Academic Journal of Applied Mathematical Sciences
ISSN(e): 2415-2188, ISSN(p): 2415-5225
Vol. 4, Issue. 3, pp: 15-21, 2018
URL: http://arpgweb.com/?ic=journal&journal=17&info=aims

Academic Research Publishing Group

Original Research Open Access

A Practical Application of the Generalized Cutting Stock Algorithm

W. D. D. Madhavee*
Department of Mathematics, University of Colombo, Colombo 3, Sri Lanka

N. Saldin
Department of Mathematics, University of Colombo, Colombo 3, Sri Lanka

U. C. Vaidyarathna
Department of Mathematics, University of Colombo, Colombo 3, Sri Lanka

C. J. Jayawardene
Department of Mathematics, University of Colombo, Colombo 3, Sri Lanka

Abstract

A watered-down version of the cutting stock algorithm has existed for a few centuries before the industrial revolution but no real formulation or solution to the problem was known other than for a few heuristic algorithms used under specific cases pertaining to the logging industry. The first formulations and solutions of the cutting stock problem was published about 6 decades ago by Gilmore and Gomory in the Operation research journal [1]. In that, they have explained the concept by using crude optimization techniques which are not applicable to most conditions found in the contemporary business environment. Our research project involves cut-ting wooden sheets and wooden rods of specific dimensions based on the requirements of the customers of Moratuwa Timber Work (MTW). The main focus of this paper is to find the optimal cutting patterns by minimizing the wastage and the trim loss. This is achieved with the aid of a web enabled database, using Java codes and Lingo programs.

Keywords: Cutting stock problem; Linear and integer programming; LINGO.
AMS Mathematics Subject Classification (2010): 90C10, 90C05, 90C90.

1. Introduction

The cutting stock algorithm relates to a wide range of industries. This situation has created an impetus with financial incentives to find more efficient algorithms to deal with different scenarios. In particular, if we consider the furniture industry where cutting wooden material play a pivotal role, many variations of the algorithms are used when cutting wood in order to reduce wastage. For example, if we need to produce wooden rods for chairs, we have to use the one-dimensional cutting method which requires the use of a single cutting blade. However, if we consider the two- dimensional or the three-dimensional cutting method, we will need to accommodate slanted or oblique cutting method which requires the use of wedged shape cutting tools, double blades or jigsaw blades. Therefore, in a more general setting, in order to implement our algorithm, we need to take in to consideration the additional constraints related to the cutting methods and the cutting equipment which need to be used.

In our research project, we look into furniture manufacturing in Moratuwa Timber Work (MTW). This requires cutting wooden sheets and wooden rods of specific dimensions. Our main focus in this paper was to find the optimal cutting patterns by minimizing the wastage. Our research project was done under the following assumptions.

1. All wooden pieces required to manufacture the furniture are cut from the sheets and rods that are exclusively in the current stock.
2. The stock consists of sufficient sheets and rods in order to manufacture the required furniture.

In the main part of the project, we solved the problem with the aid of a web enabled database, using Java codes and Lingo programs. More specifically, we implemented the program in 4 stages. In stage 1, the database was updated according to the customer input obtained via the web. In stage 2, a Java code was developed and implemented, in order to generate all possible cutting patterns. In stage 3, an Integer Program was formulated using the cutting patterns obtained in stage 2 and subsequently a Lingo Program was generated to update the database. In the final stage, which is stage 4, the updated database results were redirected to the customers through the web.

2. History

With the introduction of computer based solutions in nineteen seventies, different variations of this problem were tackled by many authors.

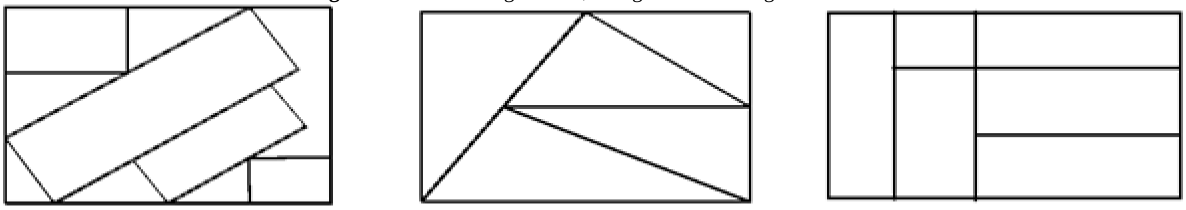
Some of the approaches that mushroomed in the following decades related to variations of this cutting plane algorithm are the Enumerative approach [2, 3], the Sequential approach (see [4, 5]), the Knapsack approach [6, 7], the Branch and Bound approach [8-11], the Synergistic approach [12, 13], the Dynamic Programming approach [14,

*Corresponding Author

15], the Tree search algorithm approach [16], the Genetic algorithms approach(see [17, 18], the Column Generation algorithm approach (see [9, 11, 16, 19] and the Vishwanath and Bagchi algorithm approach [20].

With the advancement of the industrial revolution, there was a massive demand for cutting different materials in different shapes and orientations.

Figure-1. A non-orthogonal cut, triangular cut and a guillotine cut



Some of the research papers that deal with cutting different shapes such as the non-orthogonal [21], triangular and the guillotine cuts [20, 22], are shown in the above figure.

3. Formulation of the Problem

The dimensions of an order to make furniture items received by MTW are listed below.

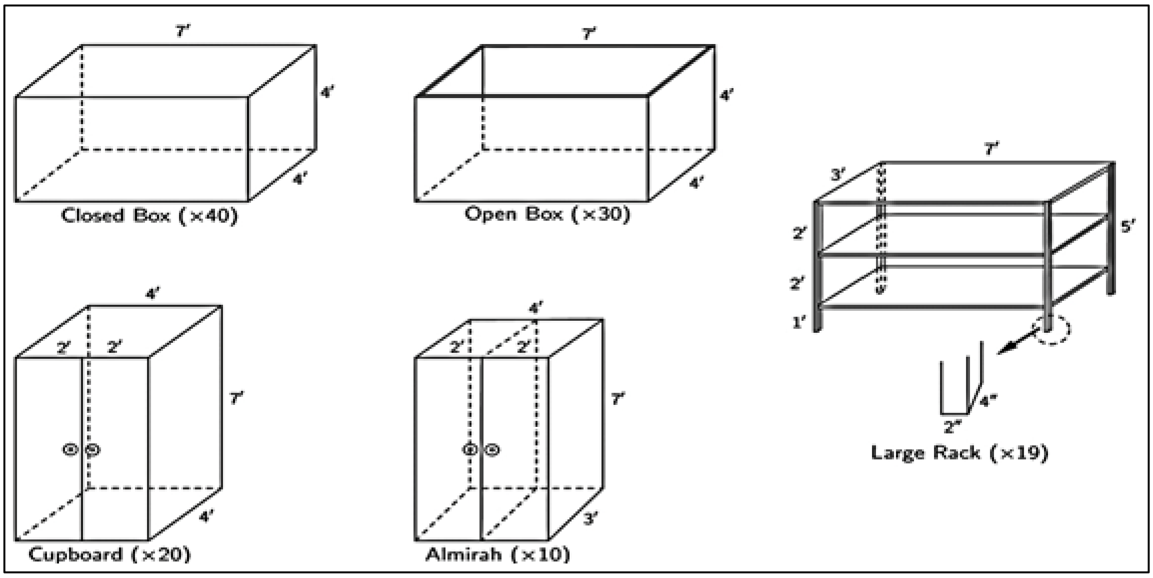
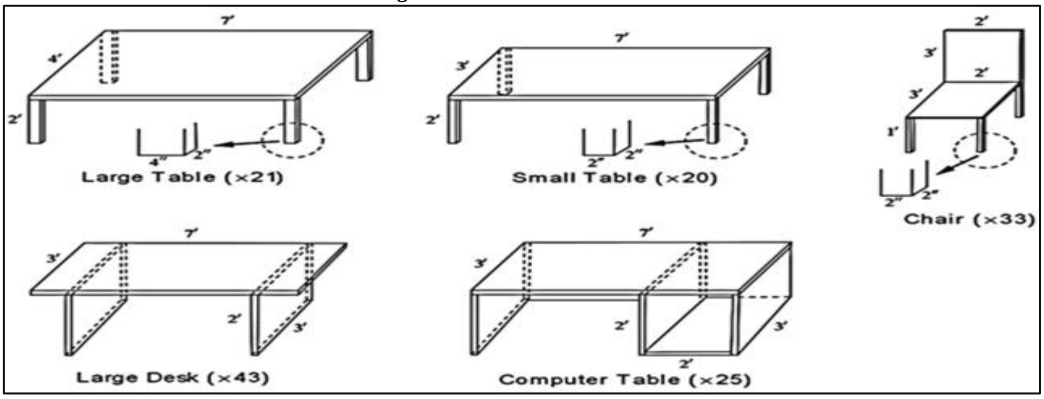


Table-1. Cutting patterns

Cutting Patterns	Variable
7'×4' 7' ×3' 4'×4' 7' ×2' 4'×3' 3'×2' (....)	x_i: The number of times the pattern generated by the tuple (cutting pattern) has been cut from the 10'×4' sheet
4'×4 4'×3' 3'×2' (......... ....)	y_i: The number of times the pattern generated by the tuple (cutting pattern) has been cut from the 6'×6' sheet
4'×3' 3'×2' (......... ....)	z_i : The number of times the pattern generated by the tuple (cutting pattern) has been cut from the 4'×3' sheet
2"×2"×2' 2"×2"×1' (............. ....)	p_i : The number of times the pattern generated by the tuple (cutting pattern) has been cut from the 2"×2"×5' rod
2"×4"×5' 2"×4"×2' (............. ....)	q_i: The number of times the pattern generated by the tuple (cutting pattern) has been cut from the 2"×4"×5' rod

Figure-2. Ordered furniture



The storage house of MTW has wooden sheets which are 1/2" thick with the dimensions 10'×4', 6'×6' and 4'×3' and wooden rods with the dimensions 2"×2"×5' and 2"×4"×5'. These sheets and rods will be used to manufacture the ordered items. In order to formulate the problem, the following variables are defined.

In order to identify all possible cutting patterns, the following assumptions were made regarding the method of cutting.

- We will start cutting the pieces from the left-hand corner of the sheet. The selection of the pieces to be cut will be in accordance with the descending order of the area, corresponding to the individual pieces.
- The sawdust produced by cutting has no effect on the wastage.
- In implementing the algorithm, initially we changed the orientation of the pieces (including the sheets) to have length greater than their width.

To generate a new cutting pattern, we first set up a heuristic algorithm by taking in to account that the total area of the individual pieces is less than the area of the wooden sheet. However, this heuristic algorithm didn't handle all possible cases satisfactorily. For example, we had a problem with cutting the pattern (1, 1, 0) which corresponds to cutting one 4'×4' piece and one 4'×3' piece from a 6'×6' sheet. Though this cutting pattern satisfies the constraint corresponding to the total area (i.e., 4'×4'+ 4'×3' ≤ 6'×6') (Figure 3), in reality it is impossible to cut a 4'×4' piece and a 4'×3' piece from the given 6'×6' sheet as depicted in figure 4.

Suppose that we have a new pattern and it successfully satisfies the heuristic algorithm condition related to total area as explained earlier. To overcome the problem and obtain a more refined solution, we finetuned the algorithm as follows. The first step is to create a solution matrix *C*, initialized to one, where the corresponding nm elements of the matrix refer to the 1'×1' wooden pieces of the n'×m' wooden sheet. The second step is to create a *k*-tuple $(r_1, r_2, \dots, r_k)$ where r_i represents the number of $a_i \times b_i$ pieces that are desired to be cut.

Figure-3. The cutting the pattern (1, 1, 0) satisfying the area constraint

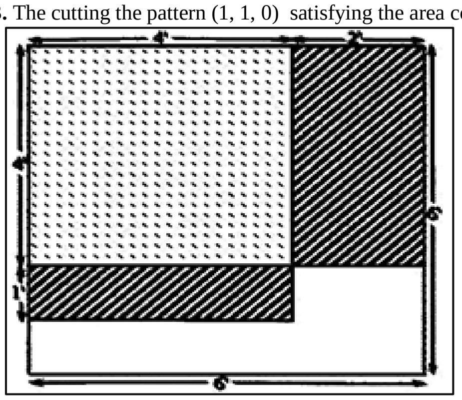
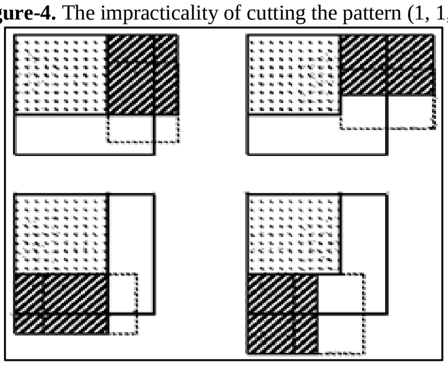


Figure-4. The impracticality of cutting the pattern (1, 1, 0)



Next, select a submatrix with the dimensions $a_1 \times b_1$ from the top left corner (as per cutting assumptions) and check if all of its elements are 1. If all elements of the sub matrix are 1 (this is achieved by comparing the area of the piece being cut, i.e. $a_1 \times b_1$and the summation of the elements of the selected submatrix), it will imply that a piece corresponding to the dimensions of the submatrix can be cut. To reflect that the piece is cut, we update the matrix along with the tuple. The tuple is updated by reducing 1 from r_1 so that the new tuple will read $(r_1 - 1, r_2, \dots, r_k)$. The submatirix is update by assigning 0 to its elements. Next provided that, $r_1 - 1$, the new first entry of the tuple is non-zero, we check if another $a_1 \times b_1$ piece could be cut. Likewise, this can be continued, by selecting submatrices and shifting the submatrix horizontally one cell at a time till it reaches the right most possible position. When the end of the row is reached, we move to the beginning of the next row in lexicographical order till we reach the end of the sheet, so that all possible submatrices are checked. In the case when a submatrix cannot be found, we utilize the same method by considering the submatrix $b_1 \times a_1$ (orientation of the piece is changed) and proceed in the same manner by updating the tuple and the matrix accordingly.

We repeat this process recursively and a successful reduction to a $(0,0,\dots,0)$ tuple will indicate an acceptable pattern $(r_1, r_2, \dots, r_k)$ whereas the alternative scenario will indicate an unacceptable pattern $(r_1, r_2, \dots, r_k)$.

In our Java program we evaluated the feasibility of each pattern separately and calculated the corresponding wastage of the feasible patterns.

Once all the variables were generated we obtain the following output. It is worth noting that in the following output x_i is represented by Xi etc.

10'×4' Sheet

(7'×4'), (7'×3'), (4'×4'), (7'×2'), (4'×3'), (3'×2')
X24 → (0, 0, 0, 2, 0, 2) - 0
X25 → (0, 0, 0, 2, 1, 0) - 0
X26 → (0, 0, 1, 0, 0, 0) - 24
X27 → (0, 0, 1, 0, 0, 1) - 18
X28 → (0, 0, 1, 0, 0, 2) - 12
X29 → (0, 0, 1, 0, 0, 3) - 6
X30 → (0, 0, 1, 0, 0, 4) - 0
X31 → (0, 0, 1, 0, 1, 0) - 12
X32 → (0, 0, 1, 0, 1, 1) - 6
X33 → (0, 0, 2, 0, 0, 0) - 8
X34 → (0, 0, 2, 0, 0, 1) - 2
X35 → (0, 1, 0, 0, 0, 0) - 19
X36 → (0, 1, 0, 0, 0, 1) - 13
X37 → (0, 1, 0, 0, 0, 2) - 7
X38 → (0, 1, 0, 0, 1, 0) - 7
X39 → (1, 0, 0, 0, 0, 0) - 12
X40 → (1, 0, 0, 0, 0, 1) - 6
X41 → (1, 0, 0, 0, 0, 2) - 0
X42 → (1, 0, 0, 0, 1, 0) – 0

6'×6' Sheet

(4'×4'), (4'×3'), (3'×2')
Y1 → (0, 0, 1) - 30
Y2 → (0, 0, 2) - 24
Y3 → (0, 0, 3) - 18
Y4 → (0, 0, 4) - 12
Y5 → (0, 0, 5) - 6
Y6 → (0, 0, 6) - 0
Y7 → (0, 1, 0) - 24
X1 → (0, 0, 0, 0, 0, 1) - 34
X2 → (0, 0, 0, 0, 0, 2) - 28
X3 → (0, 0, 0, 0, 0, 3) - 22
X4 → (0, 0, 0, 0, 0, 4) - 16
X5 → (0, 0, 0, 0, 0, 5) - 10
X6 → (0, 0, 0, 0, 0, 6) - 4
X7 → (0, 0, 0, 0, 1, 0) - 28
X8 → (0, 0, 0, 0, 1, 1) - 22
X9 → (0, 0, 0, 0, 1, 2) - 16
X10 → (0, 0, 0, 0, 1, 3) - 10
X11 → (0, 0, 0, 0, 1, 4) - 4
X12 → (0, 0, 0, 0, 2, 0) - 16
X13 → (0, 0, 0, 0, 2, 1) - 10
X14 → (0, 0, 0, 1, 0, 0) - 26
X15 → (0, 0, 0, 1, 0, 1) - 20

X16 → (0, 0, 0, 1, 0, 2) - 14
X17 → (0, 0, 0, 1, 0, 3) - 8
X18 → (0, 0, 0, 1, 0, 4) - 2
X19 → (0, 0, 0, 1, 1, 0) - 14
X20 → (0, 0, 0, 1, 1, 1) - 8
X21 → (0, 0, 0, 1, 1, 2) - 2
X22 → (0, 0, 0, 2, 0, 0) - 12
X23 → (0, 0, 0, 2, 0, 1) - 6
Y8 → (0, 1, 1) - 18
Y9 → (0, 1, 2) - 12
Y10 → (0, 1, 3) - 6
Y11 → (0, 2, 0) - 12
Y12 → (0, 2, 1) - 6
Y13 → (0, 2, 2) - 0
Y14 → (1, 0, 0) - 20
Y15 → (1, 0, 1) - 14
Y16 → (1, 0, 2) - 8
Y17 → (1, 0, 3) – 2

4'×3' Sheet
(4'×3'), (3'×2')
Z1 → (0, 1) - 6
Z2 → (1, 0) – 0

Wooden bars - 2"×2"×5'
(2"×2"×2'), (2"×2"×1')
P1→ (0, 1) – 4
P2→ (0, 2) - 3
P3→ (0, 3) - 2
P4→ (0, 4) - 1
P5→ (0, 5) - 0
P6→ (1, 0) - 3
P7→ (1, 1) - 2
P8→ (1, 2) - 1
P9→ (1, 3) - 0
P10→ (2, 0) - 1
P11→ (2, 1) - 0

Wooden bars - 2"×4"×5'
(2"×4"×5'), (2"×4"×2')
Q1→ (0, 1) - 3
Q2→ (0, 2) - 1
Q3→ (1, 0) - 0

We formulated the Integer program using the following variables.

$e_{1,i}$ = the error of the pattern that corresponds to x_i
$e_{2,i}$ = the error of the pattern that corresponds to y_i
$e_{3,i}$ = the error of the pattern that corresponds to z_i
$e_{4,i}$ = the error of the pattern that corresponds to p_i
$e_{5,i}$ = the error of the pattern that corresponds to q_i
Therefore, the objective function is to:

$$\text{Minimize } Z = \sum_i e_{1,i}\ x_i + \sum_i e_{2,i}\ y_i + \sum_i e_{3,i}\, z_i + \sum_i e_{4,i}\ p_i + \sum_i e_{5,i}\, q_i$$

The constraints correspond to the following table:

Table-2. Constraints

	Dimensions	Covered box *(40)*	Open box*(30)*	Cupboard *(20)*	Small table*(20)*	Large table*(21)*	Chair *(33)*	Rack *(19)*	Almirahs *(10)*	Large desk*(43)*	Computer table*(25)*	Total pieces
Sheets	7′×4′	4*(40)*	3*(30)*	3*(20)*		1*(21)*			1*(10)*			**341**
	7′×3′				1*(20)*			3*(19)*	3*(10)*	1*(43)*	1*(25)*	**175**
	4′×4′	2*(40)*	2*(30)*	2*(20)*								**180**
	7′×2′			2*(20)*					2*(10)*		1*(25)*	**85**
	4′×3′								2*(10)*			**20**
	3′×2′						2*(33)*			2*(43)*	4*(25)*	**252**
Rods	2″×2″×1′						4*(33)*					**132**
	2″×2″×2′				4*(20)*							**80**
	2″×4″×2′					4*(21)*						**84**
	2″×4″×5′							4*(19)*				**76**

The Integer Program that was formulated is given below:

$$\begin{aligned} Min \quad Z = {} & 34x_1 + 28x_2 + 22x_3 + 16x_4 + 10x_5 + 4x_6 + 28x_7 + 22x_8 + 16x_9 + 10x_{10} + 4x_{11} + 16x_{12} + 10x_{13} \\ & + 26x_{14} + 20x_{15} + 14x_{16} + 8x_{17} + 2x_{18} + 14x_{19} + 8x_{20} + 2x_{21} + 12x_{22} + 6x_{23} + 0x_{24} \\ & + 0x_{25} + 24x_{26} + 18x_{27} + 12x_{28} + 6x_{29} + 0x_{30} + 12x_{31} + 6x_{32} + 8x_{33} + 2x_{34} + 19x_{35} \\ & + 13x_{36} + 7x_{37} + 7x_{38} + 12x_{39} + 6x_{40} + 0x_{41} + 0x_{42} + 30y_1 + 24y_2 + 18y_3 + 12y_4 + 6y_5 \\ & + 0y_6 + 24y_7 + 18y_8 + 12y_9 + 6y_{10} + 12y_{11} + 6y_{12} + 0y_{13} + 20y_{14} + 14y_{15} + 8y_{16} + 2y_{17} \\ & + 6z_1 + 0z_2 + 4p_1 + 3p_2 + 2p_3 + p_4 + 0p_5 + 3p_6 + 2p_7 + p_8 + 0p_9 + p_{10} + 0p_{11} + 3q_1 + q_2 \\ & + 0q_3 \end{aligned}$$

Subject to

$$x_{39} + x_{40} + x_{41} + x_{42} = 341 \quad \textbf{\textit{(Number of 7'×4' wooden pieces)}}$$

$$x_{35} + x_{36} + x_{37} + x_{38} = 175 \quad \textbf{\textit{(Number of 7'×3' wooden pieces)}}$$

$$x_{26} + x_{27} + x_{28} + x_{29} + x_{30} + x_{31} + x_{32} + 2x_{33} + 2x_{34} + y_{14} + y_{15} + y_{16} + y_{17} = 180 \quad \textbf{\textit{(Number of 4'×4' wooden pieces)}}$$

$$x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20} + x_{21} + 2x_{22} + 2x_{23} + 2x_{24} + 2x_{25} = 85 \quad \textbf{\textit{(Number of 7'×2' wooden pieces)}}$$

$$x_7 + x_8 + x_9 + x_{10} + x_{11} + 2x_{12} + 2x_{13} + x_{19} + x_{20} + x_{21} + x_{25} + x_{31} + x_{32} + x_{38} + x_{42} + y_7 + y_8 + y_9 + y_{10} + 2y_{11} + 2y_{12} + 2y_{13} + z_2 = 20 \quad \textbf{\textit{(Number of 4'×3' wooden pieces)}}$$

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + x_8 + 2x_9 + 3x_{10} + 4x_{11} + x_{13} + x_{15} + 2x_{16} + 3x_{17} + 4x_{18} + x_{20} + 2x_{21} + x_{23} + 2x_{24} + x_{27} + 2x_{28} + 3x_{29} + 4x_{30} + x_{32} + x_{34} + x_{36} + 2x_{37} + x_{40} + 2x_{41} + y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 + 6y_6 + y_8 + 2y_9 + 3y_{10} + y_{12} + 2y_{13} + y_{15} + 2y_{16} + 3y_{17} + z_1 = 252 \quad \textbf{\textit{(Number of 3'×2' wooden pieces)}}$$

$$p_6 + p_7 + p_8 + p_9 + 2p_{10} + 2p_{11} = 80 \quad \textbf{\textit{(Number of 2''×2''×2' wooden pieces of rods)}}$$

$$p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + p_7 + 2p_8 + 3p_9 + p_{11} = 132 \quad \textbf{\textit{(Number of 2''×2''×1' wooden pieces of rods)}}$$

$$q_3 = 76 \quad \textbf{\textit{(Number of 2''×4''×5' wooden pieces of rods)}}$$

$$q_1 + 2q_2 = 84 \quad \textbf{\textit{(Number of 2''×4''×2' wooden pieces of rods)}}$$

$$\begin{aligned} & x_i \geq 0 \; for \; all \; i \; and \; x_i's \; are \; integers \\ & y_i \geq 0 \; for \; all \; i \; and \; y_i's \; are \; integers \\ & z_i \geq 0 \; for \; all \; i \; and \; z_i's \; are \; integers \\ & p_i \geq 0 \; for \; all \; i \; and \; p_i's \; are \; integers \\ & q_i \geq 0 \; for \; all \; i \; and \; q_i's \; are \; integers \end{aligned}$$

The optimal solution of the Integer program was generated using a Lingo package. Note that the minimum error for the order that was received was 6960 ft^2.

4. Development of the Website

To obtain the order request of MTW customers and to notify the customers of the final decisions regarding the order details, we developed a dynamic website using an Avata theme.

Acknowledgments

The authors would like to thank Ms. B. L. Samarasekara for her valuable time and cooperation.

References

[1] Gilmore, P. C. and Gomory, R. E., 1961. "A linear programming approach to the cutting-stock problem." *Operations Research,* vol. 9, pp. 849-859.

[2] Lodi, A., Martello, S., and Vigo, D., 2002. "Recent advances on two-dimensional bin packing prob-lems." *Discrete Applied Mathematics,* vol. 123, pp. 379-396.

[3] Lodi, A., Martello, S., and Monaci, M., 2002. "Two-dimensional packing problems. A survey." *Euro-pean Journal of Operational Research,* vol. 141, pp. 241-252.

[4] Art, R. C. J., 1966. "An approach to the two-dimensional irregular cutting stock problem, PhD diss., Massachusetts institute of technology."

[5] Vahrenkamp, R., 1996. "Random search in the one-dimensional cutting stock problem." *European Journal of Operational Research,* vol. 95, pp. 191-200.

[6] Gilmore, P. C. and Gomory, R. E., 1963. "A linear programming approach to the cutting stock problem-Part II." *Operations Research,* vol. 11, pp. 863-888.

[7] Haessler, R. W. and Sweeney, P. E., 1991. "Cutting stock problems and solution procedures." *Eu-ropean Journal of Operational Research,* vol. 54, pp. 141-150.

[8] Cung, V. D., Hifi, M., and Cun, B., 2000. "Constrained two-dimensional cutting stock problems a best-first branch-and-bound algorithm." *International Transactions in Operational Research,* vol. 7, pp. 185-210.

[9] Carvalho, D. J. V., 1999. "Exact solution of bin-packing problems using column generation and branch-and-bound." *Annals of Operations Research,* vol. 86, pp. 629-659.

[10] Gradisar, M. and Trkman, P., 2005. "A combined approach to the solution to the general one-dimensional cutting stock problem." *Computers & Operations Research,* vol. 32, pp. 1793-1807.

[11] Vance, P. H., Barnhart, C., Johnson, E. L., and Nemhauser, G. L., 1994. "Solving binary cutting stock problems by column generation and branch-and-bound." *Computational Optimization and Applications,* vol. 3, pp. 111-130.

[12] Gradišar, M., Resinovič, G., and Kljajić, M., 1999. "A hybrid approach for optimization of one-dimensional cutting." *European Journal of Operational Research,* vol. 119, pp. 719-728.

[13] Jaliff, D. and Dagnino, A., 1995. "An object-oriented tool-kit for building CSP decision support systems". In Systems, Man and Cybernetics, 1995. Intelligent systems for the 21st century IEEE international conference." vol. 4, pp. 3201-3206.

[14] Cintra, G. F., Miyazawa, F. K., Wakabayashi, Y., and Xavier, E. C., 2008. "Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation." *European Journal of Operational Research,* vol. 191, pp. 61-85.

[15] Haims, M. J. and Freeman, H., 1970. "A multistage solution of the template-layout problem." *IEEE Transactions on Systems Science and Cybernetics,* vol. 6, pp. 145-151.

[16] Beasley, J. E., 1985. "An exact two-dimensional non-guillotine cutting tree search procedure." *Operations Research,* vol. 33, pp. 49-64.

[17] Leung, T. W., Yung, C. H., and Troutt, M. D., 2001. "Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem." *Computers & Industrial En-gineering,* vol. 40, pp. 201-214.

[18] Onwubolu, G. C. and Mutingi, M., 2003. "A genetic algorithm approach for the cutting stock prob-lem." *Journal of Intelligent Manufacturing,* vol. 14, pp. 209-218.

[19] Gilmore, P. C. and Gomory, R. E., 1965. "Multistage cutting stock problems of two and more di-mensions." *Operations Research,* vol. 13, pp. 94-120.

[20] León, C., Miranda, G., Rodríguez, C., and Segura, C., 2007. "2D cutting stock problem: a new paral-lel algorithm and bounds." *Euro-Par 2007 Parallel Processing,* pp. 795-804.

[21] Rinnooy, K. A. H. G., Wit, D. J. R., and Wijmenga, R. T., 1987. "Nonorthogonal two-dimensional cutting patterns." *Management Science,* vol. 33, pp. 670-684.

[22] Kröger, B., 1995. "Guillotineable bin packing: A genetic approach." *European Journal of Opera-tional Research,* vol. 84, pp. 645-661.