A heuristic and GRASP algorithm for three-dimensional multiple bin-size bin packing problem based on the needs of a spare-part company

Ali Shoja Sangchooli*

Technology Development Institute (ACECR),
Sharif Branch, Tehran, 14155-4364, Iran
Email: Shoja.email@gmail.com
*Corresponding author

Seyed Mehdi Sajadifar

Industrial Engineering Department,
Faculty of Technical and Engineering,
University of Science and Culture,
Tehran, Iran
Email: sajadifar@usc.ac.ir

Abstract: The three-dimensional multiple bin-size bin packing problem (3D-MBSBPP) has many practical applications in the logistic problems such as warehouse management, transportation planning and container loading. An efficient solution to the problem can have significant effects on reducing the transportation costs, improving the status and increasing the productivity and profitability of the companies. The mathematical models proposed for this problem are few in number and can, therefore, be expanded and improved much further. In this paper, based on the needs of a spare-part company in Iran, we developed a mathematical model for 3D-MBSBPP. This model takes into account the rotation of the boxes and the maximum weight constraint of the bins and then, we used a new heuristic and a GRASP algorithm for solving the model. The obtained answers in comparison with the exact method, confirm the speed and efficiency of the proposed algorithms, especially in solving the large-scale and real-sized problems.

Keywords: 3D MBSBPP; container loading; logistic; heuristic algorithm.

Reference to this paper should be made as follows: Sangchooli, A.S. and Sajadifar, S.M. (2021) 'A heuristic and GRASP algorithm for three-dimensional multiple bin-size bin packing problem based on the needs of a spare-part company', *Int. J. Services and Operations Management*, Vol. 38, No. 1, pp.73–91.

Biographical notes: Ali Shoja Sangchooli is a PhD candidate in Industrial Engineering at the Technology Development Institute (ACECR), Sharif University branch. His research interests include heuristic methods, software development, logistics, and management.

Seyed Mehdi Sajadifar is currently an Assistant Professor in the Department of Industrial Engineering at the University of Science and Culture. He received his BSc, MSc and PhD degrees from the Sharif University of Technology in

Industrial Engineering in 1994, 1996 and 2008, respectively. His research interests include inventory management and control, production scheduling and stochastic processes.

1 Introduction

The problem of packing boxes, or items, into the bins, or containers, is one of the most important and most common problems related to logistics. Of critical significance to this problem are the sizes and dimensions of boxes and bins. In this problem, the size and dimensions of the boxes and bins have always been considered critical matters. Whereas this problem can be analysed in one, two or three-dimensional modes, this study examined its three-dimensional mode in order to evaluate it in the most realistic and practical conditions. The problem is known as three-dimensional multiple bin-size bin packing problem (3D-MBSBPP). The problem has many practical and operational applications in the field of transportation and logistics and its optimisation will have many benefits such as the effective use of empty and unoccupied spaces and outstanding saving on distribution costs for the logistics system. The cost of distribution plays a vital role in the determination of product price (Manimaran and Selladurai, 2014). also it is an important part of green supply chain models (Kurian et al., 2018; Loni et al., 2018) Due to the specificity and complexity of the problem, it remains an under-explored issue (Paquay et al., 2017) Since the conditions and objectives of real-life problems vary considerably from one another, similar problems have rarely been addressed by the few available studies. In this study, the problem examined is a research in the central depot of a great spare parts company in Iran. The orders are collected throughout the day and at the end of the day, they are placed in the bins of different sizes. In order to evaluate and optimise this problem, the researchers designed a mathematical optimisation model, aimed at minimising the costs of used bins. This was done considering the conditions, such as the weight limit of the bins and the maximum number of the available bins. The model was then solved through a heuristic algorithm. Intending to further improve the answers, the researchers also employed a greedy randomised adaptive search procedure. The resulting algorithms were next implemented in a software package and the results from these two algorithms were compared with those of the GAMS software.

The structure of this study is in this way that in Section 2, a comprehensive review of the literature and studies on the subject and problem will be conducted. In Section 3, the problem and its modelling will be described and in Section 4, the heuristic algorithm will be explained. In Section 5, GRASP algorithm will be presented. Section 6 will provide the computational results obtained for the case study as well as the evaluation of the efficiency of the presented solving methods. Finally, the overall conclusion of this study will be presented in Section 7.

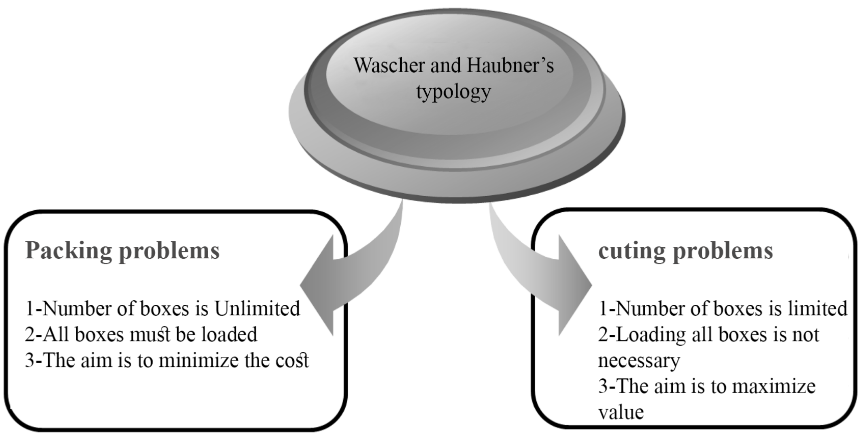
2 Literature review

Cutting and packaging problems are among the problems broadly discussed in the literature. Three reasons can be enumerated for the broadness of the number of studies in this area. The first reason is the widespread presence of these real-life problems in such areas as manufacturing, logistics and supply chain, including packaging of orders in depots, scheduling (Ramasubramaniam et al., 2010; Reisi-Nafchi and Moslehi, 2015) financial management (Song et al., 2018) and even computer science (Baker et al., 2018; Zhao et al., 2018; Mohiuddin et al., 2019). The second reason regards the complexity and severity of these problems, attracting the attention of researchers who seek accurate solving methods, along with those who advocate heuristic methods. In turn, the third reason concerns the flexibility of, as well as the striking differences between, cutting and packing problems in the real-world, such that any small change in the target function or in the cutting-packing restrictions leads to the definition of new problems with different structures (Dyckhoff, 1990).

The related literature currently introduces two typologies in the domain of cutting and packing problems. The first typology was presented by Dyckhoff (1990). Wäscher et al. (2007) pointed some of the weaknesses of this typology and presented their modifications in this regard in the form of an improved typology. In their typology, all cutting and packaging problems are divided into two general categories of output maximisation problems, i.e., cutting and backpack problems and input minimisation problems, that is, boxes packing problems. The characteristics of these categories are shown in Figure 1. The problem, studied in this paper, is the 3D-MBSBPP. For more information on the classification and the previous studies, readers are referred to article Bortfeldt and Wäscher (2012). The number of studies on the MBSBPP are few in number, even in one and two-dimensional spaces; only 12 articles had been published in this area by 2012 (Bortfeldt and Wäscher, 2012). Moreover, it should be stated that most of these studies have mainly addressed one or two-dimensional problems (Pisinger and Sigurd, 2005; Wei et al., 2013). In addition, most solving methods have, to date, been developed for one and two-dimensional problems and cannot easily be applied to a three-dimensional space, in which case an inferior answer, namely one in which most of the bin spaces are left empty, may result. As such, it is necessary to precisely evaluate and analyse three-dimensional problems with recourse to special methods designed for solving them (Crainic et al., 2009).

Ertek and Kilic (2006) presented the first article dealing directly with a 3D-MBSBPD analysis. To solve the problem, they proposed three methods including a heuristic greedy algorithm, a beam search method and a tree search method. Alvarez-Valdes et al. (2012) proposed an algorithm for solving MBSBPP in two-dimensional and three-dimensional spaces. Using this algorithm by means of a constructive method produces a number of answers. The best answers obtained from algorithms are next combined through a path re-linking method. In a follow-up study, Alvarez-Valdes et al. (2015) presented new lower bounds for the answers to their previous two- and three-dimensional problems. Alvarez-Valdes et al. (2015) also presented new lower bounds for the answer of their previous two-dimensional and three-dimensional problems. These lower bounds are derived from the formulation of integer programming for the obtained relaxed problem by a series of logical considerations, to a great extent, can be improved. Li and Zhang (2015) modelled a generalised three-dimensional packaging with heterogeneous bins in the form of a complex integer linear programming and solved a mixed evolutionary algorithm with the aim of minimising the empty space existing in the bins. In another study, Paquay et al. (2016) implementing a complex integer program with the aim of minimising unfilled space, evaluated and optimised the 3D-MBSBPP for packaging and loading containers for air transportation. In their presented model, relevant restrictions on air transportation, such as stability and fragility of boxes, were also considered (Paquay et al., 2016, 2017, 2018). Wu et al. (2017) also considered a three-dimensional, irregular (non-cubic), bin packing problem and provided a three-stage heuristic algorithm for its solution. In yet another study, Baazaoui et al. (2017) explained a specific application of the 3D-MBSBPP in an industrial company where the space inside bins was supposed to be assigned to boxes with guillotine cuttings. To deal with this, they presented an upper restriction on the basis of complex integer linear programming. Li et al. (2017), using the internet of things and information technology tools, presented a scenario for packing goods in an online store. The scenario also involved robots automatically filling bins. In one of the latest studies in this area, Hu et al. (2018) introduced a new type of three-dimensional bin packing problem, in which a number of cuboid-shaped boxes had to be put in a large bin one after another, in an orthogonal fashion. The objective was to minimise the surface area of the bin by finding the best way to place these boxes inside it.

Figure 1 Two main categories of Wäscher et al.'s (2007) typology



In the present study, the researchers, aiming to minimise total bin costs, modelled and optimised the 3D-MBSBPP in a large distribution company. The solving time of the problem was critical, and to be operational, it was expected to give a quick answer to the operator. The number of bins available was quite dynamic and had to be considered when solving the problem. Some boxes could not be rotated and each bin could withstand a limited weight. The objective function was to minimise the cost of using bins. According to the literature, considering such constraints as the maximum number of bins and the maximum tolerable weight of bins and the ability to rotate boxes by mathematical programming and to solve the resulting model through a heuristic, as well as a new greedy algorithm are the features that distinguish this study from previous studies in this area. The limitation in the number of each bin type makes the problem a practical one, since some bin types may not be available during certain times of the year. The large number of orders, the noticeable distances between the company headquarters and its clients and the company's commitment to deliver the orders in the shortest time are among the reasons why speed matters in this problem. Moreover, the fact that the orders are large in number causes the slightest optimisation of packing to greatly decrease the company's annual costs.

3 Problem statement and mathematical modelling

3.1 Problem statement

The present study was conducted based on the needs of a large distribution company in Iran. The central depot of this company, which receives the orders online from its nationwide agents, should respond to these orders as quickly as possible. For packing each order, experienced staff speculated the size, number and type of bins needed. However, in many cases, most parts of the bins remained empty. This remaining empty space alongside the necessary reloading due to the wrong estimations of the staff resulted in significant costs that the company intended to minimise. In regard to the high volume of the daily work, it was essential to design a quick methodology to obtain reasonable answers to this problem.

In Figures (2) and (3), samples of the boxes and bins used in this study are displayed, respectively.

Figure 2 Box sample

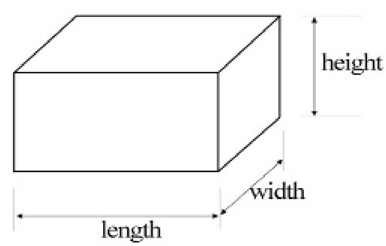
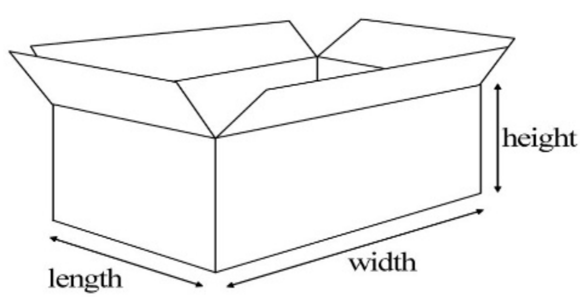


Figure 3 Bin sample for packing boxes



3.2 Model notations

We assume that the edges of the boxes parallel to the edges of the bin. The boxes can be rotated in six ways. The object is to find a solution, without overlapping, of all the boxes into the bins at minimum cost. Every box fits at least into one bin type, as otherwise no solution exists. The notations of the designed mathematical model including the sets, parameters and decision variables are given in Table 1.

Table 1 Notations of the mathematical model of the problem

Sets	
$i \in I, j \in J$	Set of boxes
$k \in K$	Set of bins
Parameters	
pl_i	Primary length of the box i
pw_i	Primary width of the box i
ph_i	Primary height of the box i
wt_i	Weight of box i
L_k	Length of bin k
W_k	Width of bin k
H_k	Height of bin k
C_k	Cost of bin k
WT_k	Maximum tolerable weight of bin k
Decision variables	
G_k	If bin k is used then 1, otherwise it is zero
f_{ik}	If box i is placed in bin k then 1, otherwise it is zero
x_i, y_i, z_i	Non-negative variables relate to the geometric position of the box i (the coordinates of the left-right behind point) in the bin in which it is placed
l_i, w_i, h_i	The variables related to the dimensions of the box i, which are determined according to its current rotate status and in the non-rotational mode are equal to the primary dimensions.
s_{ij}	Represents the geometric position of the two boxes respect to each other, and if the box i is in the left side of the box j (means $x_i + l_i \leq x_j$) then 1 and otherwise it is zero
b_{ij}	Represents the geometric position of the two boxes respect to each other, and if box i is behind the box j ($y_i + w_i \leq y_j$) then 1 otherwise it is zero
p_{ij}	Represents the geometric position of the two boxes respect to each other, and if box i is under the box j ($z_i + h_i \leq z_j$) then 1 otherwise it is zero

3.3 Mathematical model

According to the definitions of the sets, parameters and decision variables, the mathematical model of the problem, with its objective function being to minimise the costs of the bins [see equation (1)], is shown below.

$$z = \min \sum_{k=1}^{m} C_k G_k \tag{1}$$

s.t.

$$\sum_{k=1}^{m} f_{ik} = 1 \qquad i \in \{1, \ldots, n\} \tag{2}$$

$$f_{ik} \leq G_k \qquad i \in \{1, \dots, n\}, k \in \{1, \dots, m\} \tag{3}$$

$$\sum_{i=1}^{n} f_{ik} * wt_{ik} \leq WT_k \qquad k \in \{1, \dots, m\} \tag{4}$$

$$\begin{aligned} & p_{ji} + p_{ij} + b_{ji} + b_{ij} + s_{ji} + s_{ij} + (1 - f_{jk}) + (1 - f_{ik}) \geq 1 \\ & i, j \in \{1, \dots, n\}, i < j, k \in \{1, \dots, m\} \end{aligned} \tag{5}$$

$$M(1 - s_{ij}) \geq x_i - x_j + l_i \quad i, j \in \{1, \dots, n\} \tag{6}$$

$$M(1 - b_{ij}) \geq y_i - y_j + w_i \quad i, j \in \{1, \dots, n\} \tag{7}$$

$$M(1 - p_{ij}) \geq z_i - z_j + h_i \quad i, j \in \{1, \dots, n\} \tag{8}$$

$$x_i \leq L_k - l_i + (1 - f_{ik}) * M \quad i \in \{1, \dots, n\}, k \in \{1, \dots, m\} \tag{9}$$

$$y_i \leq W_k - w_i + (1 - f_{ik}) * M \qquad i \in \{1, \dots, n\}, k \in \{1, \dots, m\} \tag{10}$$

$$z_i \leq H_k - h_i + (1 - f_{ik}) * M \qquad i \in \{1, \dots, n\}, k \in \{1, \dots, m\} \tag{11}$$

$$x_i, y_i, z_i, l_i, w_i, h_i \geq 0 \qquad i \in \{1, \dots, n\} \tag{12}$$

$$G_k, f_{ik}, s_{ij}, b_{ij}, p_{ij} \in \{0, 1\} \qquad i, j \in \{1, \dots, n\}, k \in \{1, \dots, m\} \tag{13}$$

In the mathematical model above, constraint (2) implies that each box is supposed to be placed in one bin. Constraint (3) implies that if at least one box (like *i*) is placed in one bin (like *k*), then $G_k = 1$ and it means that bin has been used. Constraint (4) states that the total weight of boxes (*wt*) placed in a bin cannot exceed the maximum tolerable weight of that bin (*WT*). Constraint (5) indicates that the boxes do not overlap if $f_{ik} = 1$ and $f_{jk} = 1$, means that boxes *i* and *j* are placed in bin *k*, then to avoid interference, these boxes need to be placed either on top of each other or alongside together, which means one of the modes of $s_{ik} = 1$, $s_{ji} = 1$, $b_{ij} = 1$, $b_{ji} = 1$, $p_{ij} = 1$ or $p_{ji} = 1$. In other words, only if all of the variables are zero, the overlapping (interference) happens. Therefore, in order to avoid overlapping, constraint (5) were defined. Constraints (6), (7) and (8) relating to different geometry situations of two boxes respect to each other were respectively defined to assign value to the variables for the left, back or bottom side of the first box with respect to the second box. Constraints (9), (10) and (11), in turn, indicate that box is expected to be placed in the bin's physical range. Finally, constraints (12) and (13) define positive integer and binary variables of the model, respectively.

For the purposes of this study, the ability to rotate boxes was also considered a significant and highly applicable factor in modelling. A cube can have six rotation modes. For the rotation modelling, we first defined a Q matrix as the rotation matrix in equation (14):

$$\begin{pmatrix} l_i \\ w_i \\ h_i \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \begin{pmatrix} pl_i \\ pw_i \\ ph_i \end{pmatrix} \tag{14}$$

As can be seen below, this equation was converted into equations (15), (16) and (17) through matrix multiplication. These equations can be used to calculate the new dimensions of the box or the secondary dimensions after the box is rotated. The six rotation modes and the corresponding matrix are provided in Table 2.

$$l_i = Q_{11}^i * pl_i + Q_{12}^i * pw_i + Q_{13}^i * ph_i \qquad i \in \{1, \ldots, n\} \tag{15}$$

$$w_i = Q_{21}^i * pl_i + Q_{22}^i * pw_i + Q_{23}^i * ph_i \qquad i \in \{1, \ldots, n\} \tag{16}$$

$$h_i = Q_{31}^i * pl_i + Q_{32}^i * pw_i + Q_{33}^i * ph_i \qquad i \in \{1, \ldots, n\} \tag{17}$$

The rotation matrix (Q) should have only one element with a value equal to 1 in each row and in each column, which means that each box should have exactly one length, one width and one height after the rotation. Therefore, matrix elements should satisfy these constraints. Equations (18) and (19) represent these constraints.

$$Q_{d1}^i + Q_{d2}^i + Q_{d3}^i = 1 \qquad d \in \{1, 2, 3\} \tag{18}$$

$$Q_{1d}^i + Q_{2d}^i + Q_{3d}^i = 1 \qquad d \in \{1, 2, 3\} \tag{19}$$

By summing the indices of all variables, the number of variables of the model is obtained from equation (20). Similarly, by summing all indices related to all constraints, the number of constraints of the problem model is obtained from equation (21).

$$3n^2 + mn + 12n + m + 1 \tag{20}$$

$$3n^2 + mn^2 + 3mn + 7n + 1 \tag{21}$$

In the presented mathematical model, it was assumed that m bins exist and there was only one type of each one. It was also supposed that was equal to the distinct types of bins. If there is no limitation on the number of any type of the used bins, a maximum of *n* bins are required from each type of bins. Therefore, the total number of the bins required is obtained from equation (22).

$$m = nm' \tag{22}$$

Table 2 Six different rotation modes and corresponding matrices

Rotation matrix	*Status code*	*Rotation matrix*	*Status code*
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	0	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	3
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	1	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$	4
$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	2	$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	5

For example, in Table 3, the number of the bin types was fixed at 8, and the number of variables and constraints of the problem was obtained from the number of the different boxes. The number of the variables and constraints highlights the difficulty of the problem, even for small sizes.

Table 3 Number of variables and constraints of the problem in different box counts (n)

8	*8*	*8*	*m'*
40	24	8	N
320	192	64	M
18,401	6,817	865	Variables
555,481	126,313	5,881	Constraints

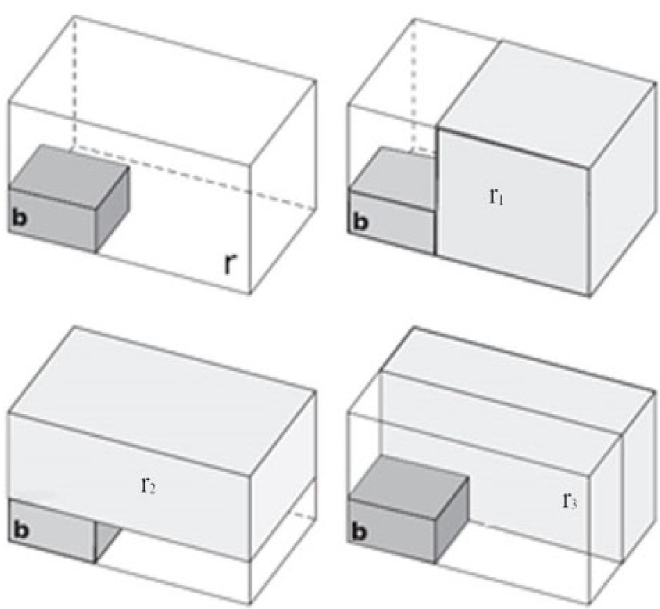
4 The proposed heuristic algorithm

What follows is a description of the proposed algorithm and a step-by-step solution of a test problem.

4.1 Algorithm explanation

Heuristic algorithms provide acceptable solutions within a reasonable period, hence they have high efficiency for solving hard problems. In relation to the 3D-MBSBPP, the heuristic methods are very practical in cases where the number of the boxes and bins is high and arriving at quick answers is a critical factor.

Figure 4 Empty maximal spaces r_1, r_2 and r_3



In this method, the concept of empty maximal spaces is used for placing boxes in bins. The empty maximal spaces refer to the largest empty spaces that appear after placing each box inside the bin. Put differently, empty maximal spaces are the largest remaining rectangular spaces in the bin once a box is put in it. In three-dimensional space, if a box is placed in the corner of a three-dimensional rectangular space, a maximum of three empty maximal spaces appear (Figure 4). The placement procedure is a mechanism in which an empty maximal space and a box are selected. Next, the selected box is placed in the chosen empty maximal space. After placing this box, the empty maximal space is used in full or in part, and this is done to update the empty maximal spaces.

The proposed algorithm for solving the problem includes the following steps:

Step 1 The types of the available bins are listed according to the cost-to-volume ratio (cost per unit volume). The bin with the lowest cost per unit volume is placed at the top of the list. If this value (cost per unit volume) is equal for two non-identical bins, the bin with the largest dimension is put higher. If the largest dimension of two bins is equal, the next dimension of the bins is considered. Likewise, in the case of equality, the third dimension is taken into account.

Step 2 The boxes are arranged according to their volume (from the largest volume to the smallest volume), and the box with the largest volume is put at the top of the list. If the volumes of two non-identical boxes are equal, their dimensions are considered in order to prioritise them. This list is completed in the same stepwise manner as that for the bins.

Step 3 The first displaced box is selected from the list of boxes. If there are not any boxes, the problem solving ends.

Step 4 If there are no empty maximal spaces available, the algorithm proceeds with Step 7. Otherwise, the empty maximal spaces are arranged. Among these empty spaces, the smallest one is put at the top of the list. If the spaces have the same volumes, their dimensions are taken into consideration. If the dimensions are the same, the space produced in an earlier step is prioritised. Also, if the spaces are produced in the same step, the space whose bottom-left hand side corner point is closer to the bottom-left hand side corner of the bin is prioritised.

Step 5 The aim is to place the selected box into the first empty space from the list of the spaces created in Step 3. That the total weight of the boxes placed inside the bin does not exceed the maximum bin weight is also considered. If the weight limit is satisfied but box placement is not possible because of geometric limitations, placement is tried by rotating the box inside this space, and if the box cannot be inserted in any rotation states, the next space is examined. If an empty maximal space is found where the box can be placed, the algorithm moves to the next step, but if there is no space or the box cannot be placed in any of the spaces, Step 7 begins.

Step 6 The selected box is placed in the bottom-left corner and behind the empty selected space, and the selected empty space is removed from the list of spaces. By doing this and placing this box, a maximum of three new empty maximal spaces are created and placed in the list of spaces. Then, the algorithm moves to Step 3.

Step 7 The aim is to decide whether or not to enter a new bin. Since the number of the boxes remaining can be much less than the volume of the bin, it is likely that we enter a big bin for a small number of boxes, and in so doing, increase the cost. In an attempt to avoid this, an investigation is done as to whether or not a bin with the lowest cost, one whose volume is more than the total volume of the boxes left, exists. If existing, this bin is added to the problem and the algorithm proceeds with the next step; otherwise, a new bin with dimensions larger than the box size is requested from the lower parts of the list. If the number of used bins of this type is fewer than the available number, Step 8 begins, otherwise the next bin in the list is checked. If there are not any other bins, the algorithm cannot obtain a feasible solution to the problem.

Step 8 An empty space equal to the dimensions of the new bin and with the coordinates (0, 0, 0) is added to the list of spaces. The algorithm moves to Step 4.

4.2 Solving a test problem

To better understand the process of the heuristic method, we solved an example test problem. Table 4 shows the characteristics of the boxes available in this problem and Table 5 displays the characteristics of the existing bins. Also, the order arrived at in Steps 1 and 2 of the algorithms can be observed in the last column of these tables. In turn, Table 6 shows the algorithm steps. Additionally, Figures 5 to 9 illustrate the pictures of box placement in the bins.

Table 4 Characteristics of boxes

ID	*Length*	*Width*	*Height*	*Weight*	*Volume*	*Order*
1	65	55	65	230	232,375	1
2	40	100	40	90	160,000	2
3	65	50	45	80	146,250	3
4	80	25	45	30	90,000	4
5	50	40	40	20	80,000	5

Table 5 Characteristics of bins

ID	*Length*	*Width*	*Height*	*Cost*	*Tolerable weight*	*Volume*	*Available count*	*Price to volume*	*Order*
1	105	54	53	400	300	300,510	1	0.133%	4
2	105	47	45	250	220	222,075	1	0.113%	3
3	153	105	105	1,200	1,690	1,686,825	1	0.071%	1
4	105	105	71	800	780	782,775	1	0.102%	2

Table 6 Problem-solving steps

Description of the relevant step	*Bin*	*Box*	*Relevant step*	*Solving step*
Forming the ordered list of bins	-	-	Step 1	1
Forming the ordered list of boxes	-	-	Step 2	2
Selecting the first non-placed box from the list	1	-	Step 3	3
Sorting empty maximal spaces	1	-	Step 4	4
Selecting a bin to enter the problem	1	4	Step 7	5
Creating an empty space for a new bin	1	4	Step 8	6
Sorting empty maximal spaces	1	-	Step 4	8
Placing the box in empty maximal space	1	4	Step 5	9
Creating new spaces	1	4	Step 6	10
Selecting the first non-placed box from the list	2	-	Step 3	11
Sorting empty maximal spaces	2	-	Step 4	12
Placing the box in empty maximal space	2	4	Step 5	13
Creating new spaces	2	4	Step 6	14
Selecting the first non-placed box from the list	3	-	Step 3	15
Sorting empty maximal spaces	3	-	Step 4	16
Placing the box in empty maximal space	3	4	Step 3	17
Creating new spaces	3	4	Step 6	18
Selecting the first non-placed box from the list	4	-	Step 3	19
Sorting empty maximal spaces	4	-	Step 4	20
Placing the box in empty maximal space	4	4	Step 5	21
Creating new spaces	4	4	Step 6	22
Selecting the first non-placed box from the list	5	-	Step 3	23
Sorting empty maximal spaces	5	-	Step 4	24
Placing the box in empty maximal space	5	4	Step 5	25
Selecting a bin to enter the problem	5	2	Step 7	26
Creating new spaces for the new bin	5	2	Step 8	27
Sorting empty maximal spaces	5	-	Step 4	28
Placing the box in empty maximal space	5	2	Step 5	29
Selecting the first non-placed box from the list (finish)	-	-	Step 3	19

Table 7 Final solution

Bin	*Box*	*Rotate status*	*X*	*Y*	*Z*
4	1	0	0	0	0
4	2	0	65	0	0
4	3	0	0	55	0
4	4	1	0	55	45
2	5	0	0	0	0

Note: Position of boxes in bins.

Figure 5 First box placement (see online version for colours)

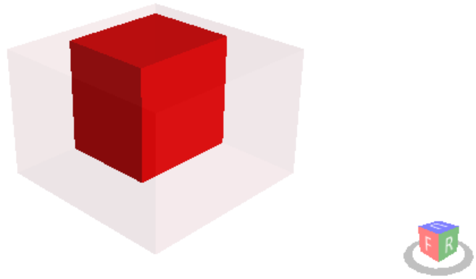


Figure 6 Second box placement (see online version for colours)

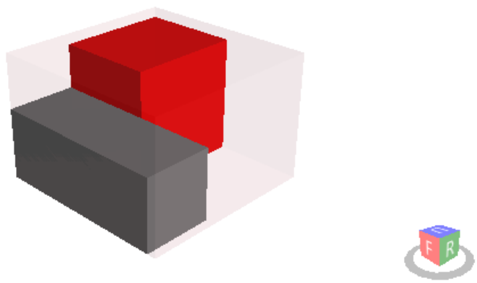


Figure 7 Third box placement (see online version for colours)

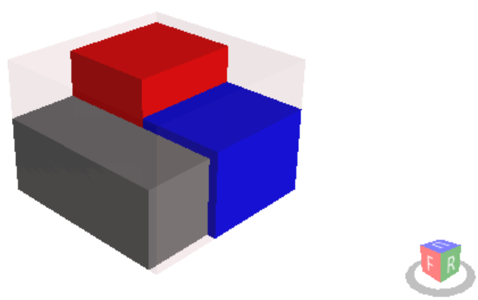


Figure 8 Fourth box placement (see online version for colours)

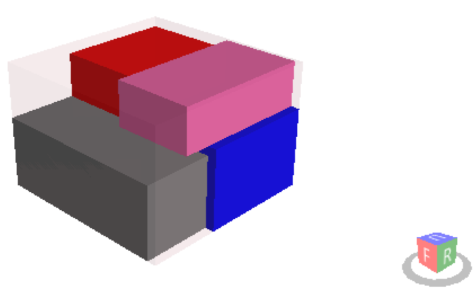
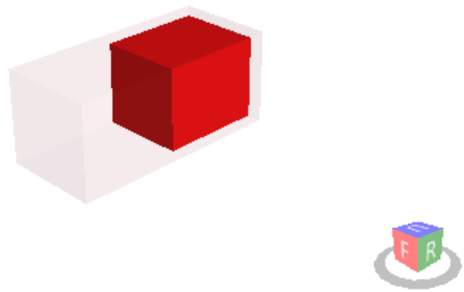


Figure 9 Fifth box placement (see online version for colours)



5 Greedy algorithm

The meta-heuristic GRASP method, a repetitive meta-heuristic procedure for solving complex optimisation problems was first presented by Feo and Resende (1995). The origins of this algorithm can be traced back to previous methods, such as random multi-start local search and semi-greedy heuristic algorithms. In this algorithm, the positive features of the greedy and randomly generating answer method are considered. The key to the GRASP algorithm is greedy generating methods. Each repetition of the GRASP algorithm consists of two steps. In the first step, a random solution is obtained, and in the second step, this solution is fed into the local search phase to obtain the optimal local solution. If the solution obtained from the local search phase is better than the current best solution, this solution will be replaced with the current best solution. These steps continue until the condition for stopping the algorithm is met. For the solution generating phase, the heuristic algorithm is randomised. The following items were random:

1 order of the boxes

2 order of the bins

3 initial situation of box rotation.

For the improvement phase, the following items were taken account of.

For all bins, we try to select a random bin, and if possible, change with another type of the bin. Obviously, the cost of this new bin should be less than the current bin cost. The bin with the least amount of occupancy is chosen and by transferring its boxes to other bins, we try to remove the bin. These two operations cause the local space to be searched and solution shifts towards the optimal local solution.

6 Case study, results and validations

The case study of this paper is the central depot of a spare parts company in which carries out a huge amount of logistical activity daily, such as packing and loading items in a progressive manner. Therefore, reaching an appropriate answer quickly to determine how packing, sorting and loading boxes for bins and various loading vehicles should be performed is very important.

Table 8 Comparison of the results obtained by applying GAMS and heuristic algorithm

Problem name	*GAMS*		*Heuristic*			*GRASP*		
	Score	*Time*	*Score*	*Time*	*% score difference compared to GAMS*	*Score*	*Time*	*% score difference compared to GAMS*
5-1	250	1.8	250	0.14	0	250	18	0
5-2	500	3.2	650	0.01	0.3	500	15.3	0
5-3	500	2	650	0.01	0.3	500	18.9	0
8-1	500	3.6	650	0.01	0.3	500	33.7	0
8-2	500	3.5	650	0.01	0.3	500	38.1	0
8-3	500	15.4	650	0.02	0.3	500	46.6	0
12-1	750	4.5	800	0.04	0.07	750	70.1	0
12-2	800	3,600.9	1,050	0.04	0.31	1,000	66.2	0.25
12-3	750	695.7	1,050	0.05	0.4	1,000	65.2	0.33
15-1	750	3,600	800	0.05	0.07	750	104.3	0
15-2	900	3,600	1,300	0.05	0.44	1,050	91.3	0.17
15-3	750	3,600	1,050	0.04	0.4	900	102.1	0.2
20-1	1,050	3,600	1,300	0.06	0.24	1,200	129.4	0.14
20-2	1,200	3,600	1,200	0.14	0	1,200	114.3	0
20-3	1,050	3,600	1,400	0.07	0.33	1,200	128.8	0.14
50-1	0	3,600	2,450	0.55	-	2,450	359.6	-
50-2	0	3,600	2,700	0.49	-	2,700	298.5	-
50-3	0	3,600	2,000	1.55	-	2,000	355.9	-
70-1	0	3,600	3,500	1.13	-	3,450	566	-
70-2	0	3,600	4,100	1	-	3,700	545.5	-
70-3	0	3,600	3,750	1.34	-	3,700	563.1	-

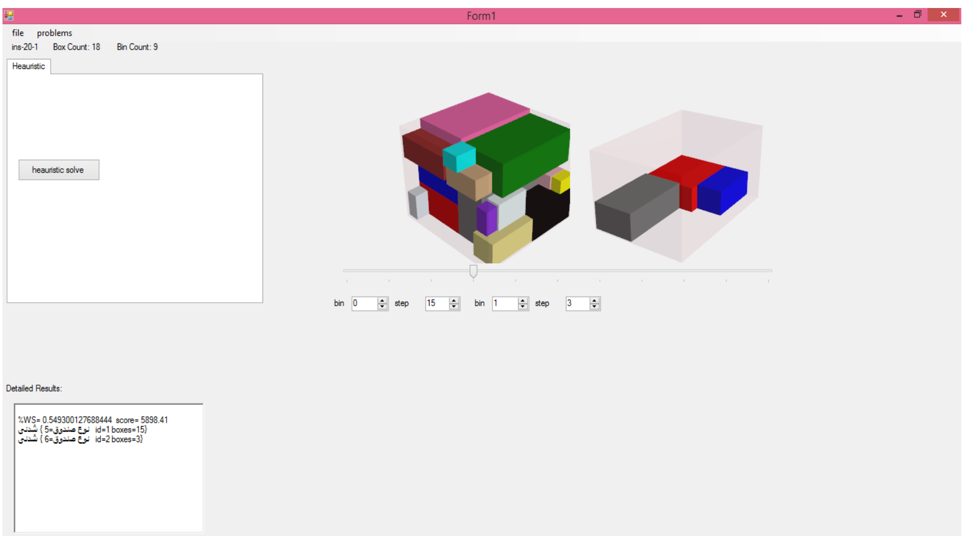
In this section, intending to analyse the efficiency of the heuristic algorithm and the GRASP, the researchers compare the results in terms of the quality and speed of solving with the results of the GAMS software. The mathematical model presented in Section 3.3 was written in the GAMS. The data provided by the company indicated there were eight different types of the bins in the depot, and we assumed the number of available count for each bin was one. For comparison purposes, several problems with the following numbers of boxes were considered: 5, 8, 10, 15, 20, 50 and 70. For each of these categories, three problems were solved. The name of the problem illustrates the number of the boxes and the related problem. For example, problem 70-1 denotes the first problem that has 70 boxes. More detailed specifications of each of the 21 problems are listed in Appendix 1. A summary of the results from the proposed heuristic algorithm and those representing the GAMS software are presented in Table 8. These problems were run on a computer with a CPU core of i5 3.2 GHz and an 8 GB RAM. It should be noted that the heuristic algorithm and the GRASP were coded in C# environment and implemented in the form of software (.exe) to analyse the above problems. The maximum time of the GAMS software was set to one hour.

In some cases, obtaining the smallest possible big-M can cause the model's space to tighten. This can also decrease runtime of exact methods. A lower bound for big-M can be obtained from equations (6) to (9). The big-M amount needs to exceed the sum of the maximum dimensions of all the boxes and that of the maximum dimensions of all the bins [equation (23)].

$$M \geq \max\{L_k, W_k, H_k\} + \max\{l_i, w_i, h_i\} \qquad i \in \{1, \ldots, n\}, k \in \{1, \ldots, m\} \tag{23}$$

For very large amounts of M, problems were run in the GAMS software. Then, the amounts of M were calculated using equation (23) and the problems were solved again through the software. It was observed that the runtime of these two methods did not differ from one another. Therefore, running time in the GAMS software is not sensitive to the different amounts of big-M in this model.

Figure 10 The software environment designed to run the heuristic algorithm and the obtained answer (see online version for colours)



According to Table 8, the results obtained for the objective function values through the heuristic algorithm show a slight difference from those of the GAMS software. The results indicate that for small-sized problems, the GAMS software provides an optimal solution, the answer provided by the GRASP algorithm is also appropriate whereas the heuristic algorithm may not lead to a proper answer. It is also clearly seen that the heuristic algorithm reached the answer at a much higher speed within a shorter time compared to the GAMS software. As the size of the problem increases, the GAMS software solving time increased considerably. This software uses advanced branch and bound algorithms and integer programming heuristic algorithms to obtain the optimal answer. Since the researchers limited the runtime to 3,600 seconds, in the case of five problems, the software obtained an answer, but the answer was probably not optimal.

In the case of six other problems, 50 and 70 problems, the software even failed to find a feasible answer. But in the case of problems solved in less than 3,600 seconds, the GAMS answer is the optimal one. Due to the large amount of daily workload of the company and its need for quick answers, it can be argued that the presented heuristic algorithm has enough sufficiency to meet these needs. With respect to the appropriate answers obtained through this model within much shorter time spans, it may be asserted that it presents a practical and efficient three-dimensional answer to the problem. Also, the GRASP algorithm could relatively improve the answer obtained from the heuristic in a reasonable time.

Figure 10 illustrates the software environment and a sample of the answer obtained through it.

7 Conclusions

This research project studied the 3D-MBSBPP, with its case being the central depot of a large automotive spare parts distribution company. In this regard, a mathematical optimisation model was developed with the aim of minimising the packing costs. Concerning the company's needs, the researchers considered the weight limit of the bins as a new constraint. Furthermore, the researchers took account of the rotation ability of boxes, together with the possibility of considering the restrictions on the number of bin types allowed. In order to solve this model, a new heuristic and a GRASP algorithm were designed and applied with all the conditions and restrictions taken into consideration. To validate and apply this algorithm, the test problems were designed in different sizes. The results of solving these test problems through the presented heuristic and GRASP algorithms were compared with the answers of the exact method. According to the obtained answers, it was found that the proposed heuristic algorithm led to appropriate answers, for large-sized problems. Also, the presented GRASP algorithm improved the heuristic answer in a reasonable time. Whereas these answers were marginally different from those of the exact method, the time to reach the answers in the heuristic algorithm was considerably shorter than that required in the exact solving method. Considering constraints like the maximum number of bin types and the maximum tolerable weight of bins, the ability to rotate boxes by mathematical programming and to solve the resulting model through a fast and simple heuristic, as well as a new greedy algorithm is the features that distinguish this study from previous studies in this area. Regarding the obtained results, the noticeable daily workload of the company and its urgent needs for quick decision-making, the heuristic and software developed are useful in real-world application.

However, the resulted packing is not always physically stable. In the future research, stability constraint can be considered to prevent items from falling down onto the bin floor. Future work can aim to develop mathematical models focused on other constraints, such as the maximum tolerable pressure on each box during packing. The researchers suggest other problems of the company such as vehicle routing and picking of the boxes can be combined with this problem. Other meta-heuristic algorithms like genetic can also be considered to generate better solution in shorter time. Future investigations may intend to shed light on some of the parameters of the problem, including the cost of each bin and the maximum tolerable weight of each bin in uncertain circumstances, in which case the problem can better simulate real-life situations.

References

Alvarez-Valdes, R., Parreño, F. and Tamarit, J. (2012) 'A GRASP/path relinking algorithm for two-and three-dimensional multiple bin-size bin packing problems', *Computers & Operations Research*, Vol. 40, No. 12.

Alvarez-Valdes, R., Parreño, F. and Tamarit, J. (2015) 'Lower bounds for three-dimensional multiple-bin-size bin packing problems', *OR Spectrum*, Vol. 37, No. 1, pp.49–74.

Baazaoui, M., Hanafi, S. and Kamoun, H. (2017) 'Three-dimensional multiple-bin-size bin packing: a case study with a new MILP-based upper bound', *Operational Research in Business and Economics*, pp.225–234, Springer, Cham.

Baker, T., Aldawsari, B., Asim, M., Tawfik, H., Maamar, Z. and Buyya, R. (2018) 'Cloud-SEnergy: a bin-packing based multi-cloud service broker for energy efficient composition and execution of data-intensive applications', *Sustainable Computing: Informatics and Systems*, Vol. 9, pp.242–252.

Bortfeldt, A. and Wäscher, G. (2012) 'Constraints in container loading – a state-of-the-art review', *European Journal of Operational Research*, Vol. 229, No. 1, pp.1–20.

Crainic, T.G., Perboli, G. and Tadei, R. (2009) 'TS2PACK: a two-level tabu search for the three-dimensional bin packing problem', *European Journal of Operational Research*, Vol. 195, No. 3, pp.744–760.

Dyckhoff, H. (1990) 'A typology of cutting and packing problems', *European Journal of Operational Research*, Vol. 44, No. 2, pp.145–159.

Ertek, G. and Kilic, K. (2006) 'Decision support for packing in warehouses', *Computer and Information Sciences – ISCIS 2006*, Springer, pp.115–124.

Feo, T.A. and Resende, M.G. (1995) 'Greedy randomized adaptive search procedures', *Journal of Global Optimization*, Vol. 6, No. 2, pp.109–133.

Hu, H., Duan, L., Zhang, X., Xu, Y. and Wei, J. (2018) *A Multi-task Selected Learning Approach for Solving New Type 3D Bin Packing Problem*, arXiv preprint arXiv: 1804.06896, https://arxiv.org/abs/1804.06896.

Kurian, J., Unnikrishnan, S. and Sawant, B. (2018) 'A study on green supply chain management practices in the Indian petroleum industries', *International Journal of Services and Operations Management*, Vol. 31, No. 2, pp.260–276.

Li, T-H.S., Liu, C-Y., Kuo, P-H., Fang, N-C., Li, C-H., Cheng, C-W., Hsieh, C-Y., Wu, L-F., Liang, J-J. and Chen, C-Y. (2017) 'A three-dimensional adaptive PSO-based packing algorithm for an IoT-based automated e-fulfillment packaging system', *IEEE Access*, Vol. 5, pp.9188–9205.

Li, X. and Zhang, K. (2015) 'A hybrid differential evolution algorithm for multiple container loading problem with heterogeneous containers', *Computers & Industrial Engineering*, Vol. 90, pp.305–313.

Loni, P., Khamseh, A.A. and Pasandideh, S.H.R. (2018) 'A new multi-objective/product green supply chain considering quality level reprocessing cost', *International Journal of Services and Operations Management*, Vol. 30, No. 1, pp.1–22.

Manimaran, P. and Selladurai, V. (2014) 'Efficient heuristics for cost optimisation in multi stage supply chain network with a fixed transportation charge', *International Journal of Services and Operations Management*, Vol. 18, No. 2, pp.115–141.

Mohiuddin, I., Almogren, A., Al Qurishi, M., Hassan, M.M., Al Rassan, I. and Fortino, G. (2019) 'Secure distributed adaptive bin packing algorithm for cloud storage', *Future Generation Computer Systems*, Vol. 90, pp.307–316.

Paquay, C., Limbourg, S. and Schyns, M. (2017) 'A tailored two-phase constructive heuristic for the three-dimensional multiple bin size bin packing problem with transportation constraints', *European Journal of Operational Research*, Vol. 267, No. 1, pp.52–64.

Paquay, C., Limbourg, S., Schyns, M. and Oliveira, J.F. (2018) 'MIP-based constructive heuristics for the three-dimensional bin packing problem with transportation constraints', *International Journal of Production Research*, Vol. 56, No. 4, pp.1581–1592.

Paquay, C., Schyns, M. and Limbourg, S. (2016) 'A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application', *International Transactions in Operational Research*, Vol. 23, Nos. 1–2, pp.187–213.

Pisinger, D. and Sigurd, M. (2005) 'The two-dimensional bin packing problem with variable bin sizes and costs', *Discrete Optimization*, Vol. 2, No. 2, pp.154–167.

Ramasubramaniam, M., Mathirajan, M. and Ramachandran, V. (2010) 'Minimising makespan on a single heat-treatment furnace in the steel casting industry', *International Journal of Services and Operations Management*, Vol. 7, No. 1, pp.112–142.

Reisi-Nafchi, M. and Moslehi, G. (2015) 'Integrating two-agent scheduling and order acceptance problems to maximise total revenue by bounding each agent penalty function', *International Journal of Services and Operations Management*, Vol. 20, No. 3, pp.358–384.

Song, B., Li, Y., Chen, Y., Yao, F. and Chen, Y. (2018) 'A repair-based approach for stochastic quadratic multiple knapsack problem', *Knowledge-Based Systems*, Vol. 145, pp.145–155.

Wäscher, G., Haußner, H. and Schumann, H. (2007) 'An improved typology of cutting and packing problems', *European Journal of Operational Research*, Vol. 183, No. 3, pp.1109–1130.

Wei, L., Oon, W-C., Zhu, W. and Lim, A. (2013) 'A goal-driven approach to the 2D bin packing and variable-sized bin packing problems', *European Journal of Operational Research*, Vol. 224, No. 1, pp.110–121.

Wu, H., Leung, S.C., Si, Y-w., Zhang, D. and Lin, A. (2017) 'Three-stage heuristic algorithm for three-dimensional irregular packing problem', *Applied Mathematical Modelling,* Vol. 41, pp.431–444.

Zhao, Y., Zhang, Z., Wu, N., Liu, H. and Zuo, D. (2018) *Heterogeneous and Multidimensional Clairvoyant Dynamic Bin Packing for Virtual Machine Placement*, arXiv preprint arXiv: 1802.03152.