter shows an implementation of the proposed heuristic algorithm LBA-3DBPP, for two box types, and illustrates the implementation results, with comparison with previously proposed algorithms, showing promising results, and very fast execution time of the algorithm. The chapter starts with implementation parameters, then the implementation, followed by the experimental results, and finally the discussion of the experimental results.

4.2 Implementation Parameters

As the LBA-3DBPP algorithm can cater for various bin's dimension, and maximum weight, and can take variable number of box types, we have set the parameters for our implementation as per Table 4.1:

Table 4.1: LBA-3DBPP implementation parameters

Type	Depth	Height	Length	Maximum Weight
Bin	243.8 cm	178 cm	317.5 cm	6804 kg
Box A	40.16 cm	32.86 cm	25.88 cm	10 kg
Box B	54.8 cm	42.1 cm	33.5 cm	25 kg

The parameters are real data from FedEx, the bin size is one of the standard bins used for Air courier.

4.3 Implementation

The algorithm was implemented using Visual Studio 2017 (C# language), and MS SQL Server 2014 database, the bin, boxes parameters

and shipments details are stored on SQL tables, the full algorithm parts were implemented using C# language, and adopting MVC (Model, View, Controller) concept, to maintain the modulatory of the application. Only one function was implemented using SQL stored procedure on SQL Server, to take advantage of memory tables on SQL, which have boosts the performance of the application.

The application was tested on 64-bit Windows 10 Lenovo PC, with 8.0 GB DDR4 SDRAM, Intel® Core™ i7-6700 CPU @ 3.40GHz, 4 cores and 8 logical processors.

4.4 Experimental Results

The implementation of the first part (phase 1, step 1) of the algorithm and the setup for the box types is shown in Table 4.2:

Table 4.2: LBA-3DBPP algorithm box types results

Type	Orientation	Items per Layer
Box A	0	72
Box B	0	36

The Layers solutions generated also from the first part (phase 1, step 2) of the algorithm are shown in Table 4.3, it shows the layers based solutions candidates, i.e. to achieve highest layers based utilization (97.4%), we have to pack 4 layers of box type A (288 boxes), and 1 layer of box type B (36 boxes). These layers solutions will be reused for packing each bin, trying to pack the boxes based on the layers solution starting from the solution candidate with highest utilization to lowest.

Table 4.3: LBA-3DBPP algorithm Layers solutions

Solution ID	BoxA Layers	BoxB Layers	BoxA total per layer	BoxB total per layer	Utilization %
1	4	1	288	36	0.974
2	0	4	0	144	0.946
3	5	0	360	0	0.923
4	1	3	72	108	0.894
5	2	2	144	72	0.842
6	3	1	216	36	0.79
7	4	0	288	0	0.738
8	0	3	0	108	0.709
9	1	2	72	72	0.657
10	2	1	144	36	0.605
11	3	0	216	0	0.553
12	0	2	0	72	0.473
13	1	1	72	36	0.421
14	2	0	144	0	0.369
15	0	1	0	36	0.236
16	1	0	72	0	0.184

The second part of the algorithm is the part that will be repeated for any set of shipments. Multiple iteration has been tested, and the obtained results are compared with the lower bound *lb*, the results provided by the algorithm proposed by J. Kaabi et al. (2018), and also EB-AFIT algorithm proposed by Saraiva et al. (2015) that is also layers based approach, Table 4.4 shows the execution results:

Table 4.4: LBA-3DBPP experiment results

Number of items	LBA-3DBPP	Execution Time (seconds)	J. Kaabi et al. (2018) Bins	*lb* (volumetric)	EB-AFIT Bins	EB-AFIT Execution time (seconds)
100	1	0.010	1	1	1	0.034
500	3	0.057	3	3	3	22.835
1,000	5	0.107	6	5	5	180
2,500	13	0.350	15	11	12	2,820
5,000	26	0.837	30	22	24	21,600
7,500	39	1.843	46	33	36	72,000
10,000	52	2.177	60	44	-	N/A
25,000	131	5.787	153	109	-	N/A
50,000	261	17.230	305	219	-	N/A
75,000	392	28.310	457	328	-	N/A
100,000	522	48.160	609	437	-	N/A

As shown in Table 4.4, EB-AFIT algorithm shows better solution than LBA-3DBPP proposed algorithm in terms of number of bins, for mid-range problem, however, the execution time of the algorithm gets exponentially increased as the number of boxes increases, from 10,000 boxes and above, the execution time becomes unfeasible to experiment. While LBA-3DBPP execution time gets slightly increased as the number of boxes increases.

LBA-3DBPP shows better solutions as well than the algorithm proposed in J. Kaabi et al. (2018) in terms of number of bins used, starting from number of boxes equals 1,000 boxes and above, and as the number of boxes increases, the number of bins used are less in LBA-3DBPP, and with the advantage of short execution time.

For problems with number of boxes is 1,000 and less, LBA-3DBPP have an optimum solution that meets the lower bound (*lb*) solutions, but

as the number of boxes increases, the solution of LBA-3DBPP deviates from the *lb* solution by approximately %18, but taking into consideration that the *lb* is volumetric based, it wouldn't be necessarily be achieved taking into account the maximum allowed weight for the bin.

The example of 2,500 boxes has been thoroughly analyzed, in terms of the bins average volume and weight utilization, and weight balance of all the bins, and it has shown promising results as illustrated in Figure 4.1 and Figure 4.2 consecutively.

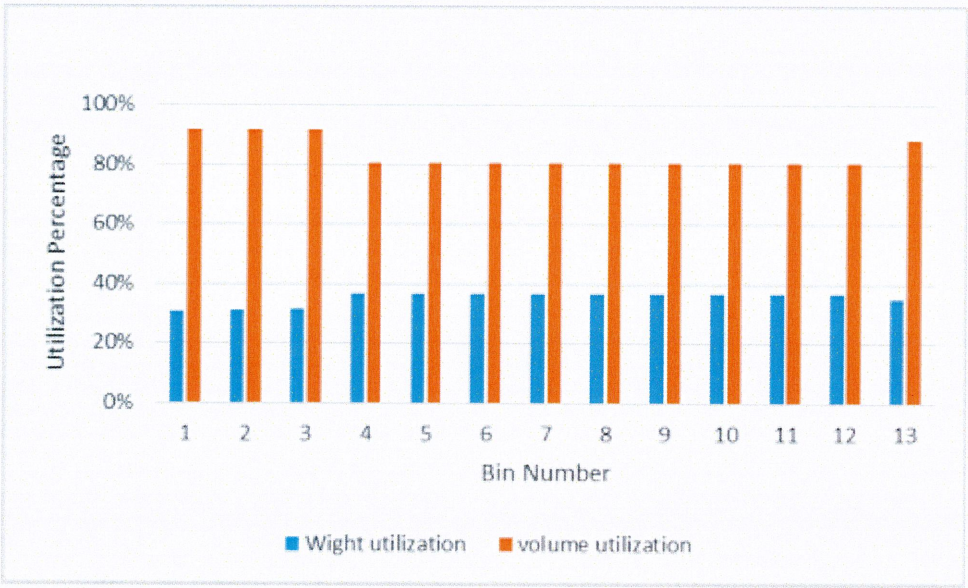


Figure 4.1: LBA-3DBPP volume and weight utilization results for 2,500 boxes

The volumetric utilization in Figure 4.1 of all the bins is very high, the waste of bin volume is minimum, and the weight utilization is between 30%-40%, as the volume utilization is higher for achieving minimum number of bins solution.

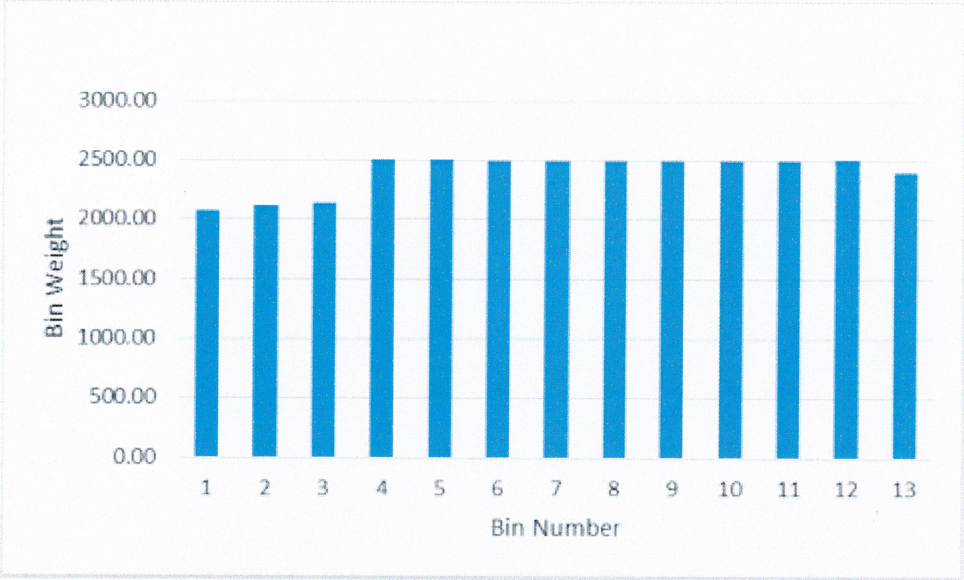


Figure 4.2: LBA-3DBPP 2,500 boxes solution bin's weight

Figure 4.2 shows minimal differences between the weights of bins, which meets the objective of loading the balance of all bins in the solution (minimize difference between the weight of each bin and the average weight).

4.5 Discussion

The new proposed algorithm LBA-3DBPP reached optimum solution for small problem set up to 1,000 item boxes on the experimental data, and has shown promising solution for large (more than 15 bins used) problem set. However, the main noticeable advantage over other algorithms was the execution time, as 100,000 boxes were packed completely in 48 seconds, and also for small sets up to 5,000 boxes the execution time is less than 1 second.

The objectives of minimizing number of bins, and spreading the load weight across the bins were both satisfied in the sample sets of data tested, volumetric utilization were above %90 and the deviation of bins weight from the average weight was minimal.

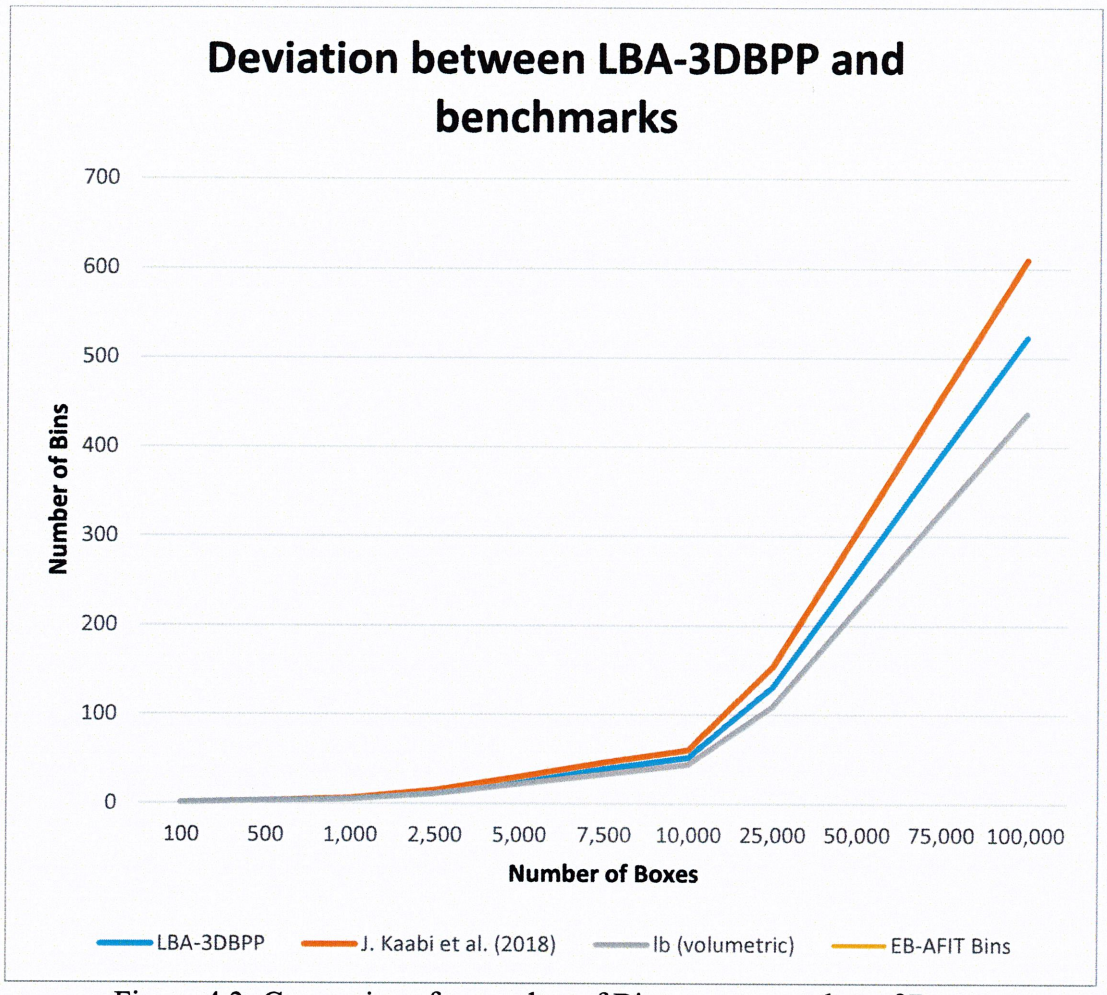


Figure 4.3: Comparison for number of Bins versus number of Boxes

The chosen implementation technologies played huge role in achieving the speed solutions, which makes the LBA-3DBPP a real advancement in the academic and industrial world, and the clean implementation of the algorithm makes ready to be commercial product to solve real world bin-packing problems.

4.6 Conclusion

The LBA-3DBPP algorithm shows promising results, and shows better results than the compared algorithms: J. Kaabi et al. (2018) and EB-AFIT. The algorithm satisfies the objectives of the research, and outperforms the compared algorithms in terms of execution time speed.

Chapter Five

Conclusion

5.1. Conclusion
5.2. Future work

Chapter Five: Conclusion

5.1 Conclusion

The bin-packing problem is an old problem and has been studied by many researchers over the years, and each has suggested a solution for the problem. However, there are many variations of bin-packing problems, depending on the objectives of the research. From the literature we've seen that even of the same objectives being investigated, still there are multiple solution approaches proposed to solve them.

Our research is contributing to the academic world with a novel algorithm that generates solutions in matter of seconds with optimum results for small problems, and promising solutions for large problem instances. The proposed algorithm LBA-3DBPP implemented, and compared with J. Kaabi etal (2018) proposed algorithm and have shown better solution in terms minimum number of bins and average bins weight, and also compared with EB-AFIT algorithm proposed by Saraiva et al. (2015), which have shown slightly more number of bins for large problems, but better bin's weight balance and far better execution speed. The maximum execution time for LBA-3DBPP algorithm was 48 seconds to pack 100,000 boxes of two box types.

The literature suggested that Genetic Algorithm have to be considered for large problems, however, if the execution speed has more priority than the number of bins and weight load balance, then LBA-3DBPP could be considered instead of Genetic Algorithms.

5.2 Future work

For future work the algorithm could be enhanced by considering the rotation of boxes in all direction and could consider layers with mixed

box types, which would make the problem more generic and applicable to wider range of real-world problems.

The proposed algorithm LBA-3DBPP could be tested on more heterogeneous box types, which will expand the applications of the algorithm.

More applications of the algorithm could be investigated to solve different real-world problems other than the bin-packing problem.

References

Babel, L., Chen, B., Kellerer, H., & Kotov, V. (2004). Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics, 143* (1), 238-251.

Bortfeldt, A., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research, 131* (1), 143-161.

Cao, J., Shi, Q., & Lee, D. H. (2008). A decision support method for truck scheduling and storage allocation problem at container. *Tsinghua Science & Technology, 13*, 211-216.

Chen, C. S., Lee, S. M., & Shen, Q. S. (1995). An analytical model for the container loading problem. *European Journal of Operational Research, 80*(1), 68-76.

Coffman Jr, E. G., Csirik, J., Rónyai, L., & Zsbán, A. (2008). Random-order bin packing. *Discrete Applied Mathematics, 156* (14), 2810-2816.

Dube, E., Kanavathy, L. R., & Woodview, P. (2006). Optimizing Three-Dimensional Bin Packing through simulation. *Proceedings of the Sixth IASTED International Conference on Modelling, Simulation, and Optimization, 1-7.*

Dyckhoff, H., & Finke, U. (1992). *Cutting and packing in production and distribution: A typology and bibliography*. Berlin, Germany: Springer Science & Business Media.

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

Elhedhli, S., Li, L., Gzara, M., & Naoum-Sawaya, J. (2011). A branch-and-price algorithm for the bin packing problem with conflicts. *INFORMS Journal on Computing, 23* (3), 404-415.

Falkenauer, E., & Delchambre, A. (1992, May). A genetic algorithm for bin packing and line balancing. In *Proceedings. 1992 IEEE International Conference on Robotics and Automation* (pp. 1186-1192). New York, NY: IEEE.

George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & OperationsResearch, 7* (3),147-156.

Gurbuz, M. Z., Akyokus, S., Emiroglu, I., & Guran, A. (2009). An Efficient Algorithm for 3D Rectangular Box Packing. *Applied Automatic Systems: Proceedings of Selected AAS*, 131-134.

Han, X., Peng, C., Ye, D., Zhang, D., & Lan, Y. (2010). Dynamic bin packing with unit fraction boxes revisited. *Information Processing Letters, 110* (23), 1049-1054.

Hopper, E., & Turton, B. (1998). Application of genetic algorithms to packing problems-a review. In P. K. Chawdhry R., Roy R., & K. Pant (Eds.), *Soft Computing in Engineering Design and Manufacturing* (pp. 279-288). London, UK: Springer-Verlag.

Huang, W., & He, K. (2009). A caving degree approach for the single container loading problem. *European Journal of Operational Research*, *196* (1), 93-101.

Kaabi, J., Harrath, Y., Bououdina, H. E., & Qasim, A. T. (2018, April). Toward Smart Logistics: A new Algorithm for a Multi-Objective 3D Bin Packing Problem In *Proceedings of the IET Smart Cities Symposium, 22-23 April 2018, Kingdom of Bahrain* (p. 1-5). London, UK: IET Digital Library.

Kang, J., & Park, S. (2003). Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, *147*(2), 365-372.

Kang, K., Moon, I., & Wang, H. (2012). A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem. *Applied Mathematics and Computation*, *219*(3), 1287-1299.

Kanna, S.R., Udaiyakumar, K.C., & Campus, R. (2017). A Complete Framework For Multi-Constrained 3D Bin Packing Optimization

Using Firefly Algorithm. *International Journal of Pure and Applied Mathematics, 114*(6), 267-282.

Leung, S.Y.S., Wong, W.K., & Mok, P.Y. (2008). Multiple-objective genetic optimization of the spatial design for packing and distribution carton boxes. *Computers & Industrial Engineering, 54*(4), 889-902.

Li, X., Zhao, Z., & Zhang, K. (2014, January). A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins. In *IIE Annual Conference Proceedings* (p. 2039). Institute of Industrial and Systems Engineers (IISE).

Liu, D. S., Tan, K. C., Huang, S. Y., Goh, C. K., & Ho, W. K. (2008). On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research, 190*(2), 357-382.

Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research, 48*(2), 256-267.

Mongeau, M., & Bes, C. (2003). Optimization of aircraft container loading. *IEEE Transactions on Aerospace and Electronic Systems, 39*(1), 140-150.

Öylek, A. (2015). Bicriteria Bin packing problem with deviation based objectives (doctoral dissertation, Middle East Technical University).

Pimpawat, C., & Chaiyaratana, N. (2001). Using a co-operative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem. In *Proceedings of the 2001 Congress on Evolutionary Computation, 2001* (Vol. 2, pp. 1197-1204). New York, NY: IEEE.

Piyachayawat, T., & Mungwattana, A. (2017, April). A hybrid algorithm application for the multi-size pallet loading problem case study: Lamp and lighting factory. In *4th International Conference on Industrial Engineering and Applications (ICIEA), 2017* (pp. 100-105). New York, NY: IEEE.

Rudy, J., & Żelazny, D. (2015). Multi-criteria 3-dimension bin packing problem. *Research in Logistics & Production*, *5*(1), pp.85-94.

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

Sheng, L., Xiuqin, S., Changjian, C., Hongxia, Z., Dayong, S., & Feiyue, W. (2017). Heuristic algorithm for the container loading problem with multiple constraints. *Computers & Industrial Engineering*, *108*, 149-164.

Sridhar, R., Chandrasekaran, M., & Page, T. (2016). Multi objective optimization of heterogeneous bin packaging using adaptive genetic approach. *Indian Journal of Science and Technology, 9*(48).

Sridhar, R., Chandrasekaran, M., Sriramya, C., & Page, T. (2017, March). Optimization of heterogeneous Bin packing using adaptive genetic algorithm. In *IOP Conference Series: Materials Science and Engineering* (Vol. 183, No. 1, pp. 1-12). Philadelphia, PA: IOP Publishing.

Takahara, S. (2006, October). A simple meta-heuristic approach for the multiple container loading problem. In *SMC'06. IEEE International Conference on Systems, Man and Cybernetics, 2006* (Vol. 3, pp. 2328-2333). New York, NY: IEEE.

Tavakkoli-Moghaddam, R., Makui, A., Salahi, S., Bazzazi, M., & Taheri, F. (2009). An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, *56*(1), 241-248.

Trivella, A., & Pisinger, D. (2016). The load-balanced multi-dimensional bin-packing problem. *Computers & Operations Research*, *74*, pp.152-164.

Wang, Z., Li, K. W., & Levy, J. K. (2008). A heuristic for the container loading problem: A tertiary-tree-based dynamic space

decomposition approach. *European Journal of Operational Research*, *191*(1), 86-99.

Wu, Y., Li, W., Goh, M., & de Souza, R. (2010). Three-dimensional bin packing problem with variable bin height. *European journal of operational research*, *202*(2), 347-355.

Appendix 1 – Finding Layers solutions source code

```
public class LayersPacking
{
    public int DataPrep()
    {
        BinPackingDataDataContext db = new BinPackingDataDataContext();

        // Find best orientation to maximize items per layer in Bin, for each
box type:

        var bins = (from c in db.Containers
                    select c).ToList();

        if (bins.Count != 1)
            return -1;

        var boxes = (from b in db.Boxes
                        select b).ToList();

        if (boxes.Count < 1)
            return -2;

        for (int i = 0; i < boxes.Count; i++)
        {
            double w1 = Math.Floor((float)(bins.First().bin_width /
boxes[i].box_wdith));
            double d1 = Math.Floor((float)(bins.First().bin_depth /
boxes[i].box_depth));
            double items1 = w1 * d1;

            double w2 = Math.Floor((float)(bins.First().bin_width /
boxes[i].box_depth));
            double d2 = Math.Floor((float)(bins.First().bin_depth /
boxes[i].box_wdith));
            double items2 = w2 * d2;

            int orientation = items1 > items2 ? 0 : 1;

            int items = (int)(items1 > items2 ? items1 : items2);

            (from p in db.Boxes
                where p.ID == boxes[i].ID
                select p).ToList()
                                .ForEach(x => { x.best_orientation =
orientation; x.items_per_layer = items; });

            db.SubmitChanges();
        }

        // Find the utilization % for each layers combination and store it in a
list

        float? binHeight = bins.First().bin_height;
        float? boxAHeight = boxes[0].box_height;
        int? boxAitemsPerLayer = boxes[0].items_per_layer;
        float? boxBHeight = boxes[1].box_height;
```

```
        int? boxBitemsPerLayer = boxes[1].items_per_layer;
        int[] maxItems = new int[boxes.Count];

        db.ExecuteCommand("TRUNCATE TABLE Layers;");

        for (int i=0; i < boxes.Count; i++)
            maxItems[i] = (int)Math.Floor((float)(bins.First().bin_height /
boxes[i].box_height));

        for(int a=0; a <= maxItems[0]; a++)
            for(int b=0; b <= maxItems[1]; b++)
            {
                float stack = (a * (float)boxAHeight) + (b *
(float)boxBHeight);
                if(stack <= binHeight)
                {
                    Layer layer = new Layer();
                    layer.boxA = a;
                    layer.boxB = b;
                    layer.boxAtotal = a * boxAitemsPerLayer;
                    layer.boxBtotal = b * boxBitemsPerLayer;
                    layer.utilization = (decimal)(stack / (float)binHeight);
                    db.Layers.InsertOnSubmit(layer);
                    db.SubmitChanges();
                }

            }

        return 0;
    }
}
```

Appendix 2 – bin-packing source code

```
// LBA-3DBPP algoirthm
public int FillBinsByLayers()
{
    bool MainFlag = true;
    int numberOfBins = 0;
    var tmp = db.ClearBins();

    var Container = (from c in db.Containers
                     select c).ToList();
    var BoxA = (from ba in db.Boxes where ba.ID == 1 select ba).ToList();
    var BoxB = (from bb in db.Boxes where bb.ID == 2 select bb).ToList();

    float containerVolume = (float)Container.First().bin_width *
(float)Container.First().bin_height * (float)Container.First().bin_depth;
    float boxAvolume = (float)BoxA.First().box_volume;
    float boxBvolume = (float)BoxB.First().box_volume;

    Log("Algorithm Start");
    while (MainFlag)
    {
        var Solutions = (from l in db.Layers
                         select l).OrderByDescending(x =>
x.utilization).ToList();
        var recordsA = (from a in db.Shipments
                        where a.bin_id == null
                        && a.box_type_id == 1
                        select a.ID).ToList();
        var recordsB = (from b in db.Shipments
                        where b.bin_id == null
                        && b.box_type_id == 2
                        select b.ID).ToList();

        bool itemsPacked = false;
        float totalBoxAvolume = recordsA.Count * boxAvolume;
        float totalBoxBvolume = recordsB.Count * boxBvolume;
        float totalBoxesVolume = totalBoxAvolume + totalBoxBvolume;

        if (totalBoxesVolume <= containerVolume )
        {
            MainFlag = false;
            numberOfBins++;
            var temp = db.addNewBin("Last Bin " + numberOfBins.ToString(),
recordsA.Count, recordsB.Count, 1, (decimal)(totalBoxAvolume /
containerVolume));
        }
        else
        {
            foreach (Layer solution in Solutions)
            {
                if (recordsA.Count >= solution.boxAtotal && recordsB.Count >=
solution.boxBtotal)
                {
                    numberOfBins++;
                    int temp = db.addNewBin("Bin " + numberOfBins.ToString(),
solution.boxAtotal, solution.boxBtotal, 0, (decimal)(solution.utilization));
                    if (temp == 0)
                    {
                        itemsPacked = true;
```

```
                    break;
                }
                else
                {
                    numberOfBins--;
                }
            }
        }

        if(itemsPacked == false)
        {
            Log("items not packed");
        }

    }

}
Log("Algorithm End");
Log("Number of Bins: " + numberOfBins.ToString());

return 0;
}
```

Appendix 3 – AddBin SQL stored procedure

```
-- =============================================
-- Author:           Jasim Hasan
-- Create date:   20.10.2018
-- Last Modified: 23.11.2018
-- Description:      Add new BIN and update shipments
-- =============================================
CREATE PROCEDURE [dbo].[addNewBin]
       @bin_number   varchar(30),
       @boxA         int,
       @boxB         int,
       @last_run     int,
       @utilization  decimal(18,3)
AS
BEGIN
       SET NOCOUNT ON;

       declare @boxA1 int, @boxA2 int, @boxB1 int, @boxB2 int,
@shipments_total_weight real, @bin_max_weight real;

       IF @last_run = 1
       BEGIN
              INSERT INTO Bins VALUES(@bin_number, @utilization, null);

              update Shipments
              set bin_id = IDENT_CURRENT('Bins')
              where bin_id is null;
              return 0;
       END;

       create table ##TempShipments
       (
              shipment_id   INT   NOT NULL INDEX ix1 NONCLUSTERED,
              shipment_weight real
       );

       truncate table ##TempShipments;

       IF @boxA > 0
       BEGIN
              set @boxA1 = @boxA / 2;
              set @boxA2 = (@boxA / 2) + (@boxA % 2);

              set ROWCOUNT  @boxA1;
              insert into ##TempShipments
              select ID, box_weight
              from Shipments where bin_id is null and box_type_id = 1 ORDER BY
box_weight;

              set ROWCOUNT  @boxA2;
              insert into ##TempShipments
              select ID, box_weight
              from Shipments where bin_id is null and box_type_id = 1 ORDER BY
box_weight desc;
       END;

       IF @boxB > 0
       BEGIN
              set @boxB1 = @boxB / 2;
```

```
                set @boxB2 = (@boxB / 2) + (@boxB % 2);

                set ROWCOUNT  @boxB1;
                insert into ##TempShipments
                select ID, box_weight
                from Shipments where bin_id is null and box_type_id = 2 ORDER BY
box_weight;

                set ROWCOUNT  @boxB2;
                insert into ##TempShipments
                select ID, box_weight
                from Shipments where bin_id is null and box_type_id = 2 ORDER BY
box_weight desc;
        END;

        set ROWCOUNT 0;

    -- check if total weight exceeds container's maximum weight
        select @bin_max_weight = bin_max_weight
        from Container;

        select @shipments_total_weight = SUM(shipment_weight)
        from ##TempShipments;

        IF @shipments_total_weight > @bin_max_weight
        BEGIN
                drop table ##TempShipments;
                return 1;
        END;

        INSERT INTO Bins VALUES(@bin_number, @utilization, null);

        update Shipments
        set bin_id = IDENT_CURRENT('Bins')
        WHERE bin_id is null
        and ID in (select shipment_id from ##TempShipments);


        drop table ##TempShipments;
        return 0;
END
```

الملخص

تم بحث موضوع مشكلة تعبئة حاويات الشحن بشكل مستفيض في السابق، ولا زال رهن البحث حتى وقتنا الحاضر للوصول للحل الأمثل. ينقسم الموضوع إلى عدة مشكلات يبحثها الباحثون، كل مشكلة تعتمد على الهدف من وراء البحث، ولكل مشكلة قدم الباحثون عدة حلول. في هذا البحث نحقق في مشكلة تعبئة الحاويات ثلاثية الأبعاد ونقدم خوارزمية مبتكرة لحل المشكلة، حيث نأخذ بعين الاعتبار الحجم والوزن كقيود للمشكلة، للوصول للحد الأدنى من الحاويات لتعبئة صناديق من أحجام مختلفة. كما يجب أن تكون الحاويات المستخدمة متساوية تقريباً قدر المستطاع من حيث وزن الشحنة. الحل المقترح يعتمد على طريقة التعبئة بالطبقات، حيث يتم تعبئة الحاوية باستخدام طبقات أفقية، الحل المتقرح يقدم نتائج مشجعة، كما أنه بالامكان تطبيقه بمرونة على أنواع مختلفة من مشكلة تعبئة الحاويات.

هذا البحث يبحث في الأدب البحثي عن عدة أنواع من مشكلات تعبئة الحاويات، ويذهب في شرح خطوات الخوارزمية المقترحة، ثم يقدم تطبيقاً عملياً على مثال حي من الواقع العملي لصناعة الشحن. تم مقارنة النتائج بنجاح مع الحد الأدنى النظري وأبحاث منشورة سابقة تقدم حلاً لنفس المشكلة.

جامعة البحرين

كلية تقنية المعلومات



خوارزمية مبتكرة لحل مشكلة تعبئة الحاويات ثلاثية الأبعاد متعددة الأهداف

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير في تقنية المعلومات

إعداد

جاسم حسن إبراهيم

الرقم الجامعي: 19990385

إشراف

د. جيهان كعبي

(أستاذ مساعد)

د. يوسف حراث

(أستاذ مساعد)

جامعة البحرين

مـملكة البـحرين

فبراير، 2019م