American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-9, Issue-8, pp-127-133
www.ajer.org
Research Paper Open Access

Cylindrical Shape Bin Packing Problem with Cartesian Coordinate System

Niluka P. Rodrigo[1*], W. B. Daundasekera[2], A. A.I. Perera[3]
[1,2,3] *Department of Mathematics, Faculty of Science, University of Peradeniya, Peradeniya, Sri Lanka*

ABSTRACT:*A bin-packing problem (BPP) can be interpreted as a finite collection of items with varying specifications to be packed into one or more containers/ box utilizing the maximum volume of the containers/ box while satisfying the supply-demand. Each container can hold any subset of the collection of objects without exceeding its capacity. Bin packing is also called as container loading, box packing, cargo loading, knapsack, etc. A burning issue faced by the industries is how to find the optimum layout (packing arrangement) of boxes or packing items such that to improve the utilization ratio of bins or minimize the bin slack without overlapping the packages. In This study, an algorithm has developed to determine the dimensions of bin/ container should be used to pack the cylindrical shape items by satisfying customer demand.*

KEYWORDS: *Bin-packing problem, Modified Branch and Bound Algorithm, Square packing method, Hexagonal packing method, Matlab Software environment.*

Date of Submission: 07-08-2020 Date of acceptance: 21-08-2020

I. INTRODUCTION

Due to manufacturing revolution, at present most of the manufacturing are focusing in the frame of globalization. In turn, industries have substantially amplified the capacity and magnitude of their global manufacture and allocation network around the world. Thus the globalized market results in rapid development of international trade and creates intensive competitions among the industries. In this research, our objective is to investigate a mathematical formulation of solving the Three-Dimensional Bin Packing Problem (3D BPP) which has been studied by many researchers in the recent past. Packing the produced goods in the optimum manner within a limited space has multiple paybacks to the production plant. A Bin Packing Problem (BPP) basically describes in three ways; 1D, 2D and 3D BPP. The BPP can be interpreted as a finite collection of items with varying specifications to be packed into one or more containers/ box utilizing the maximum volume of the containers/ box while satisfying the supply-demand. Each container can hold any subset of the collection of objects without exceeding its capacity. Bin packing is also called as container loading, box packing, cargo loading, knapsack, etc.

A burning issue faced by the industries is how to find the optimum layout (packing arrangement) of boxes or packing items such that to improve the utilization ratio of bins or minimize the bin slack without overlapping the packages. Jatinder N. D. Gupta and Johnny C. (1999) have described a new Heuristic Algorithm to solve the one-dimensional Bin Packing problem. Effectiveness of the proposed algorithm has been compared with the First Fit Decreasing (FFD) and the Best Fit Decreasing (BFD) algorithms using five different data sets [1]. Solve the one dimensional BPP with island parallel grouping genetic algorithms was the main motivation of Dokeroglu T. et al (2014). Combining state-of-the-art computation tools; parallel processing, GGAs and bin oriented heuristics to efficiently solve the intractable one-dimensional BPP. Different size case studies discussed in the paper using Minimum Bin Slack (MBS) and Best Fit Decrease (BFD) / First Fit Decrease (FFD) [2]. Mukhacheva E. A et al (2000) developed two algorithms for one-dimensional cutting packing problem, namely, a modified Branch-and-Bound method (MBB) and Heuristic Sequential Value Correction (SVC) method. Efficiency of the algorithms is discussed from the computational experiment and it seems that the efficiency of the SVC method appears to be superior to that of the MBB [3].

Fleszar K. and Khalil S. presented a new Heuristic Algorithm for minimum bin slack. In this research, each packing is determined in a search procedure that tests all possible subsets of items on the list which _t the bin capacity (C). The slack in packing A is expressed by S (A) =C and updating every time an item is added or removed from A [4]. Christian Blum and Verena Schmid (2013) have dealt with two-dimensional Bin Packing problem under free guillotine cutting, a problem in which a set of oriented rectangular items are given which must be packed into minimum number of bins of equal size. An evolutionary algorithm has been discussed and the results of the proposed algorithm are compared with some of the best approaches from the literature. There is a intercommunication in between Cutting Stock Problem (CSP) and BPP.

Many researchers have worked on the cutting stock problem as well and designed different algorithmic approaches to solve the problem [5]. Among them, Saad (2001) modified Branch and Bound Algorithm to find feasible cutting patterns for one dimensional cutting stock problem and formulated a mathematical model to minimize the total cut loss. Modified Branch and Bound Algorithm is illustrated using a case study [6]. Further Rodrigo et al (2012) developed an algorithm, based on Modified Branch and Bound algorithm to determine the feasible cutting patterns for Two-Dimensional cutting stock problem with rectangular shape cutting items. The method was illustrated with the use of a case study, where the data were obtained from a floor tile company known as Mega Marble Company located in London. A computer programme was coded using Matlab inbuilt functions [7]. As an extension of the above study, Rodrigo et al (2012) redesigned the developed algorithm to determine the locations of each cutting item using Cartesian Coordinate Geometry [8]. Rodrigo et al (2015) modified the Branch and Bound Algorithm to decide the feasible cutting patterns for One-Dimensional cutting stock problem and to determine the locations of each cutting item using Cartesian Coordinate Geometry. The algorithm has been coded using Matlab environment illustrated the algorithm using a case study [9]. Thereafter, Rodrigo et al (2017) has made an approach to nest one-dimensional cutting items within rectangular shaped main sheet with known varying dimensions. Cartesian coordinate points of each item in each cutting sheet have been determined using a computer programme coded in Matlab software environment [10]. Besides, identification of cutting-packing location within the different sizes of container (or main sheet) is significantly crucial to outline items within the container depending on the selected cutting-packing pattern. To address this issue, further development of MBBA will be made to ascertain the locations of each cutting-packing item within the main sheet (or container) by using Cartesian coordinate system. Developed algorithm will be coded and programmed in the Matlab Software environment to generate feasible packing patterns for cylindrical shape BPP.

In this paper, packing items with cylindrical shapes are selected to nest into boxes with rectangular shape base. There are different arrangements to pack the required items into the boxes and each arrangement is defined as a packing pattern. An algorithm based on *MBBA* and a computer program using Matlab software package to solve the algorithm are used to generate feasible packing patterns. Modified mathematical model is used to acquired optimum packing schedule and compare the results with real world data.

II. MATERIALS AND METHOD

2.1. Three-Dimensional (3D) BPP

There exist two different methods to pack the cylindrical shape packing items within the cuboid/ cube shape box/ bin.

Method 1: Fit the cylindrical shape item into a cuboid/ cube and then nest the fitted cuboid/cube into the bin.

Total number of cuboids/ cube that can be nested within the bin =

$$\text{Int}\left(L_k/l_i\right) * \text{Int}\left(W_k/w_i\right) * \text{Int}\left(H_k/h_i\right)$$

Method 2: An algorithm (*MBBA*) is developed to nest cylindrical shape items with fixed diameter into a bin with known dimensions.

According to the algorithm, bottom layer of the bin can be fitted as follows:

Maximum number of cylinders (a_1) are nested along the length (L_k) of the bottom layer of the bin. Number of rows can be calculated using the formula given below:

$R = \text{smallest int}\left(W_k/g\right)$, where W_k is the width of the bin, g is the distance between two adjacent rows and $g = \sqrt{3}\, d/2$.

Gradual reduction of cylinders, one followed by other of a_1, cylinders are fitted along the width of the bin. Remaining horizontal lengths in both sides (left and right) of the bin are calculated in each row. If horizontal length ≥ diameter of the cylinder, cylinder is nested to that particular residual space and so on.

2.2 Modified Branch and Bound Algorithm (MBBA) in step-wise form:

Step 1: Arrange the length of the bins (L_k ; k = 1, 2,…, m) in descending order. According to the order of the length, input the width and height of the each bin.

Arrange the length of the packing items (l_i ; i = 1, 2,…, n) in descending order. According to the order of the length, input the width and height of the each packing item.

Step 2: $R = \left[\left[\dfrac{W_k}{\sqrt{3}\, d(i)/2} \right]\right]$, for $k = 1, 2, ..., m$, where W_k is the width (shortest length) of the k^{th} bin and $d(i)$ is the diameter of the cylinder. Here, R is the number of rows of the cylinder along the width of the bin and $[[y]]$ is the greatest integer less than or equal to y.

Step 3: If $R \geq 1$, do Step 4 to Step 6.

Step 4: Set $a_{1k} = \left[\left[L_k / d(i) \right]\right]$ for $k = 1, 2, ..., m$,where L_k is the length (longest length) of the k^{th} bin. Here, a_{1k} is the number of cylinders can be cut along the first row of the k^{th} bin.

Step 5: Set $p_k = a_{1k}$, where p_k is the number of cylinders in the k^{th} bin.

Locations of the i^{th} cylinder of the first row in the k^{th} bin:

$$\left(x_{1ik}, y_{1ik}\right) = \left(\left(\frac{2i-1}{2}\right) d(i), \frac{d(i)}{2} \right); \text{ for} \forall\ i = 1, 2, ..., a_{1k} \ ; \ k = 1, 2, ..., m$$

Step 6: Cut loss along the length of the bin

$c_{1kr} = L_k - \left(a_{1k}\ d(i)\right)$;

$c_{1kl} = 0$, where c_{kr} and c_{kl} are cut loss length in the right hand side and left- hand side of the bottom layer of the k^{th} bin respectively.

Step 7: For z =2, 3, …, n

For k =2, 3, …, m

Set $a_{zk} = a_{(zk-1)} - 1$;

$p_k = p_k + a_{zk}$;

$c_{zkr} = c_{(z-1)kr} + d(i)/2$; and $c_{zkl} = c_{(z-1)kl} + d(i)/2$.

If $c_{zkr} \geq d(i)$

Set $A_{zkr} = \left[\left[c_{zkr} / d(i) \right]\right]$; $p_k = p_k + A_{zkr}$.

If $c_{zkl} \geq d(i)$

Set $p_k = p_k + A_{zkl}$.

else set $A_{zkr} = 0$; $A_{zkl} = 0$;

$p_k = p_k$.

Locations of the i^{th} cylinder of the z^{th} raw in the k^{th} bin:

$$(x_{zik}, y_{zik}) = \left((i\,d(i)), \left(h(z-1) + \frac{d(i)}{2} \right) \right); \text{ for even } z$$

$$(x_{zik}, y_{zik}) = \left(\left(\frac{2i-1}{2} \right) d(i), \left(h(z-1) + \frac{d(i)}{2} \right) \right); \text{ for odd } z$$

Step 8: Total cut loss area =

$$(L_k \times W_k \times H_k) - \left(\tfrac{1}{4}\, p_k\, \pi\, d^2(i)\, h_i \right) \quad \text{for } k = 2, 3, \ldots, m.$$

$$i = 2, 3, \ldots, n.$$

Step 9: STOP.

Developed algorithm is coded in Matlab programming language to generate feasible number of cylinders that can be nested into a box and to interpret Cartesian coordinate points of each radius which corresponds to each cylinder.

2.3 Case Study

Following case study will illustrate how to generate feasible packing patterns by minimizing unused area:
A Potato-Chips manufacturing company distributed their products as paper tube cylinders and packets. Potato-chips paper tube cylinders have transported by packing the cylinders into corrugated boxes. Dimensions of paper tube cylinders and corrugated boxes were given below:

Item no	Diameter (mm)	Height (mm)	Demand
1	75	260	300
2	65	215	190

Table 2.1 Dimensions and demand of packing items

Box no	Length (mm)	Width (mm)	Height (mm)
1	525	440	260
2	500	375	215

Table 2.2 Dimensions of boxes

III. RESULTS AND DISCUSSION

Modified Branch and Bound Algorithm is applied to the above examples to generate feasible packing patterns. According to the algorithm, number of paper tube-cylinders can be packed into a box as follows:

	No of cylinders can be packed into a box			
	Item 1		Item 2	
Box No	Method 1	Method 2	Method 1	Method 2
1	35	39	48	53
2	0	0	35	39

Table 3.1 Packing schedule of each item

```
Method 2 is the best method to pack the item no(1) into the box no(1)
Number of pieces can be packed from item no (1) to box no (1) = 39
Number of boxex should be used to satisfy the demand = 8

Method 2 is the best method to pack the item no(2) into the box no(2)
Number of pieces can be packed from item no (2) to box no (2) = 39
Number of boxex should be used to satisfy the demand = 5
```

Matlab command window (result page) related to the Item no 1 and Box no 1:

```
>> cylinderbin
Number of bins with various sizes = 2
Length of the box = [525,500]
Width of the box = [440,375]
Height of the box = [260,215]
Number of pieces = 2
Diameter of the cylinder = [75,65]
Height of the cylinder = [260,215]
Demand of the cylinder in each size = [300,200]

 Number of layers can be arranged within the box no (1) = 1
 Item type of the size 1 pack to the Box 1

 Method 2

 Number of circles in (1) row  = 7

 center of 1 circle in (1) row = (3.750000e+01,3.750000e+01)
 center of 2 circle in (1) row = (1.125000e+02,3.750000e+01)
 center of 3 circle in (1) row = (1.875000e+02,3.750000e+01)
 center of 4 circle in (1) row = (2.625000e+02,3.750000e+01)
 center of 5 circle in (1) row = (3.375000e+02,3.750000e+01)
 center of 6 circle in (1) row = (4.125000e+02,3.750000e+01)
 center of 7 circle in (1) row = (4.875000e+02,3.750000e+01)

 Number of circles in (2) row  = 6
center of 1 circle in (2) row = (75,1.024519e+02)
center of 2 circle in (2) row = (150,1.024519e+02)
center of 3 circle in (2) row = (225,1.024519e+02)
center of 4 circle in (2) row = (300,1.024519e+02)
center of 5 circle in (2) row = (375,1.024519e+02)
center of 6 circle in (2) row = (450,1.024519e+02)

Number of circles in (3) row  = 7

center of 1 circle in (3) row = (3.750000e+01,1.674038e+02)
center of 2 circle in (3) row = (1.125000e+02,1.674038e+02)
center of 3 circle in (3) row = (1.875000e+02,1.674038e+02)
center of 4 circle in (3) row = (2.625000e+02,1.674038e+02)
center of 5 circle in (3) row = (3.375000e+02,1.674038e+02)
center of 6 circle in (3) row = (4.125000e+02,1.674038e+02)
center of 7 circle in (3) row = (4.875000e+02,1.674038e+02)

Number of circles in (4) row  = 6
```

```
center of 1 circle in (4) row = (75,2.323557e+02)
center of 2 circle in (4) row = (150,2.323557e+02)
center of 3 circle in (4) row = (225,2.323557e+02)
center of 4 circle in (4) row = (300,2.323557e+02)
center of 5 circle in (4) row = (375,2.323557e+02)
center of 6 circle in (4) row = (450,2.323557e+02)

Number of circles in (5) row  = 7

center of 1 circle in (5) row = (3.750000e+01,2.973076e+02)
center of 2 circle in (5) row = (1.125000e+02,2.973076e+02)
center of 3 circle in (5) row = (1.875000e+02,2.973076e+02)
center of 4 circle in (5) row = (2.625000e+02,2.973076e+02)
center of 5 circle in (5) row = (3.375000e+02,2.973076e+02)
center of 6 circle in (5) row = (4.125000e+02,2.973076e+02)
center of 7 circle in (5) row = (4.875000e+02,2.973076e+02)

Number of circles in (6) row  = 6

center of 1 circle in (6) row = (75,3.622595e+02)
center of 2 circle in (6) row = (150,3.622595e+02)
center of 3 circle in (6) row = (225,3.622595e+02)
center of 4 circle in (6) row = (300,3.622595e+02)
center of 5 circle in (6) row = (375,3.622595e+02)
center of 6 circle in (6) row = (450,3.622595e+02)
Number of cylinders can be packed into bin (1) = 39

Unused volume = 1.524482e+07

Utilization = 7.461735e+01
```

With respect to the matlab command window, packing item 1 can be bunched only into box no 1 (since, height of the box no 2 is not equal to height of the packing item 1) using *HPM* and customer requirement can be satisfied with 8 corrugated boxes. Simultaneously, *HPM* is used to pack the packing item 2 into box no 2 with minimum unused packing volumes and customer requirement can be satisfied with 6 corrugated boxes. Then comparison has made with the company packing schedule.

Company packing schedule:

Packing item no	Dimensions of corrugated boxes	Number of boxes
1	525 mm × 460 mm × 260 mm	9
2	500 mm × 375 mm × 215 mm	6

Proposed *MBBA* packing schedule:

Packing item no	Dimensions of corrugated boxes	Number of boxes
1	525 mm × 460 mm × 260 mm	8
2	500 mm × 375 mm × 215 mm	5

IV. CONCLUSION

In this study, feasible packing patterns are generated using *HPM* and *SPM*. According to the case study given in Chapter 2, packing item 1 can be bunched only into box no 1 (because height of the box no 2 is not equal to height of the packing item 1) and *HPM* is the appropriate method to pack the chips bottles into corrugated box size 1. Similarly, packing item 2 can be packed using the same method into corrugated box size 2 by minimizing the unused packing volume. Comparing the solutions obtained from *MBBA* with the packing schedule of the company, it is clear that the less number of corrugated boxes needed to satisfy the customer demand using the proposed methods in this research than to the company schedule.

REFERENCES

[1]. Jatinder N. D Gupta and Johnny C. “A new heuristic algorithm for the one-dimensional bin-packing problem”, Production planning and Control, ISSN 0953-7287 (1999),Vol. 10, No. 6, 598-603.
[2]. Dokeroglu T., Bayir M. A and Cosar A. “Integer linear programming solution for the multiple query optimization problem”, Information Science and Systems, pp 51-60, Springer, Cham (2014). DOI: 10.1007/978-3-319-09465-6-6.
[3]. Mukhacheva E. A, G. N Belov, V.M. Kartack and Mukhacheva A.S. “Linear one-dimensional cutting-packing problems: Numerical experiments with the sequential value, Correction Method and a modified Branch and Bound method”.
[4]. Fleszar K. and Khalil S. “New Heuristics for one-dimensional bin packing”, Research-Gate: Computers and Operations Research, DOI: 10.1016/S0305-0548(00)00082-4.
[5]. Christian B. and Verena S. “Solving the 2D bin packing problem by means of a Hybrid Evolutionary Algorithm”, Elsevier, International Conference on Computational Science, ICCS (2013), 899-908.
[6]. Saad M. A. Suliman. “Pattern generating procedure for the cutting stock problem”, International Journal of Production Economics 74 (2001) 293-301.
[7]. W. N. P Rodrigo, W.B. Daundasekera, A.A.I Perera “Pattern Generation for Two-Dimensional Cutting Stock Problem” International Journal of Mathematics Trends and Technology (IJMTT), Vol. 3 Issue2 (2012), 54-62.
[8]. Rodrigo WNP, Daundasekera WB, Perera AAI, “Pattern Generation for Two-Dimensional Cutting Stock Problem with Location”, Indian Journal of Computer Science and Engineering, (IJCSE). 2012;3(2):354-368.
[9]. W. N. P. Rodrigo, W. B. Daundasekera and A. A. I. Perera, “*Modified Method for One-dimensional Cutting Stock Problem*”, Science Publishing Group 2015; 3(3); 12-17.
[10]. Niluka Rodrigo and Shashikala S., “One-Dimensional Cutting Stock Problem with Cartesian Coordinate Points”, International Journal of Systems Science and Applied Mathematics, vol 2(5), pp: 99-104, 2017.

Niluka P. Rodrigo, et. al. "Cylindrical Shape Bin Packing Problem with Cartesian Coordinate System.” *American Journal of Engineering Research (AJER)*, vol. 9(8), 2020, pp. 127-133.