IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 19, Issue 1 Ser. II (Jan. – Feb. 2023), PP 29-34
www.iosrjournals.org

Approximation Method for Solving TwoDimensional Cutting-Stock Problems

Abdul Rahman[1], Ihda Hasbiyati[2], M. D. H. Gamal[3]
[1]*(Department of Mathematics, Faculty of Mathematics and Science, Universityof Riau,Indonesia)*
[2]*(Department of Mathematics, Faculty of Mathematics and Science, University of Riau,Indonesia)*
[3]*(Department of Mathematics, Faculty of Mathematics and Science, University of Riau,Indonesia)*

Abstract:
The problem of cutting two-dimensional stock is a problem where the cutting pattern considers the length and width of a rectangular stock. The objective is to minimize the wastage which is equivalent to minimizing the number of stocks used. This article discusses the solution to the problem of cutting two-dimensional stocks assuming that the supply is unlimited, and demands requested when cutting cannot be rotated. Then, the column generating technique is implemented to obtain a knapsack problem, and the knapsack is solved by dynamic programming method. The result obtained is a best combination of cutting patterns, so it minimizes the cutting wastages.
***Key Word**: Column generating; Cutting pattern; Knapsack problem; Rectangular stock; Two dimensional cutting-stock.*

Date of Submission: 10-02-2023 Date of Acceptance: 22-02-2023

I. Introduction

Linear programming is a technique from operations research to find solutions to optimization problems using linear equations and inequalities in order to obtain optimal solutions taking into account the existing constraints. In solving a problem, linear programming uses a mathematical model. This model presents the form and arrangement of the problem to be solved. The linear programming develops and provides a strong impetus for industrial progress. One of its applications is to solve the cutting-stock problems. The cutting-stock problem is one of the optimization problems that basically reduces a linear programming into a linear programming with integer values.

The problem is called as a cutting-stock if there are limited demands or orders with various sizes while ignoring the other constraints. The demands are fulfilled by cutting the stocks with one or more standard sizes. To form a linear programming as the main problem, the possible objective function is to minimize the total amount of stocks with standard size cut.

One dimensional cutting-stock problemconsiders only one type of cutting, namely length or width. Regarding one dimensional cutting-stock, Gilmore and Gomory[1]use a linear programming approach where stocks are available in various standard sizes $L_1, L_2, \dots, L_k$using the cost function and then solve it using the dynamic programming method. Similarly, Winston[2]discusses a linear programming approach to solve the onedimensional cutting-stock where the cutting patterns are generated using the column generating technique with the help of the branch-and-bound method. In addition, Wongprakornkul and Charnsethikul[3]solve the problem of cutting one-dimensional stock with limited demands and objective capacity planning.

The two dimensional cutting-stock problem is an issue where the cutting patterns used consider the length and width of a rectangular stock. The previous papers on these two dimension cases include[4], [5], [6], [7], [8] and [9]. The problem of cutting two-dimensional stock also uses a column generation technique which is solved using a knapsack problem. Several articles have been written regarding the use of knapsack in this two-dimensional problem[10], [11], [12], [13] and [14].

This article discusses how to solve the problem of cutting two-dimensional stock with the assumption that cutting is only done on one type of stock measuring length, while width and height are ignored. Then the availability of supply is unlimited, and demands are fulfilled by cutting the standard stocks where the cutting cannot be rotated. Then the column generation technique is used to obtain the knapsack shape, and the knapsack problem is solved by considering the knapsack problem in one dimension with the dynamic programming method. The first solves the knapsack problem by making the problem case-by-case basis by using a width constraint. Then, the value of the objective function of each case is used to solve the knapsack problem with length constraints by making the knapsack problem case-by-case.

II. Methods

Column Generation Technique

Column generation is an iterative algorithm that starts with a small set of initial patterns, and then cleverly chooses new columns to add to the main cutting-stock problem so that we find the optimal solution without having to enumerate every column. Columns in the cutting-stock mathematical model indicate the cutting patterns. The column generation method takes advantage of the effectiveness of the revised simplex method in the cutting-stock problems, so that many of the processing steps refer to the revised simplex method.

Let us consider the problem of cutting iron plate with standard length L and standard width W which will be cut into length $l_i$and width w_i for $i = 1, 2, \dots, m$ with the number of orders as many as b_i. Variable x_j denotes the number of standard stocks cut according to pattern or activity j, a_{ij} is the number of standard stocks with length $l_i$and widthw_i resulting from activity j. So the general form of the cutting-stock problem is

$$\begin{aligned} &\min z = x_1 + x_2 + \cdots + x_n, \\ &\text{subject to } a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i, i = 1, 2, \dots, m, \\ &\qquad x_j \geq 0 \text{ and integer}, j = 1, 2, \dots, n. \end{aligned} \tag{1}$$

Equation (1) can be written in matrix form and in standard form as follows:

$$\min z = c_B^T x_B + c_B^T x_N,$$

$$\text{subject to} Bx_N + Nx_N = b,$$

$$x_B, x_N \geq 0 \text{ and integer}$$

where x_B and x_N are basic variables and nonbasic variables, B dan N are the column matrices corresponding to the variables x_B and x_N. During the simplex method iterations, suppose that the associated basis is $B = (A_1, A_2, \dots, A_m)$ where A_i is a column vector of dimension m, $c_B^T = (c_1, c_2, \dots, c_m)$ is the coefficient of the objective function that corresponds to $A_1, A_2, \dots, A_m$. Then from the linear programming, the promising cutting pattern j is the one whose *reduced cost*

$$z_j - c_j = c_B^T B^{-1} A_j - c_j$$

is positive for the minimization problem, where

$$A_j = \left(a_{1j}, a_{2j}, \dots, a_{mj}\right)^T$$

is a vector indicating the number of cuts with l_i and w_i resulting from the cutting pattern j.

The pattern that gives an advantage is the one whose value of $z_j - c_j$is positive for the minimization problem. For this reason, the column generation method is needed, namely a technique to obtain a most promising column or pattern. In this case, it is equivalent to solving the subproblem

$$\begin{aligned} &\max v = d_1y_1 + d_2y_2 + \cdots + d_my_m - 1 \\ &\text{subject to} \quad r_1y_1 + r_2y_2 + \cdots + r_my_m \leq R \\ &\qquad y_i \geq 0 \text{ and integer}, \end{aligned} \tag{2}$$

where d_i is the dual value of i from $c_B^T B^{-1}$, y_i is nonnegative integer, r_i is amount for order i and $R$is standard size. Problem (2) is also called as *knapsack* problem.

Knapsack Problem

The knapsack problem is an integer linear programming problem with one constraint. The knapsack problem is used to find the best solution for all possible items to be packed into the sack. For this reason, the dynamic programming method is used in solving the knapsack subproblem. Taha[15]explaines that the dynamic programming is to determine the optimal solution of a multivariable problem by decomposing it into several stages, each stage consisting of a single variable subproblem. The following are the steps in solving the knapsack problem using the dynamic programming method in the two-dimensional cutting stock problem:

1. Sorting the length size of the demand from the smallest to the largest and the width of the demand follows the order of the length size.
2. Solving the knapsack problem using the width constraint. The following is the linear programming form of the knapsack problem with initial truncation using the width constraint:

$$\begin{aligned} &\max v = d_1y_1 + d_2y_2 + \cdots + d_my_m - 1, \\ &\text{subject to} w_1y_1 + w_2y_2 + \cdots + w_my_m \leq W, \\ &\qquad y_i \geq 0 \text{ and integer.} \end{aligned} \tag{3}$$

3. Solving subproblem (3) by dividing case by case, namely $w_1y_1 \leq W$, $w_1y_1 + w_2y_2 \leq W, \dots, w_1y_1 + w_2y_2 + \cdots + w_my_m \leq W$with dynamic programming method.
4. Obtaining the objective function value from each case, for example with $e_1, e_2, \dots, e_n$. The objective function values $e_1, e_2, \dots, e_n$are used to solve the knapsack problem by using the length constraint. The following is the linear programming form of the knapsack problem using the length constraint:

$$\max v = e_1 y_1 + e y_2 + \cdots + e_m y_m - 1,$$
$$\text{subject to} l_1 y_1 + l_2 y_2 + \cdots + l_m y_m \le L, \quad (4)$$
$$y_i \ge 0 \text{ and integer.}$$

5. Solving subproblem (4) by dividing case by case, namely $l_1 \le L$, $l_1 y_1 + l_2 y_2 \le L, \ldots, l_1 y_1 + l_2 y_2 + \cdots + l_m y_m \le L$ with dynamic programming method.

III. Result

From the problem of cutting-stock, for example, a iron plate company produces iron plate in feet (ft) with the length of 12ft and the width of 10ft. Then, there are some orders requested as shown in Table no 1.

Tabel no 1: Order sample

Order (i)	Length used (ft)	Width used (ft)	Number of order (sheets)
1	3	4	22
2	4	5	14
3	7	3	18

The initial basis variables (initial basis) is obtained by selecting from pure truncation patterns. So from the pure cutting pattern, the initial basis for lengths 3ft × 4ft, 4ft × 5ft, and 7ft × 3ft yields

$$B_0 = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 3 \end{bmatrix} \text{ and } B_0^{-1} = \begin{bmatrix} \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix},$$

The dual value is obtained from the following calculation:

$$c_B B_0^{-1} = [1 \quad 1 \quad 1] \begin{bmatrix} \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & \frac{1}{6} & \frac{1}{3} \end{bmatrix},$$

for the new basis, a pattern verified by y_1, y_2 dan y_3. The value of $z_j - c_j$ is obtained as follows:

$$c_B B_0^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} - 1 = \frac{1}{8} y_1 + \frac{1}{6} y_2 + \frac{1}{3} y_3 - 1,$$

where y_1, y_2 and y_3 cannot exceed 10m, y_1, y and y_3 are nonnegative integers and patterns y_1, y_2 and $y_3$must satisfy

$$4y_1 + 5\, y_2 + 3y_3 \ge 10,$$
$$y_i \ge 0 \text{ and integer.}$$

To get a profitable pattern is by solving the equivalent knapsack problem

$$\max v = \frac{1}{8} y_1 + \frac{1}{6} y_2 + \frac{1}{3} y_3 - 1,$$
$$\text{subject to } 4y_1 + 5y_2 + 3y_3 \le 10, \quad (5)$$
$$y_i \ge 0 \text{ and integer.}$$

The knapsack problem from subproblem (5) is a knapsack problem using the width constraint. This problem is solved by using dynamic programming method.

To solve this problem, starting from the initial stage$(i = 1)$ to the final stage or moving forward.

Case 1: $4y_1 \le 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 \le 10$ is $y_1 = 2$ and $e_1 = v = \frac{2}{8}$.

Case 2: $4y_1 + 5y_2 \le 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 + 5y_2 \le 10$ is $y_1 = 0,\ y_2 = 2$ and $e_2 = v = \frac{2}{6}$.

Case 3: $4y_1 + 5y_2 + 3y_3 \le 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 + 5y_2 + 3y_3 \le 10$ is $y_1 = 0, y_2 = 0, y_3 = 3$ and $e_3 = v = \frac{3}{3} = 1$.

Then, solving the knapsack problem using the length constraint and the objective function value of each case obtained from solving the knapsack problem using the width constraint, so that the equation of the knapsack problem is as follows

$$\max v = \tfrac{2}{8}y_1 + \tfrac{1}{3}y_2 + 1y_3 - 1,$$
$$\text{subject to } 3y_1 + 4y_2 + 7y_3 \leq 12, \quad (6)$$
$$y_i \geq 0 \text{ and integer.}$$

To solve equation (6), starting from the initial stage ($i = 1$) to the final stage or moving forward.

Case 1: $3y_1 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 \leq 12$ is $y_1 = 4$ dan $v = v_1 = 0$.

Case 2: $3y_1 + 4y_2 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 + 4y_2 \leq 12$ is $y_1 = 0, y_2 = 3$ and $v = v_2 = 0$.

Case 3: $3y_1 + 4y_2 + 7y_3 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 + 4y_2 + 7y_3 \leq 12$ is $y_1 = 0, y_2 = 1, y_3 = 1$ and $v = v_3 = \frac{1}{3}$.

from cases 1, 2 and 3 which are obtained by using the length constraint, the maximum value of the objective function v is found in the 3rd case with $v = v_3 = \frac{1}{3}$, so that the optimal solution of the knapsack problem from equation (7) is obtained by an integer solution, namely $y_1 = 0$, $y_2 = 1$ and $y_3 = 1$which results in $v = \frac{1}{3}$. Then, because the knapsack solution with length constraints corresponds to the knapsack solution with width constraints $y_1 = 0, y_2 = 2$ and $y_3 = 3$, so that a new value is obtained, $y_1 = 0 \times 0 = 0$, $y_2 = 1 \times 2 = 2$ and $y_3 = 1 \times 3 = 3$.

The cutting pattern $y_1 = 0$, $y_2 = 2$ and $y_3 = 3$is not optimal yet, because the value of $v = \frac{1}{3}$. Next, a pattern will be sought that will provide an advantage by including the cuts that have been obtained into the basis, for this it is necessary to form a new right-hand side and a new column

New column of $x_4 = B_0^{-1}\begin{bmatrix}0\\2\\3\end{bmatrix} = \begin{bmatrix}\frac{1}{8} & 0 & 0\\0 & \frac{1}{6} & 0\\0 & 0 & \frac{1}{3}\end{bmatrix}\begin{bmatrix}0\\2\\3\end{bmatrix} = \begin{bmatrix}0\\\frac{1}{3}\\1\end{bmatrix}$,

New right side= $B_0^{-1}b = \begin{bmatrix}\frac{1}{8} & 0 & 0\\0 & \frac{1}{6} & 0\\0 & 0 & \frac{1}{3}\end{bmatrix}\begin{bmatrix}18\\14\\22\end{bmatrix} = \begin{bmatrix}\frac{11}{4}\\\frac{7}{3}\\6\end{bmatrix}$.

The ratio test shows that x_4 enters a new base in row 3, the new basis variable is BV = $\{x_1,\ x_2,\ x_4\}$, using the inverse product, we get

$$B_1^{-1} = E_0 B_0^{-1} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & -\frac{1}{3}\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{8} & 0 & 0\\0 & \frac{1}{6} & 0\\0 & 0 & \frac{1}{3}\end{bmatrix} = \begin{bmatrix}\frac{1}{8} & 0 & 0\\0 & \frac{1}{6} & -\frac{1}{9}\\0 & 0 & \frac{1}{3}\end{bmatrix}.$$

Now

$$c_B B_1^{-1} = [1 \quad 1 \quad 1]\begin{bmatrix}\frac{1}{8} & 0 & 0\\0 & \frac{1}{6} & -\frac{1}{9}\\0 & 0 & \frac{1}{3}\end{bmatrix} = \begin{bmatrix}\frac{1}{8} & \frac{1}{6} & \frac{2}{9}\end{bmatrix}.$$

Again, the column generation technique is used to determine the pattern that will enter the base. For the current dual value $c_B B_1^{-1}$, a pattern defined by y_1, y_2 and y_3 determines the $z_j - c_j$ value to

$$c_B B_1^{-1}\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix} - 1 = \tfrac{1}{8}y_1 + \tfrac{1}{6}y_2 + \tfrac{2}{9}y_3 - 1.$$

The equivalent knapsack problem is

$$\max v = \frac{1}{8}y_1 + \frac{1}{6}y_2 + \frac{2}{9}y_3 - 1,$$
$$\text{subject to } 4y_1 + 5\,y_2 + 3y_3 \leq 10, \quad (7)$$

$y_i \geq 0$ and integer.

The knapsack problem from equation (7) is solved by by using dynamic programming method using width as a constraint.

To solve this problem, starting from the initial stage $(i = 1)$ to the final stage or moving forward.

Case 1: $4y_1 \leq 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 \leq 10$ is $y_1 = 2$ and $e_1 = v = \frac{2}{8}$.

Case 2: $4y_1 + 5\,y_2 \leq 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 + 5\,y_2 \leq 10$ is $y_1 = 0, 2_2 = 2$ and $e_2 = v = \frac{2}{6} = \frac{1}{3}$.

Case 3:$4y_1 + 5\,y_2 + 3y_3 \leq 10$.

By using the dynamic programming method, the optimal solution for case $4y_1 + 5\,y_2 + 3y_3 \leq 10$ is $y_1 = 0, y_2 = 0, y_3 = 3$ and $e_3 = v = \frac{6}{9}$.

Then, solving the knapsack problem using the length constraint and the objective function value of each case obtained from solving the knapsack problem using the width constraint, so that the equation of the knapsack problem is as follows

$$\max v = \frac{2}{8}y_1 + \frac{1}{3}y_2 + \frac{6}{9}y_3 - 1,$$
$$\text{subject to } 3y_1 + 4y_2 + 7y_3 \leq 12, \qquad (8)$$
$$y_i \geq 0 \text{ and integer.}$$

To solve equation (8), starting from the initial stage $(i = 1)$ to the final stage or moving forward.

Case 1: $3y_1 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 \leq 12$ is $y_1 = 4$ and $v = v_1 = 0$.

Case 2: $3y_1 + 4y_2 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 + 4y_2 \leq 12$ is $y_1 = 0, y_2 = 3$ and $v = v_2 = 0$.

Case 3: $3y_1 + 4y_2 + 7y_3 \leq 12$.

By using the dynamic programming method, the optimal solution for case $3y_1 + 4y_2 + 7y_3 \leq 12$ is $y_1 = 0, y_2 = 1, y_3 = 1$ and $v = v_3 = 0$.

from cases 1, 2 and 3 which are obtained by using the length constraint, the maximum value of the objective function v is found in the 3rd case with $v = 0$, so that the optimal solution of the knapsack problem of equation (8) is obtained by an integer solution, namely $y_1 = 0,\ y_2 = 1$and $y_3 = 1$ which results $v = 0$. Then, because the knapsack solution with length constraints corresponds to the knapsack solution with width constraints $y_1 = 0,\ y_2 = 2$ dan $y_3 = 3$, so the new values $y_1, y_2$and y_3 obtained are $y_1 = 0 \times 0 = 0, y_2 = 1 \times 2 = 2$ and $y_3 = 1 \times 3 = 3$.

Because the solution $v = 0$, this means that there are no more profitable patterns if it is included in the basis because the basic variable is already optimal. The basic variable obtained is $\{x_1,\ x_2,\ x_4\}$. To determine the basic variables in the optimal solution, look for the value of the right-hand side as follows:

$$B_1^{-1}b = \begin{bmatrix} \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{6} & -\frac{1}{9} \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 22 \\ 14 \\ 18 \end{bmatrix} = \begin{bmatrix} \frac{11}{4} \\ \frac{1}{3} \\ 6 \end{bmatrix}.$$

IV. Discussion

The optimal solution for the cuttting-stock problem above is $x_1 = 11/4, x_2 = 1/3$and $x_4 = 6$. Integer solutions are obtained by rounding up, namely $x_1 = 3, x_2 = 1$and $x_4 = 6$. A Request of 22 sheets for size 3ft×4ft, 14 sheets for size 4ft ×3 ft and 18 sheets for size 7ft ×3ft can be fulfilled by cutting iron plate with the standard size 12ft ×10ft. Cutting process is done by adjusting the cutting knife. Initial cutting of the iron plate from the long side then to the wide side. In this case, the items demande of deduction cannot be rotated so that they are obtained as follows:

1. Cut 3 sheets of standard size iron plate 12 ft x 10 ft. First cut 12 ft with a length of 3 ft as many as 4 sheets then 10 ft with a width of 4 ft as many as 2 sheets so obtained 8 sheets of size 3 ft x 4 ft for each standard size of 12 ft x 10 ft. The result in total is 24 sheets size 3 ft x 4 ft.

2. Cut 1 sheet of standard size iron plate 12 ft x 10 ft. First cut 12 ft with a length of 4 ft as many as 3 sheets then 10 ft with a width of 5 ft as many as 2 sheets so obtained 6 sheets of size 4 ft x 5 ft for each standard size of 12 ft x 10 ft. The result in total is 6 sheets size 4 ft x 5 ft.
3. Cut 3 sheets of standard size iron plate 12 ft x 10 ft. First cut 12 ft with a length of 7 ft as many as 1 piece and 4 ft as many as 1 sheet then 10 ft with a width of 3 ft as many as 3 sheets and 5 ft as many as 2 sheets so obtained 3 sheets of size 7 ft x 3 ft and 2 sheets of size 4 ft x 5 ft for each standard size of 12 ft x 10 ft. The result in total is 18 sheets size 7 ft x 3 ft and 12 sheets size 4 ft x 5ft.

Based on the solution above, deviation of the required number of iron plate is about 25% from the result obtained from complete enumeration.

V. Conclusion

The problem of cutting a two-dimensional stock is a problem that considers length and width when cutting. Many patterns can be formed when making cuts. However, using the pattern column generation technique the best pattern is obtained. To get the best pattern, the knapsack sub-problem is solved first. Because the problem is two-dimensional, there are two knapsack subproblems, namely the length and width knapsack subproblems. Then, the dynamic programming method is used to solve them.

References

[1] P. C. Gilmore and R. E. Gomory, “A linear programming approach to the cutting-stock problem,” *Oper. Res.*, vol. 9, pp. 849–859, 1961.
[2] W. L. Winston, *Operations Research: Applications and Algorithms*, 4th ed. Belmont: Thomson Brooks/ Cole, 2004.
[3] S. Wongprakornkul and P. Charnsethikul, “Solving one-dimensional cutting stock problem with discrete demands and capacitated planning objective,” *J. Math. Stat.*, vol. 6, pp. 79–83, 2010.
[4] F. Furini, E. Malaguti, R. Medina Durán, A. Persiani, and P. Toth, “A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size,” *Eur. J. Oper. Res.*, vol. 218, pp. 251–260, 2012.
[5] S. Juttijudata and P. Sudjarittham, “Two-dimensional cutting stock problems with a modified column generation method,” *Curr. Appl. Sci. Technol.*, vol. 20, pp. 217–225, 2020.
[6] A. Mellouli, F. Masmoudi, I. Kacem, and M. Haddar, “A hgybrid genetic algorithm for optimization of two-dimensional cutting-stock problem,” *Int. J. Appl. Metaheuristic Comput.*, vol. 1, pp. 34–49, 2010.
[7] K. Novianingsih, R. Hadianti, and S. Uttunggadewa, “Column generation technique for solving two-dimensional cutting stock problems: method of stripe approach,” *J. Indones. Math. Soc*, vol. 13, pp. 161–172, 2007.
[8] X. Song and J. A. Bennell, “Column generation and sequential heuristic procedure for solving an irregular shape cutting stock problem,” *J. Oper. Res. Soc.*, vol. 65, pp. 1037–1052, 2017.
[9] S. Sumetthapiwat and B. Intiyot, “A column generation on two-dimensional cutting stock problem with fixed-size usable leftover and multiple stock sizes,” *Int. J. Logist. Syst. Manag. Manag.*, vol. 35, pp. 273–288, 2020.
[10] P. V. Carvajal, G. Camacho, D. C. Usaquen, and D. A. Martinez, “Solving the two-dimensional knapsack problem considering cutting-time and emission of particulate matter in the metalworking industry,” *IEEE Lat. Am. Trans.*, vol. 16, pp. 2888–2895, 2018.
[11] M. Dolatabadi, A. Lodi, and M. Monaci, “Exact algorithms for the two-dimensional guillotine knapsack,” *Comput. Oper. Res.*, vol. 39, pp. 48–53, 2012, doi: 018.
[12] M. A. Hatefi, “Developing column generation approach to solve the rectangular two-dimensional single knapsack problem,” *Sci. Iran.*, vol. 24, pp. 3287–3296, 2017.
[13] K. Kim, B. Kim, and H. Cho, “Multiple-choice knapsack-based heuristic algorithm for the two-stage two-dimensional cutting stock problem in the paper industry,” *Int. J. Prod. Res.*, vol. 52, pp. 5675–5689, 2014.
[14] K. J. Woo, “Application of fitness switching genetic algorithm for solving multidimensional knapsack problem,” *J. Theor. Appl. Inf. Technol.*, vol. 97, pp. 4837–4849, 2019.
[15] H. A. Taha, *Operations Research: An Introduction*, 8th ed. Newark: Person Education Hall, 2007.

Abdul Rahman. et. al. "Approximation Method for Solving TwoDimensional Cutting-Stock Problems." *IOSR Journal of Mathematics (IOSR-JM)*, 19(1), (2023): pp. 29-34.