Multi-objective 3D bin-packing problem

Jasim Hasan
College of Information Technology
University of Bahrain
Manama, Bahrain
jasim.bh@gmail.com

Jihene Kaabi
Dept. of Information Systems
College of Information Technology
University of Bahrain
Manama, Bahrain
jkaapi@uob.edu.bh

Youssef Harrath
Dept. of Computer Science
College of Information Technology
University of Bahrain
Manama, Bahrain
yharrath@uob.edu.bh

***Abstract*—This research paper deals with a special case of a multiobjective 3D-Bin Packing Problem. n 3D boxes of different volumetric dimensions are to be filled in a minimum number of identical bins. The boxes have different weights and can be only horizontally rotated when placed in the bins. Two objectives are simultaneously considered: Use the minimum number of bins to pack all the boxes and have balanced bins in term of total weight. The investigated problem is NP-hard. A 3D algorithm is proposed to solve this problem in two main phases. During the first phase, all the possible combinations of layers of same or different types of boxes that can fit in a bin are generated. These combinations represent candidate solutions in term of bin's volume. Finally, in the second phase the boxes are packed in the bins according to the best use of the bin's volume from the solutions candidates. This 3D algorithm was validated using real world data from the courier company Fedex and compared with the results of a lower bound and another recently published algorithm. The results showed that the proposed algorithm is much better in both criteria.**

***Index Terms*—3D bin-packing, multi-objectives optimization, layer based algorithm, lower bound.**

I. Introduction

The Bin Packing Problem (BPP) is an NP-hard combinatorial optimization problem [8]. The BPP consists on filling available bins with boxes of different dimensions. The aim is to load the boxes using a minimum number of bins respecting diverse constraints related to the boxes' dimensions, the bins' maximum weight, volume, and 3D dimensions. This problem of finding the suitable placement of boxes into the bins is known as 3D-BPP. One or several objectives can be considered. These range from minimizing the number of bins used or the unused space to balancing the bins' load. Several industrial applications of the 3D-BPP can be found e.g. vehicle loading and cargo handling in airplanes.

Researchers have examined many variations of Bin Packing Problems, and recently more focus is on multi-objective problems. Although this type of problems is the most complicated, it is faced in many real life situations such as in courier industry.

The 3D-BPP was widely studied in the last few decades under various constraints and objectives. Many solution approaches have been proposed and can be classified mainly into three categories: exact algorithms, evolutionary and nature-based methods and heuristic approaches.

The exact algorithms developed to solve the 3D-BPP include the Branch-and-Bound algorithm [1] and the Branch-and Price algorithm [2].

Evolutionary and nature-based algorithms such as Genetic Algorithms and particle Swarm Optimization are very commonly used to solve various variants of the BPPs. In [3], the researchers proposed an Adaptive Genetic Algorithm (AGA) to solve the 3D multi-objective bin packing problem. The authors considered constraints on the number of bins, size, shape, weight, and bin dimension. They stated that the computational results are satisfactory and their method is effecient in term of the execution time. The authors in [5] considered more practical constraints in the cargo industry; such as, box orientation, stack priority, container stability, weight, overlapping, and placement. They achieved the research goal, by optimizing the empty volume inside the bin using a hybrid Genetic approach.

Researchers have contributed with many other algorithms other than Evolutionary algorithms. In [6], a layer-based solution for the 3D multi-objective bin-packing problem was proposed. The solution approach consists on building horizontal layers of identical items first, and then generating packing pattern by greedily loading the layers using multiple selection criteria.

For further details, the reader can refer to [4]. In [4], the authors surveyed a wide range of exact and approximation techniques developed to solve many classes of BPP problems under different constraints and objectives.

This paper deals with a multiobjective 3D-Bin Packing Problem. The main contribution includes the development of a Layer Based Approach consisting of two main phases. The first phase generates all the possible combinations of horizontal layers with only one box type per layer that can fit in a bin volume-wise. However, the second stage explores all the possible combinations of horizontal layers obtained in the first phase to find the minimum number of bins needed to pack the boxes waiting to be shipped as well as their placements within the bins.

This paper is organised as follows. In the next section, the problem formulation and notations used will be detailed. Section III, will illustrate the proposed solution approach. The experimental results will be provided in section IV. Finally, a summary of the study findings will be given in section V.

978-1-5386-7684-4/19/$31.00

II. Problem Description and Formulation

The 3D-Bin Packing Problem consists on loading n cubic 3D boxes of different dimensions and weights into a minimum number of identical bins. The parameters used to formulate the problem and describe the solution approach are summarized in Table I

TABLE I
USED NOTATIONS

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

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 boxed are to be placed. Let

$$x_{ij} = \begin{cases} 1 & \text{if box } i \text{ is placed in bin } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The objective is to pack all the boxes in a minimum number of bins respecting the following constraints.

- The total weight of all boxes within a bin j shouldn't exceed the maximum allowed weight of the bin. This constraint can be formulated as

$$\sum_{k=1}^{n} w_k x_{kj} \leq W \quad \forall j, \quad j = 1, \ldots, m$$

- The total boxes volume within the bin j shouldn't exceed the bin's volume. This constraint can be formulated as

$$\sum_{k=1}^{n} v_k x_{kj} \leq V \quad \forall j, \quad j = 1, \ldots, m$$

  This constraint is necessary but not sufficient. Indeed, if the set of selected boxes have a total volume less than the bin's volume, the problem is to prove whether there is a way to load these boxes respecting the bin's dimensions.

III. Research Methodology

This study aims to propose a solution to the 3D-BPP under two objectives: minimizing the number of bins used and balancing the bins loads. These two objectives affect the shipment cost. Indeed, the smaller the number of bins used to pack the boxes, the smaller is the cost. Additionally, having the bins weight balanced or almost balanced, increases the safety of the transportation mean used for shipment.

In order to meet the objectives of the research and satisfy all the constraints, a novel Layer Based 3D Bin Packing Problem algorithm was designed, denoted by LBA-3DBPP. The main idea of LBA-3DBPP consists on filling the bins on layer basis. This algorithm is composed of two parts.

The first part will run only one time when setting new bin dimensions and maximum weight, and boxes types along with their dimensions. This part consists on the following two steps:

- **Step1**: Identify the best orientation for each 3D box type considering only it's base face so as to place the maximum of boxes per horizontal layer. Each layer should consist only from boxes of the same type as shown in Fig 1 and Fig 2. The output of this step will store box type orientation and the maximum number of each box type per layer within the horizontal dimensions; length and depth.

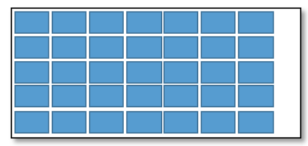

Fig. 1. orientation option 0

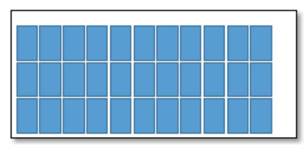

Fig. 2. orientation option 1

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**
  $hlc_i = max(\lfloor \frac{L}{l_i} \rfloor \times \lfloor \frac{D}{d_i} \rfloor, \lfloor \frac{D}{l_i} \rfloor \times \lfloor \frac{L}{d_i} \rfloor)$
  return hlc_i.
end for

- **Step 2**: Identify all possible candidate solutions formed by combining layers of different box types that can vertically fit in a bin without exceeding it's height. In this step, the total weight of boxes is not considered. Therefore, some candidate solutions would not be feasible and will be discarded further. The details of this step are summarized in Algorithm 2.

The second part of the proposed LBA-3DBPP algorithm focuses on placing the received boxes into the 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

Algorithm 2 3D-candidate solutions

Input D, L, H, d_i, l_i, h_i, $i = 1 \ldots t$.
Output set of combined layers that can volumetrically fit within a bin.
$S = \emptyset$
for $r_1 = 0$ to $\lfloor \frac{H}{h_1} \rfloor$ **do**
  for $r_2 = 0$ to $\lfloor \frac{H}{h_2} \rfloor$ **do**
    $\vdots$
    for $r_t = 0$ to $\lfloor \frac{H}{h_t} \rfloor$ **do**
      if $(\sum_{i=1}^{t} r_i h_i < H)$ **then**
        $S = S \cup (\cup_{i=1}^{t} r_i$ layers of boxes type $i)$
      end if
    end for
  end for
end for
return S.

Algorithm 3 LBA-3DBPP

1: **Input:** D, L, H, W, d_i, l_i, h_i, n_i, $i = 1 \ldots t$ and w_k, $k = 1 \ldots n$.
2: **Output:** m filled bins (m to be minimized).
3: $S =$ result of **3D-candidate solutions** 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** $U \neq \emptyset$ **do**
12:   **for** $p = 1$ to $|S|$ **do**
13:     $W_{S_p} = 0$
14:     **for** $i = 1$ to t **do**
15:       $n_{ip} =$ number of boxes type i in solution S_p
16:     **end for**
17:     **if** $W_{S_p} > W$ **then**
18:       accept$S_p =$ false
19:       check the next solution (line 12)
20:     **end if**
21:     **for** $i = 1$ 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.

a bin respecting the bin's maximum weight. The pseudocode 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. This condition guarantees that the provided solution satisfies the constraint of the bin's maximum weight.

IV. Experimental Results

The LBA-3DBPP algorithm was programmed using Visual Studio 2017 (C# language), and MS SQL Server 2014 database. The bins and boxes parameters as well as the shipments details are stored in SQL tables. The full algorithm parts were implemented using C# language adopting MVC (Model, View, Controller) concept with the aim to maintain the modulatory of the program. The program 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.

For the purpose of testing the proposed algorithm, real data from the courier company Fedex was used. Two types of boxes were considered, namely BoxA and BoxB, and one type of bins having a standard size used for air courier. Table II shows the details of the dimensions and weights of the bins as well as the boxes.

TABLE II
Fedex Data used to test LBA-3DBPP

Type	Depth (cm)	Height (cm)	Length (cm)	Max. Weight (Kg)
Bin	243.8	178	317.5	6804
BoxA	40.16	32.86	25.88	10
BoxB	54.8	42.1	33.5	25

The results of the implementation of 2D-layer-candidates algorithm applied for the case of Fedex boxes and bins are shown in Table III

TABLE III
2D-layer-candidates results

Box Type	Orientation	number of boxes per horizontal layer
BoxA	0	72
BoxB	0	36

Once the number of horizontal layers from each box type was calculated, the 3D-candidate solutions algorithm was tested. The computational results include the number of layers from each box that can be fitted in a bin in term of height, the number of boxes of each type in one horizontal layer and the percentage of utilization of the bin (denoted by % Utl. The candidate solutions will be then stored in a database. Table IV illustrates the results obtained for the two considered types of boxes.

After generating all the possible volumetrically feasible solutions, the proposed novel algorithm LBA-3DBPP was tested on problem instances with different numbers of boxes.

TABLE IV
3D-CANDIDATE SOLUTIONS RESULTS

Solution ID	#layers (BoxA)	#layers (BoxB)	#BoxA per layer	#BoxB per layer	% Utl
1	4	1	288	36	97.4
2	0	4	0	144	94.6
3	5	0	360	0	92.3
4	1	3	72	108	89.4
5	2	2	144	72	84.2
6	3	1	216	36	79
7	4	0	288	0	73.8
8	0	3	0	108	70.9
9	1	2	72	72	65.7
10	2	1	144	36	60.5
11	3	0	216	0	55.3
12	0	2	0	72	47.3
13	1	1	72	36	42.1
14	2	0	144	0	36.9
15	0	1	0	36	23.6
16	1	0	72	0	18.4

The computational results were then compared with those of a study published earlier by the co-authors of this research [7]. Table V displays the minimum number of bins needed to pack all the boxes obtained by LBA-3DBPP, the algorithm developed in [7] (denoted by A) and a simple lower bound LB, respectively. Equation 2 illustrate the mathematical formulation of LB.

$$LB = \frac{\sum_{k=1}^{n} w_k}{W} \tag{2}$$

From Table V, it is noticed that LBA-3DBPP provides optimal number of bins for a number of boxes up to 1000 and outperforms the other approaches for a bigger number of boxes.

TABLE V
COMPUTATIONAL EXPERIMENTS WITH DIFFERENT NUMBER OF BOXES

# boxes	LBA-3DBPP	A	LB
100	1	1	1
500	3	3	3
1,000	5	6	5
2,500	13	15	11
5,000	26	30	22
7,500	39	46	33
10,000	52	60	44
25,000	131	153	109
50,000	261	305	219
75,000	392	457	328
100,000	522	609	437

Taking an example of the execution of LBA-3DBPP with 2,500 boxes, Fig 3 shows the volume and weight utilization of each bin, where the total number of bins used is 13 bins, while Fig 4 shows the total weight for each of the 13 bins for the same problem. Fig 4 shows minimal differences between the bins, which meets the objective of loading the balance of all the bins (minimizing difference between the weight of each bin and the average weight).

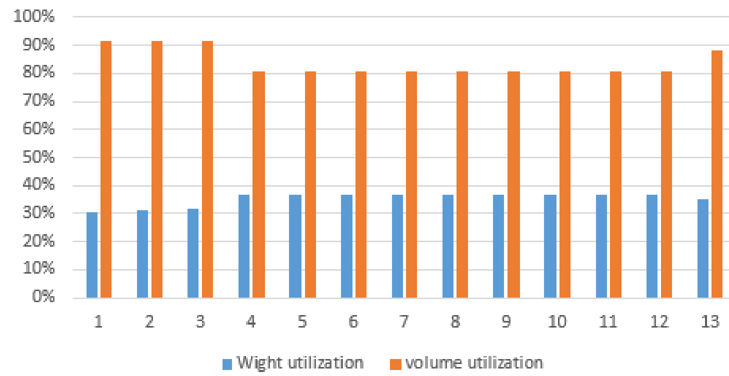

Fig. 3. LBA-3DBPP volume and weight utilization results for 2500 boxes

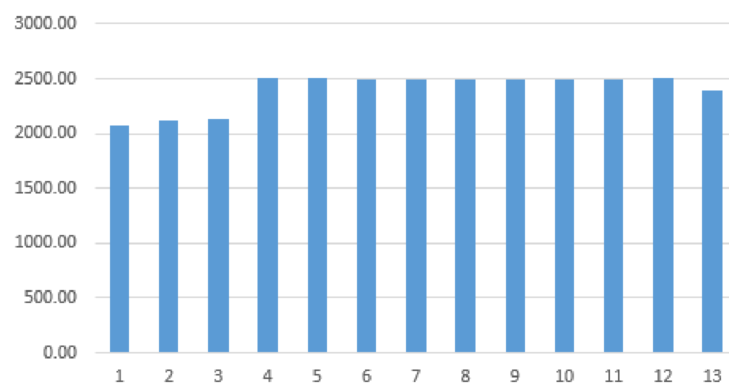

Fig. 4. LBA-3DBPP 2500 boxes solution bin's weight

The main noticeable advantage of the proposed algorithm was the running time. Table VI displays the computational times needed to provide solutions for the different problem instances. It clearly shows that LBA-3DBPP is able to generate a solution for instances with up to 5,000 boxes in less than 1 second. In addition for very large problems such as with 100,000 boxes, the computational time was 48 seconds.

TABLE VI
LBA-3DBPP COMPUTATIONAL TIMES WITH DIFFERENT NUMBER OF BOXES

# boxes	CPU (sec)
100	0.01
500	0.057
1,000	0.107
2,500	0.350
5,000	0.837
7,500	1.843
10,000	2.177
25,000	5.787
50,000	17.23
75,000	28.31
100,000	48.16

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 the algorithm ready to be a commercial product to solve real world bin-packing problems.

V. Conclusion

In this research, a multiobjective 3D-Bin Packing Problem was investigated. Two conflicting objective were considered. The first is to minimize the number of bins used to pack all the boxes waiting to be shipped. Whereas the second criterion focuses on balancing the load of the bins used.

A novel algorithm was developed consisting of two main phases. The first phase generates all the possible candidate solutions respecting the bin's depth, length and height. The output of this later will be input for the second phase that explores the solutions set to select the one that uses the minimum number of bins needed to pack the boxes waiting for shipment. The second phase will also determine the best placement of the boxes within the bins. The algorithm was made generic in a way that it can be applicable for any number of box types. However, in the implementation phase the real case of the courier company, Fedex was considered, where only two types of boxes are used. The computational results were promising especially when a big numbers of boxes was considered.

Further researches can be made to improve the developed algorithm and make it more generic and applicable for wider range of real world problems. Indeed, the rotation of boxes in all directions and layers with mixed box types can be further considered. Also, due the design of the algorithm and its high speed in finding solutions, it could be implemented as an Application on smart phones.

References

[1] S. Kuhn and D. Pisinger and D. Vigo, "The three-dimensional bin packing problem," Operations Research, vol. 48, pp. 256–267, 2000.

[2] S. Elhedhli and L. Lingzi and M. Gzara and J. Naoum-Sawaya, "A Branch-and-Price Algorithm for the Bin Packing Problem with Conflicts," Journal on Computing, vol. 23(3), pp. 404–415, 2011.

[3] R. Sridhar and M. Chandrasekaran, "Multi Objective Optimization of Heterogeneous Bin Packing using Adaptive Genetic Approach," Indian Journal of Science and Technology, vol. 9(48), 2016.

[4] H.I Christensen and A. Khan and S. Pokutta and P. Tetali, "Approximation and online algorithms for multidimensional bin packing: A survey," Computer Science Review, vol 24, pp. 63–79, 2017.

[5] R. Sridhar and M. Chandrasekaran and C. Sriramya and T. Tom, "Optimization of heterogeneous Bin packing using adaptive genetic algorithm," in International Conference on Emerging Trends in Engineering Research: Vol. 183(1), 2017, pp. 12-26.

[6] R. D. Saraiva and N. Nepomuceno and P. R. Pinheiro, "A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company," IFAC-PapersOnLine, vol. 48(3), pp. 490–495, 2015.

[7] J. Kaabi and Y. Harrath and H.E. Bououdina and A.T. Qasim, "Toward Smart Logistics: A New Algorithm for a Multi-Objective 3D Bin Packing Problem," in Smart Cities Symposium 2018 (SCS'18): 22-23 April, 2018.

[8] M. R. Garey and D. S. Johnson, (1979). Computers and intractability: a guide to NP-completeness.