OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2016 – 9(3): 834-859 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
http://dx.doi.org/10.3926/jiem.1653

Minimizing Waste (Off-cuts) Using Cutting Stock Model: The Case of One Dimensional Cutting Stock Problem in Wood Working Industry

Gbemileke A. Ogunranti[1], Ayodeji E. Oluleye[2]

[1]*Drexel University (United States)*
[2]*University of Ibadan (Nigeria)*

gao32@drexel.edu, ayodeji.oluleye@mail.ui.edu.ng

Received: August 2015
Accepted: August 2016

Abstract:

Purpose: The main objective of this study is to develop a model for solving the one dimensional cutting stock problem in the wood working industry, and develop a computer program for its implementation.

Design/methodology/approach: This study adopts the pattern oriented approach in the formulation of the cutting stock model. A pattern generation algorithm was developed and coded using Visual basic.NET language. The cutting stock model developed is a Linear Programming (LP) Model constrained by numerous feasible patterns. A LP solver was integrated with the pattern generation algorithm program to develop a one - dimensional cutting stock model application named GB Cutting Stock Program.

Findings: Applying the model to a real life optimization problem significantly reduces material waste (off-cuts) and minimizes the total stock used. The result yielded about 30.7% cost savings for company-I when the total stock materials used is compared with the former cutting plan. Also, to evaluate the efficiency of the application, Case I problem was solved using two top commercial 1D-cutting stock software. The results show that the GB program performs better when related results were compared.

Research limitations/implications: This study round up the linear programming solution for the number of pattern to cut.

Practical implications: From Managerial perspective, implementing optimized cutting plans increases productivity by eliminating calculating errors and drastically reducing operator mistakes. Also, financial benefits that can annually amount to millions in cost savings can be achieved through significant material waste reduction.

Originality/value: This paper developed a linear programming one dimensional cutting stock model based on a pattern generation algorithm to minimize waste in the wood working industry. To implement the model, the algorithm was coded using VisualBasic.net and linear programming solver called lpsolvedll (dynamic link library) was integrated to develop a one dimensional cutting stock Program.

Keywords: 1D-cutting stock problem, cutting stock model, pattern generation algorithm, linear programming

1. Introduction

A cutting stock problem (CSP) basically consists of cutting large pieces available in stock to produce smaller pieces (called items) in order to meet a given demand. The cutting is planned to minimize waste of the stock material (other objectives may arise). These objectives may be to minimize trim loss, the number of cutting lines or production cost, maximize profit and so on. In a cutting plan, the required set of pieces from the available stock lengths must be obtained. The objective is to minimize the number of used stock lengths or, equivalently, trim loss (waste) (Murat, Urfat & Ahmet, 2011). This type of problem occurs in several industries such as paper, aluminium, steel, glass, and furniture industry among others. Kallrath, Rebennack, Kallrath and Kusche (2014) solved cutting stock problem in the pulp and paper industry with the objective of minimizing the number of rolls and the patterns, thereby preventing overproduction.

The one-dimensional cutting stock problem (1D-CSP) can be stated as follows: Given a set of items I, each item $i \in \{1, ..., m\}$ of length l_i and demand of b_i pieces, to be cut out of a virtually infinite supply of stock length L (where $0 < l_i \leq L, \forall\ i \in I$), in order to minimize the number of stock objects used (Garraffa, Salassa, Vancroonenburg, Berghe & Wauters, 2014). The problem considered in this paper is to find a cutting plan that would minimize the waste material when a set of orders different in length and quantity is to be cut from a pack of stocks with constant length (Yang, Sung & Weng, 2006).

The purpose of this paper is to develop a mathematical model for solving the one dimensional cutting stock problem in the wood working industry and develop a user friendly computer program for its implementation. Many wood working companies are seeking ways of minimizing production waste (and maximize profit) but faced with lack of technical know-how and competency to so do. Thus, there is need for a very easy to operate and interpret cutting stock program for improving operational efficiency. Based on the problem encountered, a methodology for integrating the cutting patterns generation algorithm with LP solver was developed. In order to achieve the waste minimization objective, the cutting stock model was coded into a computer application using Visual Basic.NET language.

This paper is organized as follows: section 2 reviews the related work on the cutting stock problem. In section 3, the methodology and solution approach are presented. Section 4 provides the problems solved to demonstrate practical application of the model. Section 5 presents results obtained by applying the one dimensional cutting stock model. Finally, the last section concludes the study and optimal cutting using the 1D-cutting stock program was proposed.

2. Literature Review

2.1. Cutting and Packing Problems

Cutting and packing problems are combinatorial optimization problems. As with many other problems of this kind, they are easy to state, and difficult to solve. Matsumoto, Umetani and Nagamochi (2011) also referred to these problems as NP hard. The standard problem is defined as follows: given a set of small and large objects, how the small objects should be obtained from the large ones in order to optimize a given criterion. The typical restrictions to which a cutting or packing plan is submitted are the impossibility for the small objects to overlap, and the limited capacity or length of the large objects.

There are two main approaches to solve this problem: exact and heuristic methods. Exact algorithms are mainly based on linear/dynamic programming and branch-and-bound techniques. Vanderbeck (1999) introduced a branch-and-price algorithm based on column generation approach for solving the cutting stock problem. The author focus on how standard branch- and-bound enhancement features such as early branching, variable fixing, and the use of cutting planes can be incorporated in the branch-and-price algorithm. Then, how to select appropriate branching priorities, and implement a rounding heuristic.

More recently, Kartak, Ripatti, Scheithauer and Kurz (2014) used algorithmic approach based on exhaustive enumeration and integer linear programming to solve one-dimensional cutting stock problem (1CSP) with respect to the integer round-up property (IRUP).

Moreover, heuristic methods have greater flexibility in taking into account problem specific constraints and offer a trade-off between the quality of a solution and its computational effort. They are required to provide good, but not necessarily optimal solutions. Some of the new heuristics methods found in the current literature are: (Dikili, Takinacı & Pek, 2008; Matsumoto et al., 2011; Araujo, Poldi & Smith, 2014; Cherri, Arenales & Yanasse, 2013; Garraffa et al., 2014; Cui, Zhong & Yao, 2015). Araujo et al. (2014) presents a heuristics method based on genetic algorithm to solve one-dimensional cutting stock problem by considering two conflicting objective functions: minimization of both the number of objects and the number of different cutting patterns used.

Previously, Dyckhoff (1990) defined a formal typology for cutting and packing problems by systematically integrating various kinds of problems and notions. This typology was improved by Wäescher, HauBner and Schumann (2007) with the definition of new categorization criteria. Table 1 shows the typology found in the literature.

Dyckhoff's C&P typology		Waescher's improved C&P typology	
Dimensionality			
1	One-dimensional	1	One-dimensional
2	Two-dimensional	2	Two-dimensional
3	Three-dimensional	3	Three-dimensional
N	N-dimensional	N	N-dimensional
Kind of assignment			
B	All large objects and a selection of small objects	OM	Output value maximization
V	A selection of large objects and all small objects	IM	Input value minimization
Assortment of large objects			
O	One large object	O	One Large object
I	Many identical large objects		OA all Fixed dimensions
D	Different large objects		OO one variable dimension
			OM more variable dimensions
		SO	Several Large objects
			SI Identical Large objects
			SW weakly heterogeneous assortment
			SS strongly heterogeneous assortment
Assortment of small objects			
F	Few small objects of different figures	IS	Identical small items
M	Many small objects of many different figures	W	Weakly heterogeneous assortment
R	Many small objects of relatively few different figures	S	Strongly heterogeneous assortment
C	Many identical small objects		

Table 1. Dyckhoff's (1990) and Waescher's (2007) typologies for C&P problems

Dyckhoff classifies the solution of one dimensional cutting stock problem into two: *item oriented* and *pattern-oriented* approaches. Item oriented approach is characterized by individual treatment of every item to be cut. In the pattern oriented approach, at first, order lengths are combined into cutting patterns, for which - in a succeeding step - the cutting frequencies are determined in order to satisfy the demands. The constraints in the pattern-oriented approach are based on the algorithm of Gilmore and Gomory (1961, 1963). However, a pattern-oriented approach is possible only when the stock is of the same length or of several standard lengths. An item-oriented approach is used when all stock lengths are different and frequencies cannot be determined.

2.2. One Dimensional Cutting and Packing Models

In the literature, researchers have used different exact and heuristics solution approaches to support cutting stock decision making in the industry (Kallrath et al., 2014). Most recently, Delorme, Iori and Martello (2016) provides a comprehensive review of the main mathematical models and algorithms developed for exactly solving the one-dimensional bin packing and cutting stock problems. The problems considered in their review are classified as 1-dimensional SBSBPP (Single Bin Size Bin Packing Problem) and 1-dimensional SSSCSP (Single Stock Size Cutting Stock Problem). Also, a survey of cutting stock problems with usable leftovers can be found in Cherri, Arenales, Yanasse, Poldi and Vianna (2014) as these problems does not fit the classification based on the typology proposed by Wäscher et al. (2007). Most of the approaches for solving the standard cutting and packing problems in a single dimension based on linear programming models that have been proposed can be divided into four categories:

a) The Assignment Formulations

b) The Pattern-Oriented Formulations

c) The One-Cut Formulations

d) The Flow Models

2.2.1. The Assignment Formulations

The first integer linear programming formulation was proposed by Kantorovich (1960), based on assignment of variables as follows.

$$Minimize \sum_{j=1}^{n} y_j \tag{1}$$

Subject to:

$$\sum\nolimits_{j=1}^{n} X_{ij} \geq b_i \quad i = 1, \ldots, m \tag{2}$$

$$\sum\nolimits_{i=1}^{m} w_i X_{ij} \leq W_i y_j \quad , j = 1, \ldots, n \tag{3}$$

$$y_j \in \{0, 1\}, \qquad j = 1, \ldots, n \tag{4}$$

$$x_{ij} \geq 0 \text{ and integer}, \quad i = 1, \ldots, m; \quad j = 1, \ldots, n \tag{5}$$

Where: the above model has binary and general integer variables; with y_j been the choice of roll j, X_{ij} is the number of items of size w_i assigned to roll j and W_i is the capacity of the bin i. Equation 3 is the knapsack constraints, one for each roll, while equation 2 is the demand (b_i) constraints working as the linking constraints. Where n is a known upper bound on the number of rolls needed, y_j = 1, if roll k is used, and 0 otherwise, and variable X_{ij} is the number of times item i is cut in roll j. A lower bound for the optimum can be obtained from the optimum of its LP relaxation, which results from substituting the two last constraints for $0 \leq y_j \leq 1$ and $X_{ij} \geq 0$.

Martello and Toth (1990) showed that the lower bound provided by the LP relaxation can be very weak. This model's main drawback is that it produces poor lower bound with large waste. Good quality lower bounds are of vital importance when using LP based approaches to solve integer problems (Carvalho, 2002).

2.2.2. The Pattern-Oriented Formulations

The pattern oriented model of Gilmore and Gomory (1961) followed the assignment formulation of Kantorovich. Gilmore and Gomory (1961) formulated the cutting stock problem as an integer programming problem. The cutting stock problem can be modelled as:

$$Minimize \sum_{j=1}^{n} X_j \tag{6}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} X_j \geq b_i \, , i = 1, \ldots, m \tag{7}$$

$$\sum_{i=1}^{m} a_{ij}\, l_i \leq L \tag{8}$$

$$a_{ij} \geq 0 \text{ and integer} \tag{9}$$

$$X_j \geq 0 \text{ and integer}, \quad j = 1, \ldots, n \tag{10}$$

Where,

L is the length of the objects in stock;

l_i is the length of the items, $i = 1, 2, \ldots, m$;

b_i represents the demand of item i, $i = 1, 2, \ldots, m$;

$A_j = (a_{1j}, \ldots, a_{ij}, \ldots, a_{mj})^T$ be a cutting pattern, $i = 1, 2, \ldots, n$, where a_{ij} is the number of items i in the pattern;

X_j be the frequency that the pattern j is cut.

The difficulty in using this model is the large number of patterns to be enumerated in practical problems. Since, the possible patterns numbers increases exponentially as the number of different items and the demands increases.

2.2.3. The One-Cut Formulations

In the one-cut models, independently proposed by Rao (1976) and Dyckhoff (1981), the principle is to determine how to apply a single cut (a one-cut) on an original or residual piece of material. A one-cut divides the raw material in two pieces: an ordered item and a residual object as shown in the figure below.

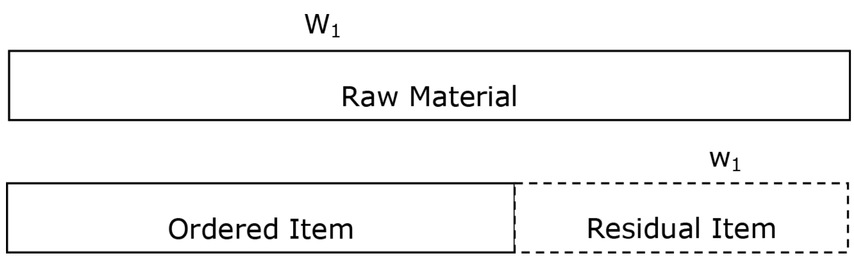


Figure 1. One-Cut

The latter part can be trim loss, a portion to be cut further, or another ordered width. The model proposed by Dyckhoff considers the case in which different stock lengths are available. In the sequel, D represents the set of item widths to cut from the stock rolls whose lengths belong to $S = \{W_1, ..., W_K\}$. The set of residual objects whose length is sufficiently large to cut an item is denoted by R. The decision variables $y_{p,q}$ indicate the number of times a piece with length p is to be cut so as to produce an item of width q, and a residual object of width $p-q$. The z_k variables indicate the number of stock rolls with lengths W_k that are used. The Dyckhoff's model is as follows:

$$Minimize \sum_{k=1}^{K} W_k Z_k \tag{11}$$

Subject to:

$$Z_k + \sum_{p \in D: p+q \in SUR} y_{p+q,p} \geq \sum_{p \in D: p<q} y_{q,p} \quad \forall q \in S = \{W_{1,} \dots, W_k\} \tag{12}$$

$$\sum_{p \in SUR: p>q} y_{p,q} + \sum_{p \in D: p+q \in SUR} y_{p+q,p} \geq \sum_{p \in D: p<q} y_{q,p} + N_q \forall q \in (DUR) \backslash S \tag{13}$$

$$z_k \leq B_k \quad k = 1, 2, ..., K \tag{14}$$

$$y_{p,q} \geq 0 \text{ and integer}, \qquad p \in S \cup R, q \in D, q < p \tag{15}$$

$$z_k \geq 0 \text{ and integer}, k = 1, 2, ..., K \tag{16}$$

Where,

S is the set of the (sizes of the) original objects, $p \in\in \{W_1, W_2, \dots, W_K\} \subset N$

R be the set of the (sizes of the) residual pieces obtained after cutting an item from an object or a residual piece itself

D be the set of the (sizes of the) items $q \in \{w_1, w_2, \dots, w_m\} \subset N$. We assume without loss of generality that $S \cap D = \Phi$

$y_{p,q}$ be the number of pieces of size p that are divided into a piece of size q, and a residual piece of size $p - q$

z_k denotes the number of objects of size W_k used

N_q stands for the demand of items of size q

B_p is the number of objects of size p in stock, $p = 1, 2, \dots, K$.

The major disadvantage of this model is the increasing number of variables as the number of items increases. The possible sizes of residual pieces can be large and they need to be enumerated to solve the problem. This model also cannot be extended, at least in a straightforward manner to bi-dimensional or multidimensional cutting stock problems.

2.2.4. The Flow Models

The arc flow formulations are based on flow variables. Let $G = (V, A)$ be an acyclic directed graph with vertices $V = \{0, 1, 2,\ldots, W\}$ where W is the size of the object in stock and $A = \{(i, j): 0 \leq i < j \leq W\}$ is the set of arcs. There exists a directed arc (i, j) in this graph G if there is an item d of size w_d and $j - i = w_d$.

Carvalho (1999) modelled the one-dimensional cutting stock problem as an arc flow model. In this model, a unit of flow from node 0 to node W corresponds to a cutting pattern since it defines a path from node 0 to node W where the addition of the sizes of the items in correspondence to the arcs in this path is smaller than the size of the object. The mathematical formulation for the arc flow model is given below:

$$\text{Min } Z \tag{17}$$

Subject to:

$$\sum_{(i,j)\in A} \mathrm{X}_{ij} - \sum_{(j,k)\in A} X_{jk} = -Z \qquad if\ j = 0 \tag{18}$$

$$\sum_{(i,j)\in A} \mathrm{X}_{ij} - \sum_{(j,k)\in A} X_{jk} = 0 \qquad if\ j = 1,2,\ldots,w-1 \tag{19}$$

$$\sum_{(i,j)\in A} \mathrm{X}_{ij} - \sum_{(j,k)\in A} X_{jk} = Z \qquad if\ j = w \tag{20}$$

$$\sum_{(k,k+W_d)\in A} \mathrm{X}_{k,k+W_d} \geq b_d \quad d = 1,2,\ldots,m \tag{21}$$

$$X_{ij} \geq 0 \text{ and integer } \forall\ (i,j) \in A \tag{22}$$

Where,

Z is the flow in a feedback arc, from vertex W to vertex 0

X_{ij} is the flow in arc (i, j)

b_d is the demand of item d, $d = 1, 2, \ldots, m$.

3. Development of Cutting Stock Program

One of the major objectives during the software development is the user friendliness of the application. Visual Basic.NET language was used for the development because of its capability to design a good graphic user interface. The one dimensional cutting stock program was develop by integration of pattern generation and LP model solver. The pattern generation algorithm was coded using Visual Basic.NET language to produce all the feasible cutting patterns. The cutting stock program utilizes the linear programming to select the optimal cutting patterns from the list of numerous patterns generated (which served as the constraint of the LP cutting stock model).

A pattern is a plan of how to cut a given set of items from a stock material. An algorithm for generating feasible cutting patterns was developed using logical operation sequence. The cutting patterns vector $P_j = \{a_{1j}, \ldots, a_{2j}, \ldots, a_{nj}\}^T$ is a feasible pattern, if it satisfies the Condition stated below:

$$\sum_{i=1}^{n} a_{ij}\ l_i \leq L \tag{23}$$

3.1. Pattern Generation Algorithm

An outline of algorithm for generating feasible cutting patterns is presented in this sub section. This algorithm can be stated as follows:

(Note that l_i has been reordered in descending order)

Step 1: Set $i = 1$

Step 2: Cut the maximum number of l_i possible to generate the first pattern using:

$a_{ij} = L/l_i$, other $a_{ij} = 0$

Pattern 1: $P_j = \{a_{ij}, 0, \ldots, 0\}^T$; Waste 1: $W_j = L - a_{ij}l_i$

Step 3: Compare W_j with other l_n; such that: $a_{11} = a_{12}$

If $W_j < l_n$; Discard new pattern

Else,

$a_{i2} = W_j/l_i$, other $a_{ij} = 0$

Next pattern: $P_j = \{a_{12}, a_{i2}, \ldots, a_{nj}\}^T$; Waste j: $W_j = L - \sum_{i=1}^{n} a_{ij}l_i$

Step 4: To generate next pattern decrease a_{1j} by 1 and estimate the remaining length (L_r)

$a_{1j} = a_{1,j-1}-1$, then $L_r = L - a_{1j}l_1$ Therefore, $a_{2j} = L_r/l_2$

Calculate new remaining length (R)

$R = L_r - a_{2j}l_2$

Compare R with remaining l_i for $i = 3, \ldots, n$

If $R < l_i$, $a_{ij} = 0$, otherwise $a_{ij} = R/l_i$

Next pattern: $P_j = \{a_{1j}, a_{2j}, \ldots, a_{nj}\}^T$; Waste j: $W_j = L - \sum_{i=1}^{n} a_{ij}l_i$

Step 5: From the last pattern generated, retain a_{ij} decreased by 1, then decrease the next a_{ij} by 1 and find L_r.

$a_{1,j+1} = a_{1j}$

$a_{2,j+1} = a_{2j} - 1$, then $L_r = L - \sum_{i=1}^{2} a_{ij}l_i$ Therefore, $a_{3,j+1} = L_r/l_3$

Calculate remaining length (R)

$R = L_r - a_{3,j+1}l_3$

Compare R with remaining l_i for i = 3,…, n

If $R < l_i$, $a_{ij} = 0$, else $a_{ij} = R/l_i$

Next pattern: $P_j = \{a_{1,j+1}, a_{2,j+1}, \ldots, a_{n,j+1}\}^T$; Waste j +1: $W_{j+1} = L - \sum_{i=1}^{n} a_{ij}l_i$

Repeat until $a_{ij} = 1$

Step 6: Repeat steps 4 and 5 until you compare with R with only the last l_i.

Step 7: Repeat step 1 – 6 for items l_i from $i = 2, \ldots, n-1$ and entry elements before a_{ij} in the new patterns generated is equal to zero.

Step 8: Set $i = n$

Step 9: Cut the maximum number of l_n possible from the stock object using:

$a_{nj} = L/l_n$, other $a_{ij} = 0$

Pattern m: $P_m = \{0, 0, \ldots, a_{ij}\}^T$; Waste m: $W_m = L - a_{ij}l_n$

Go to step 10

Step 10: Arrange patterns such that:

$$\left\{ \begin{array}{l} \quad\quad P_1 \;\; P_2 \;\; P_3 \;\; \dots \;\; P_m \\ l_i \\ : \\ l_n \\ W_j \end{array} \right\}$$

The patterns generated eliminate dominant patterns, leaving non dominated patterns in the feasible solution space.

3.2. Proposed Cutting Stock Model

3.2.1. Notations

The following notations were used in the cutting stock model:

L is the length of the objects in stock;

l_i is the length of the items, i = 1, 2, …, n;

b_i represents the demand of item i, $i = 1, 2, \dots, n$;

$P_j = (a_{1j}, \dots, a_{ij}, \dots, a_{nj})^T$ be a cutting pattern,

$j = 1, 2, \dots, m$, where a_{ij} is the number of times items i appears in the j^{th} cutting pattern;

X_j be the number of times that the pattern j is cut.

W_j is the waste material (off-cuts) of pattern j

3.2.2. Linear Programming Model

The mathematical model for waste (off-cuts) minimization can be stated as follows:

$$Minimize \sum_{j=1}^{m} W_j X_j \tag{24}$$

Subject to:

$$\sum_{i=1}^{n} a_{ij} X_j = b_i, \qquad i = 1, \dots n \tag{25}$$

$$a_{ij} \geq 0 \text{ and integer} \tag{26}$$

$$X_j \geq 0 \text{ and integer}, \quad j = 1, \dots, m \tag{27}$$

The total number of stock objects used can be estimated using equation 28 expressed as:

$$\text{Total Stock Object Used} = \sum_{j=1}^{m} X_j \tag{28}$$

In the one dimensional cutting stock problem, the waste (off-cuts) associated can be defined as:

$$W_j = L - \sum_{i=1}^{n} a_{ij}\, l_i\,, \quad j = 1, \dots, m \tag{29}$$

3.3. Software Development

The Visual Basic.NET programming language was used to develop a versatile user friendly one dimensional cutting model application called GB cutting stock program. This software works perfectly on Microsoft windows operating systems and can also work on other windows operating systems. Visual Basic.NET enables us to build application that can be compiled into executable files which make its independent of Visual Basic.NET environment. This helps protect the source code against unwanted modification.

3.3.1. Software Design and Tools

The Windows Forms Application (WFA) on the Visual Basic.NET was used in designing of the GB cutting stock program. The software development makes use of the tools provided by the integrated development environment (IDE). The Visual Studio IDE is incredibly powerful and provides hundreds of tools for building and modifying projects. Menus and toolbars contain hundreds if not thousands of commands for creating, loading, saving, and editing different kinds of projects and files.

The program for generating patterns was written in Visual Basic.NET language. Having generated patterns using VB code, there is a need for optimization engine for selecting optimal patterns that minimizes waste. Thus, we integrated an Lp_solve to our program. Lp_solve is a free solver based on revised simplex algorithm for solving linear programming (LP) problems and the branch and bound technique for solving integer programming (IP) and mixed integer programming (MIP) problems. Even though it is developed in ANSI C, it can be called from programs written in other programming languages such as C++, Java, C# and visual basic as well.

Our Visual Basic.NET program reads from API (Application Programming Interface), generates patterns efficiently through an array based representation of patterns, calls lp_solve to build and solve the LP model and displays the solution on two different tabs (text solution and Visual solution). It has good Graphic User Interface for displaying the visual solution because Visual Basic.Net is provided with GDI (Graphics Design Interface) functions that enable creation of two-dimensional vector graphics, imaging, and typography within the Windows operating system.

3.3.2. Software Interface

The GB cutting stock program consists of three main areas. The Input bar, parts details section and solution tabs section as shown in Figure 2. The input bar has the solve button on it.

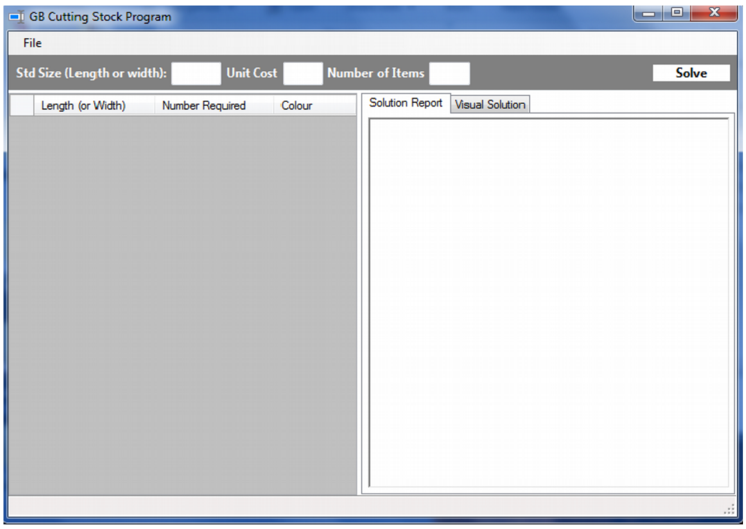


Figure 2. GB Cutting stock program Interface

3.3.3. Flow Chart

A simplified version of the flow chart for the cutting stock model software is presented in Figure 3 and the step by step explanation of the flow chart is presented in the next section.

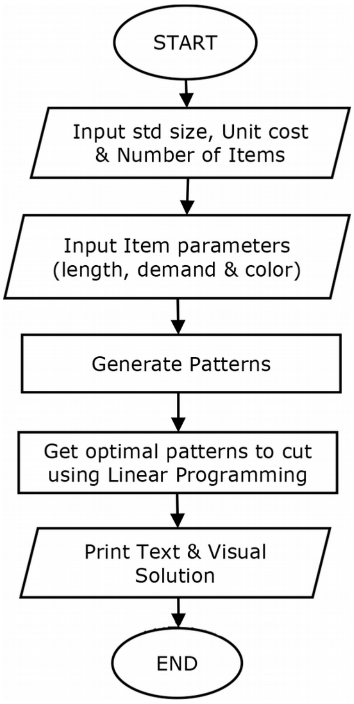


Figure 3. Flow Chart for the GB Cutting Stock Program

3.4. Solution Procedure

The one dimensional cutting stock program developed solves at 0.01 secs CPU time when it was installed and tested on Acer 5532, 3G RAM, 1.6GHz AMD Athlon 64 Processor with Windows 7 Operating System (OS). It can also work well on other OS such as Windows XP, Vista and so on with a minimum RAM of 64 MB.

The solution procedure to follows after starting the program is as follows:

1. Input the Stock length (L), Cost per unit (C), numbers of items needed (n) and press enter.
2. Input the values of the length of each items (l_i) and corresponding demands (b_i).
3. Generate all feasible cutting patterns such that a vector $P_j = \{a_{1j}, a_{2j}, \ldots, a_{nj}\}^T$ represent a feasible pattern, which must satisfy the Condition stated in Equation 23.

4. The optimal cutting pattern is now selected by solving the LP model (with our application) which minimizes the total waste.

5. Clicking on solve button on the GB cutting stock program, the results displays on two different tabs (Text solution and Visual solution respectively).

6. The total stock materials used (Summation of all X_j) is given as:

$$T_s = \sum_{j=1}^{m} X_j \tag{30}$$

Total cost of stock sheet used is estimated using equation 31 stated below.

$$T_c = C * \sum_{j=1}^{m} X_j \quad = \quad C * T_s \tag{31}$$

7. From the text solution of the LP model, the total waste (off-cuts) can be evaluated from the value of the objective function of the model

8. A visual solution is presented showing items and off-cuts (waste) on each optimal cutting pattern with different colour on the second result tab.

4. Model Application

To demonstrate the applicability of the model, two practical cases are presented in this paper. Case I relates to the cutting problem encountered at a Furniture and Joinery Company while Case II the cutting problem at a general wood works company in Nigeria. Both companies are privately owned.

4.1. Case I

Twelve (12) flush doors are to be produced from 55mm width wood of standard length 12ft (3660mm). Five 55mm width wood can be cut from a standard plank, since 10mm width is cut off to straighten the standard plank edges and the blade width takes about 4mm away at each cut.

Part Number	Part Length(mm)	Number Required for a Door	Total Number Required
1	2000	2	24
2	710	4	48
3	540	2	24
4	620	2	24
5	625	5	60

Table 2. Requirements of the Order

From the current practice in the company I, thirteen (13) planks were used to fulfil this task. Since, a plank ordinarily yields five 55mm width wood, therefore a total of sixty-five (65) 55mm width wood is required. The cost per unit of plank is one thousand two hundred naira (#1200). Five 55mm width wood can be obtained from a plank; this implies that a unit of 55mm width wood technically costs two hundred and forty naira (#240).

4.2. Case II

Two products were selected out of the data collected from Company II. The products are as follows:

4.2.1. Product I: Conference Chairs

Tables 3 and 4 summarise the data required to produce 30 pieces of conference chairs as required.

Width	Number Used	Planks Used
9cm	32	11
7cm	10	3
5cm	12	2
Total Planks Used		16

Table 3. Number of stock length used by current practice to produce the chairs

Parts	Component	Length (cm)	Number per Chair	Number Used
9cm width				
1	Arms	64	2	60
2	Front legs	58	2	60
3	Crosses (Front & Back)	54	2	60
7cm Width				
1	Sides	58	2	60
5cm Width				
1	Back legs	68	2	60

Table 4. Details of Parts list of the conference chair

4.2.2. Product II: Panel Doors

The standard size of a panel door is 206cm X 84cm. Current practice revealed that ten (10) panel doors are produced from thirty (30) planks based on the present cutting technique. A panel door consists of three different types of the stock material from which each required length is obtained. These are of widths 28cm, 15cm and 10cm. The details of each required component is given on Table 5 below.

Parts	Component	Length (cm)	Number per Door	Number Used
28cm width				
1	Small panel	32	2	20
2	Large panel	64	4	40
15cm Width				
1	Upright	206	2	20
2	Crosses	76	3	30
10cm Width				
1	Crosses	76	3	30
2	V. small cross	33	1	10

Table 5. Details parts list of the Panel Doors

5. Results and Discussion

The Software presents results in two forms: The text solution and the visual solution. The text solution gives the description of patterns to be cut, total number of stock to be used and its cost; while visual solution shows the diagram of how the optimal pattern is to be cut.

From Table 6, the model recommends the total stock (55mm width wood) to be used is 42. Since, five pieces 55mm width wood are obtained from a plank; forty-two (42) 55mm width wood will give 8.4planks (approximately 9 standard size planks should be used). Nine planks yield forty-five 55mm width wood, thereby allowing two 55mm width wood to be kept as reserve in case of mistakes while cutting. Figure 4 shows the optimal cutting patterns to be used. The results obtained show that 9 planks should be used instead of the current 13, which result in a saving of 30.7%.

5.1. Text Solution

The text solution obtained from the model can be summarized below:

Pattern No. (j)	LP Solution (X_j)	Number Cut	Lengths (mm)					Total Waste(W_jX_j) mm
			2000	710	625	620	540	
2	21.6	22	22	44	-	-	-	5280
8	2.4	2	2	-	-	-	8	120
9	0.96	1	-	5	-	-	-	110
35	15	15	-	-	60	15	15	0
50	1.8	2	-	-	-	10	2	40
Total	41.76	42	24	49	60	25	25	5550

Table 6. Solution to Case I problem (Flush Door)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Length (cm)			Total Waste (W_jX_j) cm
			64	58	54	
3	15	15	60	-	30	30
21	15	15	-	60	30	390
Total	30	30	60	60	60	420

Table 7. Number of 9cm width to be used to produce Product I (Case II)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Width (cm)			Total Waste (W_jX_j) cm
			9	7	5	
1	3.17	4	12	-	-	12
2	10	10	20	10	10	-
7	0.5	1	1	-	4	1
Total	13.67	15	33	10	14	13

Table 8. Total number of planks to be used for Case II Product I (Conference Chairs)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Length (cm)		Total Waste (W_jX_j) cm
			64	32	
2	8	8	40	8	112
7	1.09	2	-	22	28
Total	9.09	10	40	30	420

Table 9. Number of 28cm width to be used to produce Product II (Case II)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Length (cm)		Total Waste (W_jX_j) cm
			206	76	
1	5	5	5	-	800
2	15	15	15	30	120
Total	20	20	20	30	920

Table 10. Number of 15cm width to be used to produce Product II (Case II)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Length (cm)		Total Waste (W_jX_j) cm
			76	33	
2	6.92	7	28	7	203
3	0.77	1	3	4	6
Total	7.69	8	31	11	209

Table 11. Number of 10cm width to be used to produce Product II (Case II)

Pattern No. (j)	LP Solution (X_j)	Number Cut	Length (cm)			Total Waste (W_jX_j) cm
			28	15	10	
1	10	10	10	-	-	20
2	10	10	-	20	-	0
4	2.67	3	-	-	9	0
Total	22.67	23	10	20	9	420

Table 12. Solution to Case II Product II (Panel Door)

5.2. Visual Solution

Figure 4 present an illustration of the visual solution tab of the application. This shows the number of each item produced per pattern and the off – cuts (waste). Similarly, GB Cutting Stock Program provides visual solution for all other cutting plan solutions.

In addition, implementing the cutting patterns result shown in Tables 7 and 8 respectively for the conference chairs reveals that fifteen (15) planks must be consumed as against the 16 planks used. This gives a saving of 6.25% to the organisation. In a large scale operation, the savings could be substantial in monetary terms.

Similarly, implementing the cutting patterns result shown Tables 9-12 respectively to produce the panel doors shows that twenty-three (23) planks are required instead of thirty (30) planks used. This leads to a net saving of 23.3% when compared to the number of planks used by the former cutting plan.

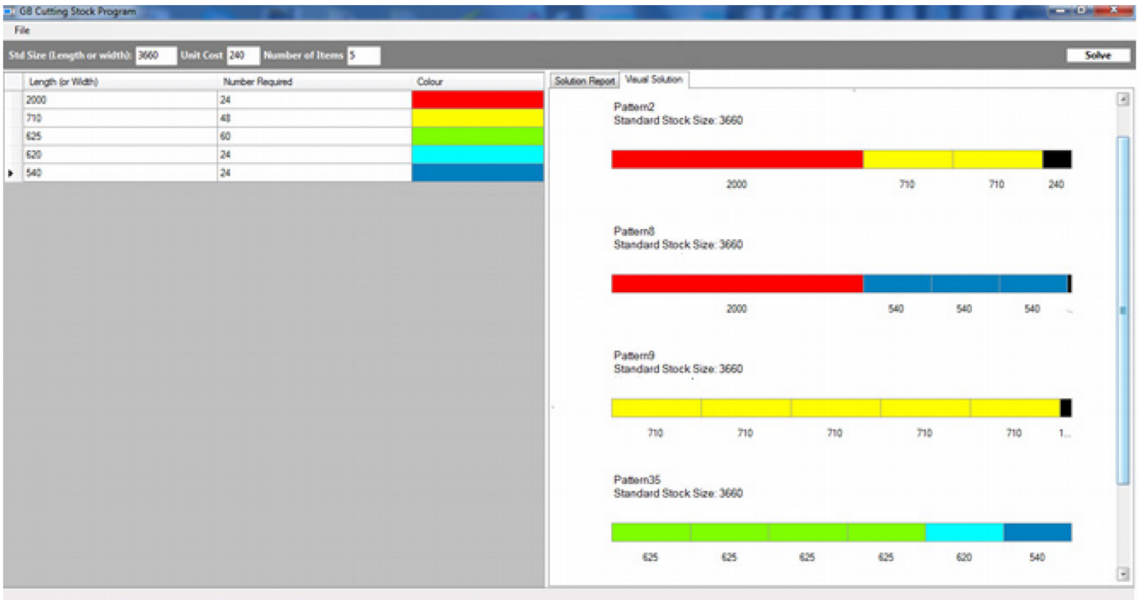


Figure 4. Optimal Cutting Plans for the flush door problem.

5.3. Comparison Study

To evaluate the efficiency of GB cutting stock program, two commercial softwares that have executable trial version available on the internet was used to solve the problem presented in case I. The summary results obtained are as shown on Table 13.

Software	Materials Used	Pattern Used	Total Waste (mm)	Materials Total Cost (#)
GB Cutting Program	42	5	5,550	10,080.00
A	42	6	6,300	10,080.00
B	44	6	13,620	10,560.00

Table 13. Number of materials, pattern used, total waste and cost by each program

The website of the commercial softwares are listed on appendix I as A and B. Commercial software package A produce the same optimal result as our cutting stock program in terms of material used and utilized 6 patterns as shown in Figure 5 to implement the same result. While commercial software package B used the same number of patterns with software A but more materials than both GB cutting program and A. Therefore, GB cutting stock is better because it uses less number of patterns and reduces operation time than A and B. See appendix II for optimal cutting patterns solution using software B.

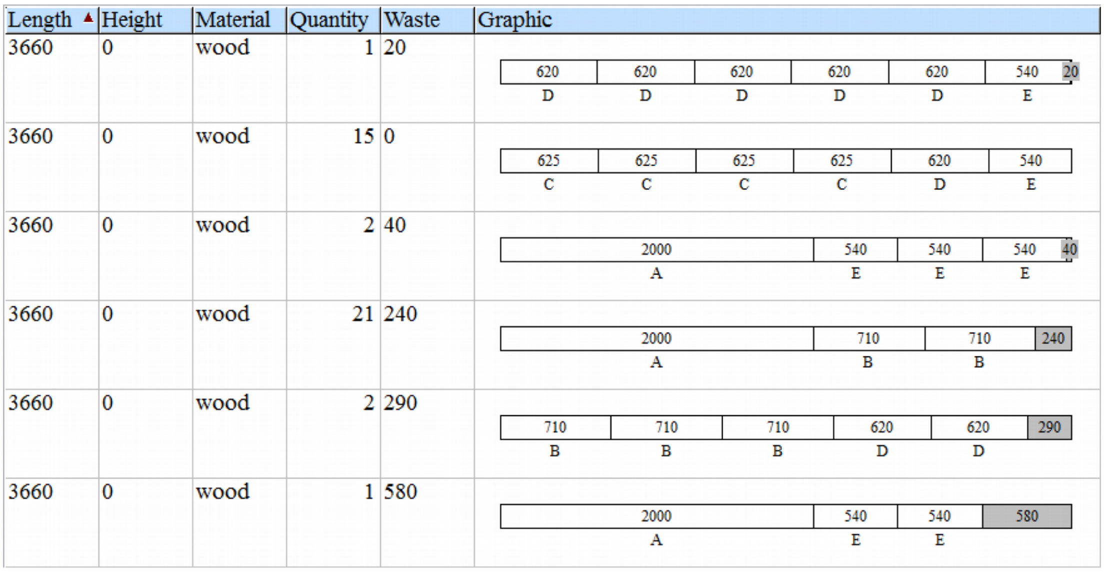


Figure 5. Optimal Cutting Plans generated using package A

Also, conducting price survey on commercial software packages reveals that it would cost within \$100 - \$400 (20,000 - 78,000 local currency) to purchase a license. This is costly to local woodwork industry in Nigeria which cannot transact in international currency. Thus, GB cutting stock provides easy to use program with no prior computer knowledge to local woodworking industry.

6. Conclusion

The cutting stock model and a pattern generation algorithm were developed. The algorithm was coded using VisualBasic.net and a free linear programming solver called lpsolvedll (dynamic link library) was integrated to develop a one dimensional cutting stock program named GB cutting stock program. The cutting stock model significantly reduces material waste, eliminates calculating errors, drastically reduces operator mistakes and minimizes the total stock used, thereby improves productivity. Also, the results obtained were compared against the best solution generated by two commercial softwares for the same problem. It can be concluded that GB cutting stock program yielded a very good result at reasonable computational time and performs better than the commercial software in terms of operational efficiency that is it requires less time to implement results. Finally, due to its affordability; implementing our result yielded a significant cost savings of about 30.7% for company-I when the total stock materials used is compared with the former cutting plan.

Acknowledgements

We would like to thank the British Council for supporting this work under the AKTP (African Knowledge Transfer Partnership) initiative with a Grant. We also thank the anonymous referees for their helpful comments and all those who made suggestions as to making the application simple to use in practice.

References

Araujo, S.A., Poldi, K.C., & Smith, J. (2014). A Genetic Algorithm for the One Dimensional Cutting Stock Problem with Setups. *Pesquisa Operacional,* 34(2), 165-187. http://dx.doi.org/10.1590/0101-7438.2014.034.02.0165

Carvalho, J.M.V. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research,* 86, 629-659. http://dx.doi.org/10.1023/A:1018952112615

Carvalho, J.M.V. (2002). LP Models for Bin Packing Problem and Cutting Stock Problems. *European Journal of Operational Research,* 141, 253-273. http://dx.doi.org/10.1016/S0377-2217(02)00124-8

Cherri, A.C., Arenales, M.N., & Yanasse, H.H. (2013). The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operations Research,* 20, 189-199. http://dx.doi.org/10.1111/j.1475-3995.2012.00868.x

Cherri, A.C., Arenales, M.N., Yanasse, H.H., Poldi, K.C., & Vianna, A.C.G. (2014). The one-dimensional cutting stock problem with usable leftovers – A survey. *European Journal of Operational Research,* 236(2), 395-402. http://dx.doi.org/10.1016/j.ejor.2013.11.026

Cui, Y., Zhong, C., & Yao, Y. (2015). Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost. *European Journal of Operational Research,* 243, 540-546. http://dx.doi.org/10.1016/j.ejor.2014.12.015

Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research,* 255, 1-20. http://dx.doi.org/10.1016/j.ejor.2016.04.030

Dikili, A.C., Takinacı, A.C., & Pek, N.A., (2008). A new heuristic approach to one-dimensional stock-cutting problems with multiple stock lengths in ship production. *Ocean Engineering,* 35(7), 637-645. http://dx.doi.org/10.1016/j.oceaneng.2008.01.003

Dyckhoff, H. (1981). A New Linear Programming Approach to the Cutting Stock Problem. *Operations Research,* 29(6), 1092-1104. Published by INFORMS Stable URL: http://www.jstor.org/stable/170363 http://dx.doi.org/10.1287/opre.29.6.1092

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research,* 44, 145-159. http://dx.doi.org/10.1016/0377-2217(90)90350-K

Garraffa, M., Salassa, F., Vancroonenburg, W., Berghe, G.V., & Wauters, T. (2014). The one-dimensional cutting stock problem with sequence-dependent cut losses. *International Transactions in Operations Research,* 00, 1-20.

Gilmore, P.C., & Gomory, R.E. (1961). A linear programming approach to the cutting stock problem. *Operations Research,* 9, 849-859. http://dx.doi.org/10.1287/opre.9.6.849

Gilmore, P.C., & Gomory, R.E. (1963). A linear programming approach to the cutting stock problem: Part II. *Operations Research,* 11, 863-888. http://dx.doi.org/10.1287/opre.11.6.863

Kallrath, J., Rebennack, S., Kallrath, J., & Kusche, R. (2014). Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. *European Journal of Operational Research,* 238, 374-389. http://dx.doi.org/10.1016/j.ejor.2014.03.027

Kantorovich, L.V. (1960). Mathematical models of organizing and planning production. *Management Science,* 6, 366-422. Translated from the Russian original, 1939. http://dx.doi.org/10.1287/mnsc.6.4.366

Kartak, V.M., Ripatti, A.V., Scheithauer, G., & Kurz, S. (2014). Minimal proper non-IRUP instances of the one-dimensional cutting stock problem. *Discrete Applied Mathematics,* 187. http://dx.doi.org/10.2139/ssrn.2440742

Martello, S., & Toth, P. (1990). Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics,* 28, 59-70. http://dx.doi.org/10.1016/0166-218X(90)90094-S

Matsumoto, K., Umetani, S., & Nagamochi, H. (2011). On the one-dimensional stock cutting problem in the paper tube industry. *Journal of Scheduling,* 14(3), 281-290. http://dx.doi.org/10.1007/s10951-010-0164-2

Murat, E.B., Urfat, N., & Ahmet, Y. (2011). A software for the one-dimensional cutting stock problem. *Journal of King Saud University (Science),* 23(1), 69-76. http://dx.doi.org/10.1016/j.jksus.2010.06.009

Rao, M.R. (1976). On the cutting stock problem. *Journal of the Computer Society of India,* 7, 35-39.

Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming,* 86, 565-594. http://dx.doi.org/10.1007/s101070050105

Waescher, G., HauBner, H., & Schumann, H. (2007). An Improved Typology of Cutting and Packing Problems. *European Journal of Operational Research,* 183, 1109-1130. http://dx.doi.org/10.1016/j.ejor.2005.12.047

Yang, C., Sung, T., & Weng, W. (2006). An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems. *Advances in Engineering Software Advances in Engineering Software,* 37(8), 502-513. http://dx.doi.org/10.1016/j.advengsoft.2006.01.005

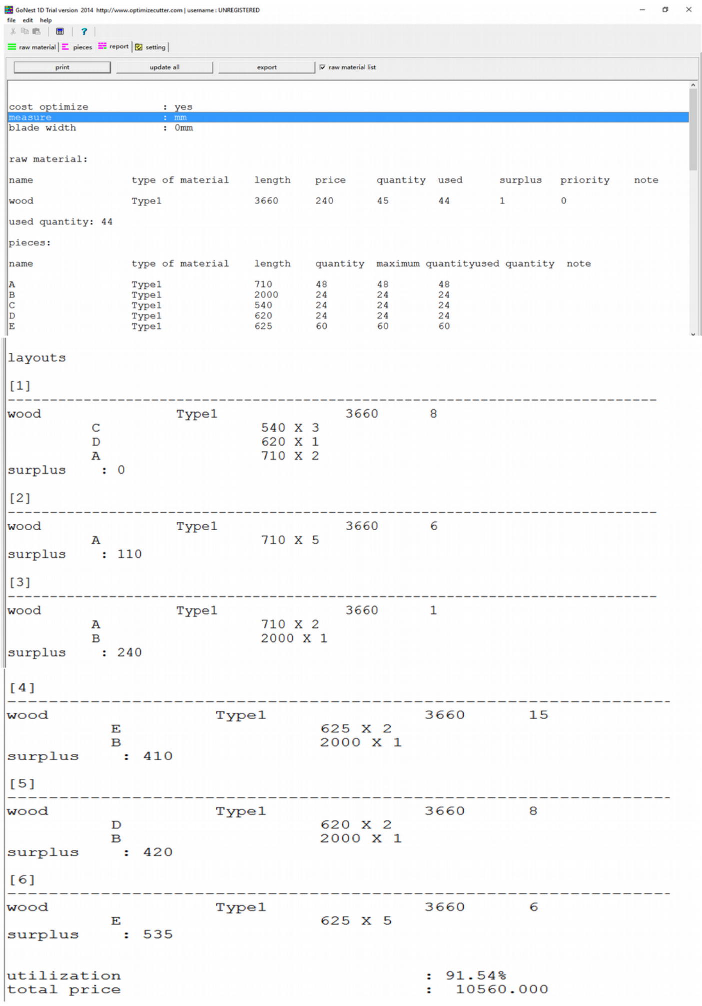
Appendix I

[A] http://www.optimalprograms.com/realcut1d.htm

[B] http://www.optimizecutter.com/product_pipe_cutting_software.html

Appendix II

Optimal cutting patterns using software B



```
GoNest 1D Trial version 2014 http://www.optimizecutter.com | username : UNREGISTERED
file  edit  help
raw material | pieces | report | setting
print    update all    export    raw material list

cost optimize            : yes
measure                  : mm
blade width              : 0mm

raw material:

name          type of material   length   price   quantity  used   surplus  priority  note

wood          Type1              3660     240     45        44     1        0

used quantity: 44

pieces:

name          type of material   length   quantity  maximum quantityused quantity  note

A             Type1              710      48        48      48
B             Type1              2000     24        24      24
C             Type1              540      24        24      24
D             Type1              620      24        24      24
E             Type1              625      60        60      60

layouts

[1]
--------------------------------------------------------------------------------
wood               Type1                 3660       8
           C                  540 X 3
           D                  620 X 1
           A                  710 X 2
surplus    : 0

[2]
--------------------------------------------------------------------------------
wood               Type1                 3660       6
           A                  710 X 5
surplus    : 110

[3]
--------------------------------------------------------------------------------
wood               Type1                 3660       1
           A                  710 X 2
           B                  2000 X 1
surplus    : 240

[4]
--------------------------------------------------------------------------------
wood               Type1                 3660       15
           E                  625 X 2
           B                  2000 X 1
surplus    : 410

[5]
--------------------------------------------------------------------------------
wood               Type1                 3660       8
           D                  620 X 2
           B                  2000 X 1
surplus    : 420

[6]
--------------------------------------------------------------------------------
wood               Type1                 3660       6
           E                  625 X 5
surplus    : 535

utilization                              : 91.54%
total price                              :  10560.000
```