Global Journal of Science Frontier Research: F
Mathematics and Decision Sciences
Volume 20 Issue 8 Version 1.0 Year 2020
Type : Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

Modified Integer Linear Programming Model for Bin-Packing Problems

By Niluka P. Rodrigo, W. B. Daundasekera & A. A. I. Perera
University of Peradeniya

Abstract- In this paper, our objective is to compare the solutions obtained from develop mathematical model and general model for Two-Dimensional Bin Packing Problem (*BPP*) with different sizes of boxes. It has become one of the most important mathematical applications throughout the time. In our study, Modified Branch and Bound Algorithm (*MBBA*) is developed to generate all the feasible packing patterns of given boxes to required containers for Two-Dimensional *BPP*. A computer program is developed using Matlab software package to generate feasible packing patterns and optimum packing plan. As a case study, three different sizes of rectangular shape packing items are selected with given demand in order to pack into two different sizes of pallets. Applying the proposed algorithm, demand is satisfied and total unused area significantly minimized. Accordingly, proposed mathematical model is more appropriate for medium size two dimensional *BPP*.

Keywords: *bin-packing problem, modified branch and bound algorithm, integer linear programming model, matlab software environment.*

GJSFR-F Classification: *MSC 2010: 15A86*

MODIFIEDINTEGERLINEARPROGRAMMINGMODELFORBINPACKINGPROBLEMS

Strictly as per the compliance and regulations of:

Modified Integer Linear Programming Model for Bin-Packing Problems

Niluka P. Rodrigo [α], W. B. Daundasekera [σ] & A. A. I. Perera [ρ]

Abstract- In this paper, our objective is to compare the solutions obtained from develop mathematical model and general model for Two-Dimensional Bin Packing Problem (*BPP*) with different sizes of boxes. It has become one of the most important mathematical applications throughout the time. In our study, Modified Branch and Bound Algorithm (*MBBA*) is developed to generate all the feasible packing patterns of given boxes to required containers for Two-Dimensional *BPP*. A computer program is developed using Matlab software package to generate feasible packing patterns and optimum packing plan. As a case study, three different sizes of rectangular shape packing items are selected with given demand in order to pack into two different sizes of pallets. Applying the proposed algorithm, demand is satisfied and total unused area significantly minimized. Accordingly, proposed mathematical model is more appropriate for medium size two dimensional *BPP*.

Keywords: bin-packing problem, modified branch and bound algorithm, integer linear programming model, matlab software environment.

I. Introduction

Competence of production system becomes a key factor of the success in today's competitive manufacturing environment and industries. Productivity can be enhanced by minimizing waste, lead time and hence reducing the cost of manufacture. For that reason, Operations Research plays a major role in minimizing manufacture waste. Many people including scientists have contributed and engaged in their research and other activities to overcome above challenges. *BPP* is become an important way of optimization of resources in manufacturing and trading industries. *BPP* could be one-dimensional (1D), two-dimensional (2D) and three-dimensional (3D) packing which gives optimum layout of packing items such that to improve the utilization ratio of bins. In other word; minimizing the bin's slack without overlapping the packages. Output obtains by BPP optimize all the aspects of resources while meeting the given supply-demand by minimizing wastage leading to reduction of cost of production. The *BPP* can be interpreted as a finite collection of items with varying specifications to be packed into one or more box utilizing the maximum volume of the box while satisfying the supply-demand. Each container can hold any subset of the collection of objects without exceeding its capacity (customer requirements). Bin packing is known 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. Researchers have worked on the *BPP* and used different approaches to solve the existing problem.

Among them, Gilmore et al [1] conducted some of the earliest research in this area and one-dimensional cutting stock problem is solved using *Linear Programming Technique*. In this study, unlimited numbers of raw materials with different lengths are assumed available in stock, and a mathematical model has been developed to minimize

Author α σ ρ: Department of Mathematics, Faculty of Science, University of Peradeniya, Sri Lanka. e-mails: nilukarodrigo@yaho.co.uk, wbd@gmail.com, athula103@yahoo.com

Ref

1. P. C. Gilmore and R. E. Gomory, "*A Linear Programming Approach to the Cutting Stock Problem*", Operations Research, Vol. 9, No. 2 (1961), 849 - 859.

the total cutting cost of the stock length of the feasible cutting patterns and *Column Generation Algorithm* has been developed to generate feasible cutting patterns. Then, Gilmore has claimed that feasible cutting patterns are increased with the required cutting items and *Linear Programming Technique* is not applicable to solved mathematical model with too many variables. Gilmore [2] has made an approach for one-dimensional cutting stock problem as an extended of earliest paper (Gilmore et al (1961)) and cutting stock problem has been described as a NP-hard problem. A new and rapid algorithm for the knapsack problem and changes in the mathematical formulation1 has been evolved and Gilmore has explained the procedure of the *Knapsack Method* using a test problem. 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) [3].

Christian Blum and Verena Schmid (2013) have dealt with two-dimensional *BPP* 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 [4]. Puchinger has described a combined genetic algorithm/branch & bound approach for solving a real world glass-cutting problem. The Genetic Algorithm uses an order-based representation, which is decoded using a greedy heuristic. The Branch & Bound algorithm was applied with a certain probability enhancing the decoding phase by generating locally optimal sub patterns. Reported results indicate that the approach of occasionally solving sub patterns to optimality may increase the overall solution quality [5]. 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 [6]. 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 [7]. Other than that, Octarina et al (2019) has used Branch and Bound Algorithm and Gilmore and Gomary model on two dimensional multiple stock size cutting stock problem. A case study has discussed with four stock sizes and five products. Optimum packing schedule illustrated for each stock [8].

Further, Fernandez (2010) has presented a method to find Integer Solutions to One-dimensional CSP to minimize the trim loss or to minimize the number of master rolls needed to satisfy the orders. Integer linear programming models for the One-dimensional CSP with different objective cost functions have been considered. Fernandez has studied an approach based on the classical column-generation procedure by Gilmore and Gomary for solving the linear programming (LP) relaxations. Also to obtain an integer solution, a final integer CSP after using an extra column-generation procedure has been solved [9]. Also, 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 [10].

In this paper, packing items with rectangular shapes are selected to nest into pallets with rectangular shape base. An algorithm based on *MBBA* and a computer program using Matlab software package to generate feasible packing patterns and to solve the formulated integer linear programming model is used. Then, the optimum packing schedule obtained from *MILPP* with the results of *BBA* with Gilmore & Gomory for *BPP* are compared.

Ref

2. P. C. Gilmore and R. E. Gomory, "*A Linear Programming Approach To The Cutting Stock Problem – Part II*", Operations Research, Vol. 11, No. 6 (1963), 863 – 888.

II. Materials and Methods

a) Bin-Packing Problem

Any firm's main objective is to maximize the annual contribution margin accruing from its production and sales. By reducing wastages and maximizing sales, productivity can be improved. Two-dimensional *BPP* can be defined as packing rectangular shape boxes with known dimensions into the base of the container while minimizing the unused area in order to meet the given demand.

Notes

b) Gilmore and Gomory Model

A mathematical model is formulated based on the concept of packing patterns to satisfy the requirements of each item while minimizing the unused packing area or volume:

Mathematical Model (Gilmore and Gomory, 1961)

$$\text{Minimize} \quad z = \sum_{k=1}^{r} \sum_{j=1}^{n} c_{jk} x_{jk} \quad (\text{Total Unused area})$$

$$\text{Subject to} \quad \sum_{j=1}^{n} p_{ijk} x_{jk} \geq d_i \quad \text{for all } i = 1,2,\ldots,m \quad (\text{Demand Constraints})$$

$$x_{jk},\ p_{ijk} \geq 0 \text{ and integer for all } i, j, k.$$

where r = Number of containers,
m = Number of items
n = Number of patterns,
p_{ijk} = The number of occurrences of the i^{th} item in the j^{th} pattern of the k^{th} container,
c_{jk} = Unused area for each j^{th} pattern of the k^{th} container,
d_i = Demand of the i^{th} item,
x_{jk} = The number of containers used to be packed according to the j^{th} pattern.

According to the mathematical model developed by Gilmore and Gomory, total unused area is minimized with satisfying the customer requirements. Sometimes, it is impracticable to obtain the optimal packing schedule only minimizing the unused area. Because, the total number of pieces of each packing item should be satisfy at least the demand of each packing item. That is there can be more pieces than to the demand of packing items in the optimal packing schedule. In this context, allocating extra pieces of each packing item is wastage.

In order to rectify the above disparity, mathematical model has developed to meet the exact demand.

Modified Integer Linear Programming Problem (MILPP) for BPP:

$$\text{Minimize} \quad z = \sum_{k=1}^{r} \sum_{i=1}^{m} \sum_{j=1}^{n} \left[(a_{ijk} x_{jk}) - (d_i' x_{jk}) \right]$$

$$(\text{Total packing items used from each item})$$

$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ijk} x_{jk} \geq d_i \quad \text{for all } i = 1,2,\ldots,m$$

$$(\text{Demand Constraints})$$

$$k = 1,2,\ldots,r$$

$$x_{jk},\ a_{ijk} \geq 0 \text{ and integer for all } i, j, k$$

where r = Number of containers,
m = Number of pieces,
n = Number of patterns,
a_{ijk} = The number of occurrences of the i^{ht} item in the j^{th} pattern of the k^{th} container,
c_{jk} = Unused packing area for each j^{th} pattern of the k^{th} container,
d_i = Demand value of the i^{th} item,

x_{jk} = The number of containers should be used according to the j^{th} pattern from the k^{th} container.

c) Modified Branch and Bound Algorithm in stepwise form:

Step 1: Arrange required lengths, Lk, $k = 1, 2, \ldots, r$ in decreasing order, ie $L_1 > L_2 > \ldots > L_r$, where r = number containers.

Arrange required widths, Wk, $k = 1, 2, \ldots, r$ according to the corresponding length Lk, $k = 1, 2, \ldots, r$.

Step 2: Arrange required lengths, l_i, $i = 1, 2, \ldots, m$ in decreasing order, ie $l_1 > l_2 > \ldots > l_m$, where m = number of packing items.

Arrange required widths, w_i, $i = 1, 2, \ldots, m$ according to the corresponding length l_i, $i = 1, 2, \ldots, m$.

Step 3: For $i = 1, 2, \ldots, m$ and $j = 1$ do Steps 4 to 6.

Step 4: Set $a_{111} = \left[\left[L_1 / l_1 \right]\right]$;

$$a_{ijk} = \left[\left[\left(L_k - \sum_{z=1}^{i-1} a_{zjk} l_z \right) \Big/ l_i \right]\right], \qquad (B.\ 1)$$

where L_k is the length of the base of the k^{th} container.

Here, a_{ijk} is the number of pieces of the i^{th} packing items in the j^{th} pattern of the k^{th} container along the length of the k^{ht} container and $[[y]]$ is the greatest integer les than or equal to y.

Step 5: If $a_{ijk} > 0$, then set

$$b_{ijk} = \left[\left[W_k / w_i \right]\right], \qquad (B.\ 2)$$

else set $b_{ijk} = 0$,
where W_k is the width of the base of the k^{ht} container.

Here, b_{ijk} is the number of pieces of the i^{th} packing item in the j^{th} pattern along the width of the base of the k^{th} container.

Step 6: Set $p_{ijk} = a_{ijk} \times b_{ijk}$,

where p_{ijk} is the number of pieces of the i^{th} item in the j^{th} pattern of the k^{th} container.

Step 7: Unused packing area

(i) Unused packing area along the length of the k^{ht} container:

$$c_{uk} = \left(L_k - \sum_{i=1}^{m} a_{ijk} l_i \right) \times W_k$$

For $i = 1, 2, \ldots, m$

If $\left(L_k - \sum_{i=1}^{m} a_{ijk} l_i \right) \geq w_i$ and $W_k \geq l_i$, then

Notes

(Considering 90° rotation for the given packing items)

$$\text{set } A_{ijk} = \left[\left[\left(L_k - \sum_{i=1}^{m} a_{ijk}\, l_i\right)\Big/ w_i\right]\right];$$

Notes

$$B_{ijk} = \begin{cases} \left[\left[W_k \big/ l_i\right]\right], & \text{if } A_{ijk} > 0 \\ 0\,, & \text{otherwise.} \end{cases}$$

$$p_{ijk} = p_{ijk} + \left(A_{ijk} \times B_{ijk}\right).$$

$$\text{else set } A_{ijk} = 0;$$

$$B_{ijk} = 0;$$

$$P_{ijk} = P_{ijk}.$$

If $A_{ijk} > 0$, then

$$\text{set } C_{uk} = \left[\left(L_k - \sum_{i=1}^{m} a_{ijk}\, l_i\right) - A_{ijk}\, w_i\right] \times B_{ijk}\, l_i\,;$$

$$C_{vk} = \left(L_k - \sum_{i=1}^{m} a_{ijk}\, l_i\right) \times \left(W_k - B_{ijk} l_i\right).$$

$$\text{else} \quad C_{uk} = \left(L_k - \sum_{i=1}^{m} a_{ijk}\, l_i\right) \times W_k\,,$$

where, A_{ijk} and B_{ijk} are the number of pieces of the i^{th} item in the j^{th} pattern of the k^{ht} container along the length and width of the c_{uk} rectangle respectively and C_{uk} and C_{vk} are the total unused area along the length and width of the c_{uk} rectangle respectively.

(ii) Unused area along the width of the container:

$$c_{vk} = \left(a_{ijk} \times l_i\right) \times K_{ijk}\,.$$

$$\text{Here, } K_{ijk} = W_k - \left(b_{ijk} \times w_i\right);$$

$$\text{If } \left(b_{ijk} \times w_i\right) = 0, \text{ then}$$

$$\text{set } K_{ijk} = 0\,,$$

where K_{ijk} is the remaining width of each item in each pattern.

For $z \neq i$

If $\left(a_{ijk} \times l_i\right) \geq l_z$ and $K_{ijk} \geq w_z$, then

$$\text{set } A_{zjk} = \left[\left[\left(\left(a_{ijk} \times l_i\right)\big/ l_z\right)\right]\right];$$

$$B_{zjk} = \begin{cases} \left[\left[\left(K_{ijk} / w_z \right)\right]\right], & \text{if } A_{zjk} > 0 \\ 0, & \text{otherwise.} \end{cases}$$

$$p_{zjk} = p_{zjk} + \left(A_{zjk} \times B_{zjk}\right)$$

$$\text{else set } A_{ijk} = 0;$$

$$B_{ijk} = 0;$$

$$P_{ijk} = P_{ijk}.$$

If $A_{zjk} > 0$, then

$$\text{set } C_{uk} = \left(\left(a_{ijk} \times l_i\right) - \left(A_{zjk} \times l_z\right)\right) \times B_{zjk}\, w_z;$$

$$C_{vk} = \left(a_{ijk} \times l_i\right) \times \left(K_{ijk} - \left(B_{zjk} \times l_z\right)\right).$$

$$\text{else } C_{vk} = \left(a_{ijk} \times l_i\right) \times K_{ijk},$$

where, A_{zjk} and B_{zjk} are the number of pieces of the i^{th} item in the j^{th} pattern of the k^{ht} container along the length and width of the c_{vk} rectangle respectively. C_u and C_v are the total unused area along the length and width of the c_{vk} rectangle respectively.

Step 8: Set $r = m - 1$.

While $r > 0$, do Step 9.

Step 9: While $a_{rjk} > 0$

set $j = j + 1$ and do Step 10.

Step 10: If $a_{rjk} \geq b_{rjk}$, then generate a new pattern according to the following conditions:

$$\text{For } z = 1, 2, \ldots, r-1$$

$$\text{set } a_{zjk} = a_{z(j-1)k};$$

$$b_{zjk} = b_{z(j-1)k}.$$

$$\text{For } z = r$$

$$\text{set } a_{zjk} = a_{z(j-1)k} - 1;$$

$$\text{if } a_{zjk} > 0, \text{ then set } b_{zjk} = \left[\left[W_k / w_z \right]\right];$$

$$\text{else set } b_{zjk} = 0.$$

For $z = r+1, \ldots, m$
calculate a_{zjk} and b_{zjk} using Equations (2.3) and (2.4). Go to Step 5.
else generate a new pattern according to the following conditions:

$$\text{For } z = 1, 2, \ldots, r-1$$

$$\text{set } a_{zjk} = a_{z(j-1)k};$$

$$b_{zjk} = b_{z(j-1)k}.$$

$$\text{For } z = r$$

$$\text{set } a_{zjk} = a_{z(j-1)k};$$

$$b_{zjk} = b_{z(j-1)k} - 1.$$

Notes

For $z = r+1, \dots, m$

calculate a_{zjk} and b_{zjk} using Equations (2.3) and (2.4).
Go to Step 6.

Step 11: Set $r = r - 1$.

Step 12: STOP.

d) Case Study

Proposed *MBBA* to solve one-dimensional BPP is tested and analyzed to determine feasible and optimal packing patterns. Following examples will illustrate how to generate feasible packing patterns by minimizing the total bin slack. Following data table represents two one-dimensional bin packing problem given in the paper [8] and proposed algorithm in this paper has been applied to solve those problems.

Table 1: The sizes of the stocks

*i*th *stock*	*Length (inches)*	Width *(inches)*
1	24	14
2	24	13
3	18	10
4	13	10

Table 2: The sizes and demands of the products

*i*th *product*	Length (inches)	Width (inches)	Number of Demand
1	8	5	2
2	7	4	3
3	5	3	2
4	4	2	5
5	2	1	5

III. Results and Discussion

MBBA is applied to the above examples to generate feasible packing patterns.

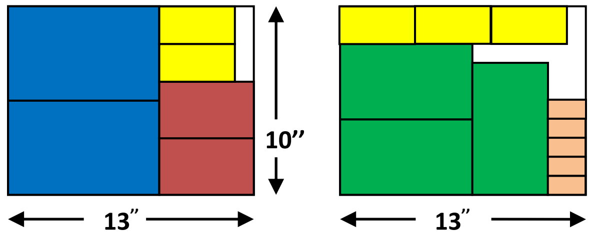
Effectiveness of the proposed *Integer Linear Programming Problem* has been compared with the results given in the paper [8].

Table 1: No of bins for problems given in the paper [8]

No of Containers from different algorithms with LP models / *i*th *stock*	*MBBA* with *MILPP*	BBA with Gilmore & Gomory model
1	1	1
2	1	1
3	2	2
4	2	3

Ref

8. Sisca Octarina, Vinny Ananda, Evi Yuliza, "Gilmore and gomory model on two dimensional multiple stock size cutting stock problem", Sriwijaya International Conference on Basic and Applied Science, J. Phys.: Conf. Ser. 1282 (2019) 012015. doi: 10.1088/1742-6596/1282/1/012015



IV. Conclusion

In this study, feasible packing patterns are generated using *MBBA* and the integer linear programming model developed by Gilmore and Gomory and *MILPP* are applied to generate the optimum packing schedule. According to the optimum solution obtain from *MILPP*, only 2 plates of 13 inches 10 inches are used to fulfil the customer demand while 3 plates of 13 inches 10 inches are used with Gilmore and Gomory model. Comparing the solutions obtained from *MBBA* with *MILPP* and *BBA* with Gilmore and Gomory model, it is clear that the less number of plates needed to satisfy the customer demand using the proposed method in this research than to the *BBA* with Gilmore and Gomory model [8].

References Références Referencias

1. P. C. Gilmore and R. E. Gomory, "*A Linear Programming Approach to the Cutting Stock Problem*", Operations Research, Vol. 9, No. 2 (1961), 849 - 859.
2. P. C. Gilmore and R. E. Gomory, "*A Linear Programming Approach To The Cutting Stock Problem – Part II*", Operations Research, Vol. 11, No. 6 (1963), 863 – 888.
3. 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.
4. 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.
5. Jakob Puchinger, Gunther R. Raidl and Gabriele Koller, "*Solving a Real-World Glass Cutting Problem*".
6. 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.
7. 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.
8. Sisca Octarina, Vinny Ananda, Evi Yuliza, "Gilmore and gomory model on two dimensional multiple stock size cutting stock problem", Sriwijaya International Conference on Basic and Applied Science, J. Phys.: Conf. Ser. 1282 (2019) 012015. doi: 10.1088/1742-6596/1282/1/012015
9. L. Fernjandez, L. A. Fernjandez, C. Pola, "*Integer Solutions to Cutting Stock Problems*", 2nd International Conference on Engineering Optimization, Sep 6-9, 2010.
10. 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.

Ref

8. Sisca Octarina, Vinny Ananda, Evi Yuliza, "Gilmore and gomory model on two dimensional multiple stock size cutting stock problem", Sriwijaya International Conference on Basic and Applied Science, J. Phys.: Conf. Ser. 1282 (2019) 012015. doi: 10.1088/1742-6596/1282/1/012015.