An Algorithm for the Two-Dimensional Cutting-Stock Problem Based on a Pattern Generation Procedure

Article *in* International Journal of Information and Management Sciences · June 2008

2 authors:

Ahmed Mellouli
Ecole Nationale d'Ingénieurs de Sfax

Abdelaziz Dammak
University of Sfax

Information and Management Sciences
Volume 19, Number 2, pp. 201-218, 2008

An Algorithm for the Two-Dimensional Cutting-Stock Problem Based on a Pattern Generation Procedure

Ahmed Mellouli
University of Sfax
Tunisia

Abdelaziz Dammak
University of Sfax
Tunisia

Abstract

This paper deals with the cut of a set of rectangular pieces, requested in large quantities and obtained with oriented guillotine cuts from a set of long rolls of material with standard widths in a way that minimises the total waste. For this class of problems, a heuristic based on three steps has been developed. First, an enumeration of all feasible non dominated patterns of the different widths with a pattern-generation procedure aims at constructing the constraints matrix. Second, a relaxation of the constraints of our problem is performed to obtain a linear formulation. Third a solution to the basic problem is generated through the solution of the associated problem with relaxed constraints. An example has been used to illustrate and clarify each step. The effectiveness of the algorithm has been tested by a set of random instances.

Keywords: Cutting Stock Problem, Column Generation, Heuristics.

1. Introduction

Cutting stock problems arise in a wide range of industries, with a common feature that some stock material is to be cut to produce smaller pieces of material in quantities matching orders received. The objective of this process is to minimise trim loss, i.e., the material needed by all cuts is minimised, (for application see Sculli, 1981; Farley, 1988, 1990b; Schultz, 1995; Suliman, 2001; Gradistar, 2005).

One of the most important variants of the cutting stock problem (CSP) is the two-dimensional cutting stock problem (2DCSP). This variant can be divided into regular (rectangular, circular, ...) and irregular shapes (example: Farley, 1988). Rectangular shapes can be obtained through guillotine or non guillotine, oriented or non oriented cutting. A guillotine cut means that each cut must go from one side of a rectangle straight to the opposite. So, each cut produces two sub-rectangles. An oriented cutting

Received November 2006; Revised February 2007; Accepted May 2007.
Supported in part by the "Industrial Management and Decision Aide : GIAD Laboratory" University of Sfax, Faculty of Economics and Management Sciences in Sfax, Tunisia, under NSC 99/UR/06/08.

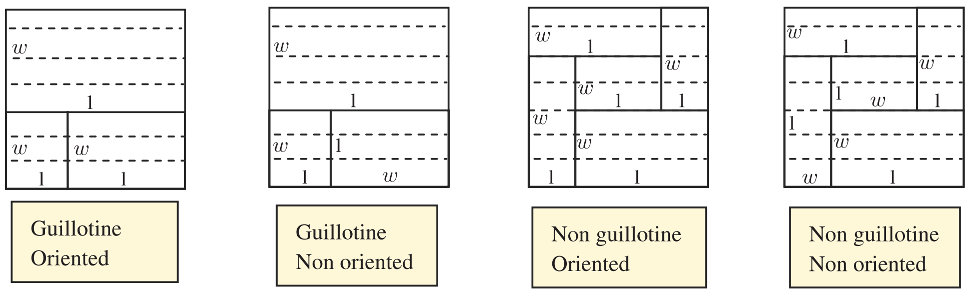


Figure 1. Types of cuts.

means that the lengths of rectangles are aligned parallel to length of the stock sheet or roll. So, a piece of length l and width w is different from a piece of length w and width l when $l \neq w$ (Figure 1).

In this work the two dimensional rectangular guillotine CSP is considered. This problem has been treated by Gilmore and Gomory (1965). A set of pieces is to be cut, using the least numbers of identical sheets. The solution is an integer programming model with a column generation procedure. Each column represents a cutting pattern and in each iteration the column that reduces the number of sheets is generated.

Integer linear programming models for orthogonal guillotine CSP has been considered by several authors (Gilmore and Gomory, 1961; Farley, 1988, 1990a; Schultz, 1995).

Since integer linear programming models are not practical to solve large size problems, some heuristics have been proposed for the generation of good cutting patterns to produce the columns of an integer programming problem. For example, a cutting pattern can be generated by dynamic programming (Farely, 1990b) or by constructive generation of guillotine cuts (Wang, 1983).

Benati (1997) developed a fast heuristic based on partial enumeration of all feasible patterns. The required rectangular pieces are cut from long rolls of materials.

Other researchers proposed a two-stage approach for the integer programming cutting stock problems, based on a rounding up procedure after a linear program (LP) relaxation (Johnson, 1986, 1997). To construct the matrix of constraint, Suliman (2001) proposed a branch and bound column generation method for the one-dimensional CSP that can be projected to the 2DCSP.

Suliman (2005) developed a three-stage sequential heuristic. In the first stage, a width-cutting pattern is determined. Determining the table length and the associated layout of the pieces length wise to produce a good cutting pattern is the second stage. In the final stage, the number of times in which the generated cutting pattern will be used is determined.

2. Problem Description

A set of rectangular order pieces with dimensions $w_1 \times l_1, \ldots, w_N \times l_N$ are requested with quantity d_i $(i = 1, \ldots, N)$. d_i is usually a very large number, more than one hundred, it is to be cut from rolls of material with standard width W_j $(j = 1, \ldots, K)$, each in sufficient length to satisfy the entire demand.

The cutting patterns must satisfy the following technological constraints:

- The pieces are obtained from oriented guillotine cut.
- The number of transversal cuts can not exceed the number of knives available in the machine (in this case, the number is equal to six).

For this problem it is required to determine the production schedule that minimises the total waste, i.e. material needed, while satisfying the given demand.

Notations: The following notations are used in this paper to facilitate communication:

$[x]$: Biggest integer lower than x.

$\lceil x \rceil$: Lowest integer bigger than x.

V_{kji}: Number of units of width w_i being cut according to the j^{th} pattern from the k^{th} roll.

L_{kj}: Length produced with the pattern V_{kj}.

J_k: Number of non dominated patterns of W_k roll width.

This problem can be formulated as follows:

$$\text{Minimise} \quad \sum_{k=1}^{K} \sum_{j=1}^{J_k} W_k L_{kj}$$

$$\text{Subject to} \quad \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left[\frac{L_{kj}}{l_i} \right] V_{kji} \geq d_i \qquad \text{for all } i \tag{1}$$

$$\sum_{i=1}^{N} V_{kji} \leq 6 \qquad \text{for all } k \text{ and } j \tag{2}$$

$$W_{k-1} < \sum_{i=1}^{N} V_{kji} w_i \leq W_k \qquad \text{for all } k \text{ and } j \tag{3}$$

$$L_{kj} \geq 0$$

The objective function represents the total area utilised from the different rolls which represent the sum of the area demand and the waste. Since the total area demand is constant, minimising the total area is equivalent to minimise the waste.

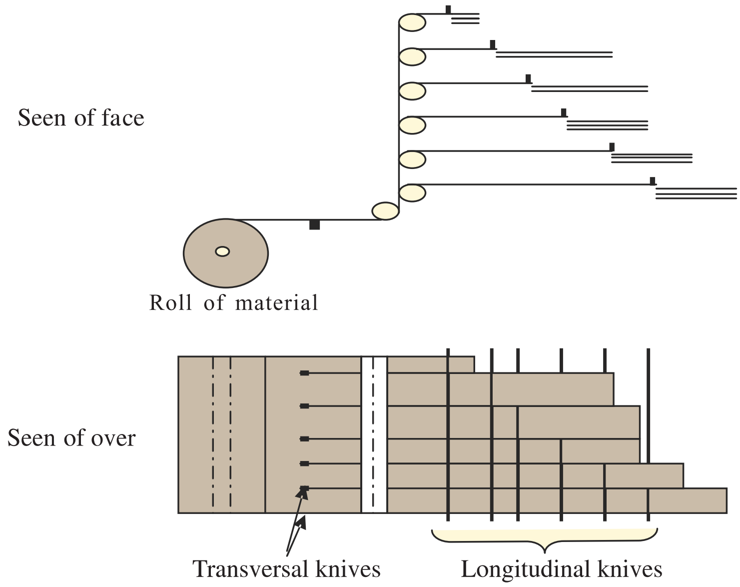


Figure 2. Schematic representation of cutting machine.

Constraints (1) guarantee that the demand of each type of items is satisfied.
Constraints (2) specify that the number of transversal cuts in each pattern can not exceed the number of knives available in the machine (in this case, the number is equal to six).
Constraints (3) reduce the number of patterns by selecting for each pattern width the closest feasible roll.

The mathematical formulation has two problems:

- The presence of the operator $[x]$ on the set of constraints (1) that represents a source of non linearity.
- The number of constraints is exponential for problems of medium and high size.

In order to solve the above mentioned problems, heuristics are used.

3. The Heuristic Approach

We can not apply the classic methods such as dynamic programming, branch and bound, simplex ... to search the optimal solution of this problem because the associated formulation includes of non-linear terms. That's why we have sacrificed the optimality and developed a heuristic. The heuristic approach is based on:

Step 1: Enumerate all feasible non-dominated patterns of the different widths with a pattern generation procedure to construct the matrix of constraints.

Step 2: Relax the constraints through an elimination of the operator $[x]$ from the set of constraints (1) to obtain a linear formulation.

Step 3: Generate a solution to the basic problem through the solution of the associated problem with relaxed constraints.

The heuristic approach flowchart is shown in Figure 3.

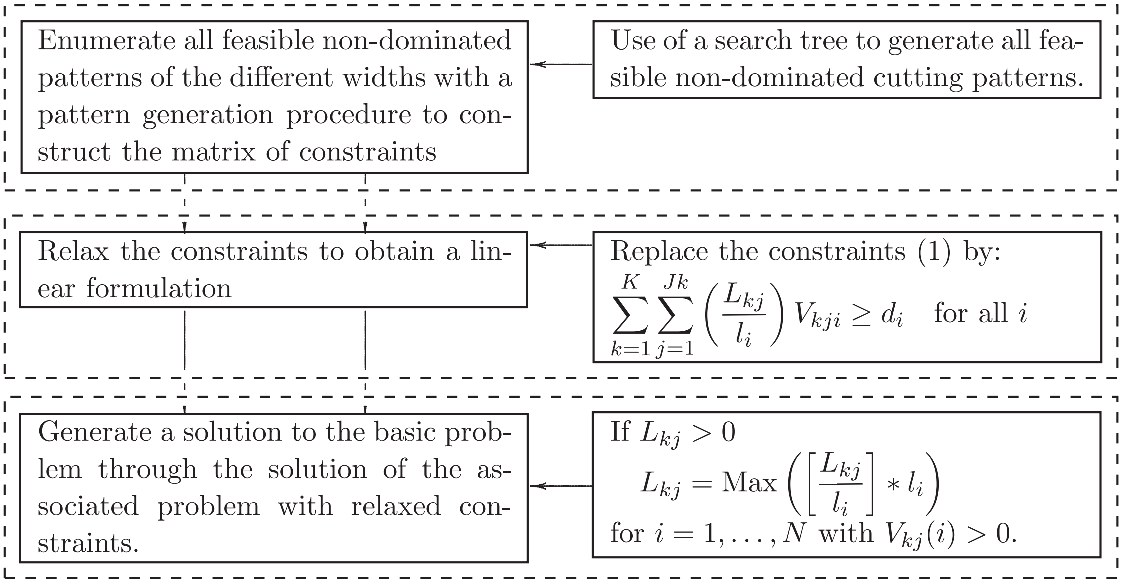


Figure 3. Heuristic approach flowchart.

To illustrate and clarify each step of this heuristic the following example is used.

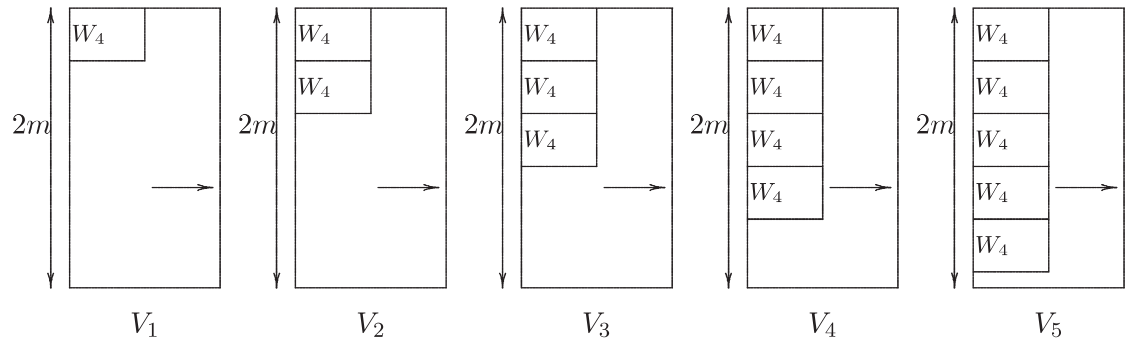
Four rectangular pieces with the specifications shown in Table 1 are to be cut from rolls with three widths: $W_1 = 2.5m$; $W_2 = 2.25m$; $W_3 = 2m$.

Table 1. Example data.

Item number	Width w_i	Length l_i	Demand d_i
1	1.35	0.5	200
2	1.05	0.4	150
3	0.78	0.8	400
4	0.37	0.9	400

Remark. a pattern V_1 is dominated by a pattern V_2 if the first is including in the second and each of the two patterns is feasible with the same roll width.

Example 1.

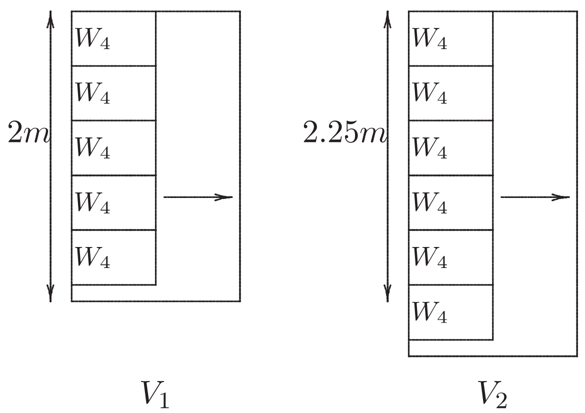


V_1	V_2	V_3	V_4	V_5
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	2	3	4	5

Figure 4. Example 1 patterns.

The Four first patterns (V_1 V_2 V_3 V_4) are include in the fifth one and all of these patterns belong to the set of pattern feasible with the roll of $2m$ width, therefore the set of patterns from the first to the fourth are dominated by the fifth (Figure 4).

Example 2.



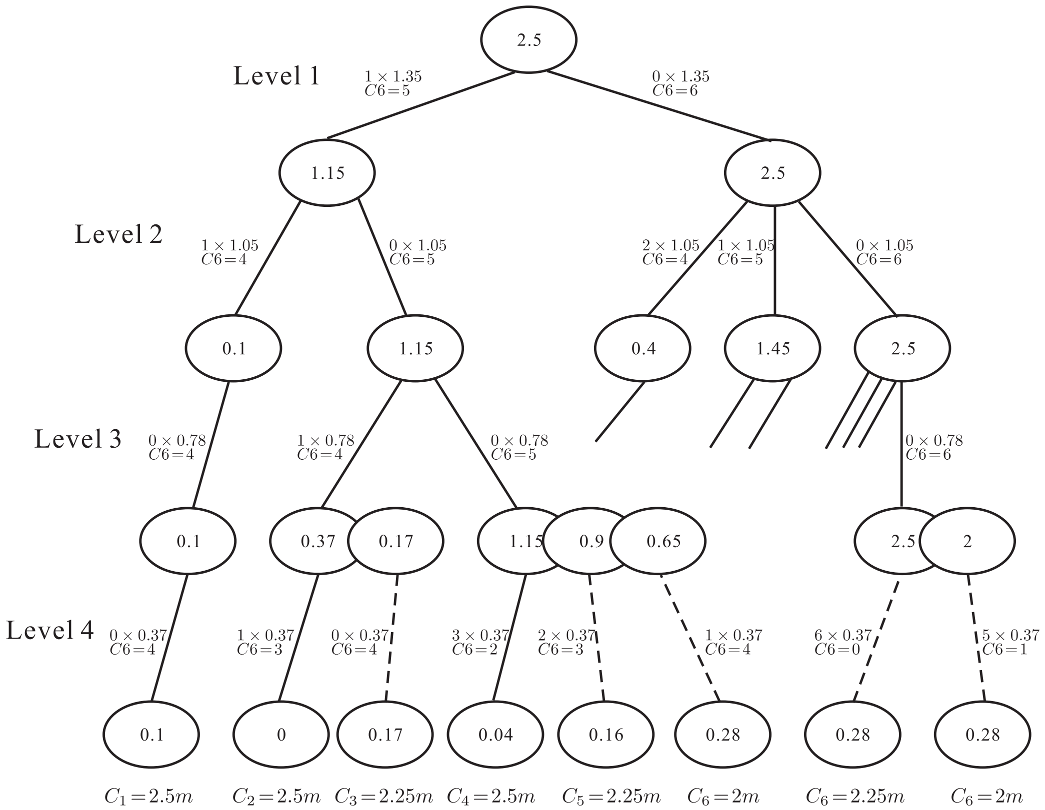
V_1	V_2
0	0
0	0
0	0
5	6

Figure 5. Example 2 patterns.

In this example the first pattern is include but not dominated by the second, because the first belongs to the set of patterns feasible with the roll of 2m width, but the second belongs to the set of patterns feasible with the roll of 2.25m width (Figure 5).

3.1. Construction the matrix of constraints (step 1)

The enumeration of all feasible non-dominated patterns of the different widths is achieved through a search tree with a number of levels equal to the number of orders (Figure 6). The levels of the tree represent the required widths that are arranged in a decreasing order with the largest in the first level and the smallest in the highest level of the tree. The first node of the tree represents the largest standard width available in the stock W_1. Thus a separate search tree is used to generate patterns associated with each node. The branches of level i of a search tree represent the number of units of the required width w_i which are cut in the j^{th} pattern, with the number of knives available for the successors branches.



The variable C6 species the number of knives available in each branch of the tree.

Figure 6. Example search tree.

The search tree is constructed by moving from bottom to top and from left to right with a set of rules presented below in the pattern generation procedure.

Based on the following search tree, all the non-dominated feasible patterns of the example are generated.

Notations:

A : Matrix of constraints formed by the non-dominated feasible patterns of the different widths rolls.

A_{ij} : Number of units of width w_i being cut in the j^{th} pattern.

C_j : Roll width used for the pattern j.

The non-dominated feasible patterns of different widths are presented in the following matrix:

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 2 & 2 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 3 & 2 & 1 & 1 & 0 & 1 & 0 & 3 & 2 & 0 & 2 & 1 & 4 & 3 & 6 & 5 \end{vmatrix}$$

The associated roll widths of theses patterns are presented in the following line Matrix:

$$C = [2.5\ \ 2.5\ \ 2.25\ \ 2.5\ \ 2.25\ \ 2.0\ \ 2.5\ \ 2.25\ \ 2.25\ \ 2.0\ \ 2.25\ \ 2.0\ \ 2.5\ \ 2.5\ \ 2.0\ \ 2.5\ \ 2.0\ \ 2.25\ \ 2.0]$$

Procedure of pattern generation

Step 1:

- Arrange the rolls in a decreasing of their width $W_1 > W_2 > \cdots > W_K$.
- Arrange the orders in a decreasing of their width $w_1 > w_2 > \cdots > w_N$.
- $j = 1$, $k = 1$ and $C6 = 6$.

Step 2: Generate the first column by:

$$A_{11} = \mathrm{Min}(C6, [W_1/w_1]) \tag{4}$$

$$C6 = C6 - A_{11} \tag{5}$$

$$A_{i1} = \mathrm{Min}(C6, [(W_1 - \sum_{z=1}^{i-1} A_{z1} \times w_z)/w_i]) \qquad \text{for } i = 2 \text{ to } N \tag{6}$$

$$C6 = C6 - A_{i1} \tag{7}$$

$C_1 = W_1$

Step 3: Step 3.1:

- If $k < K$, $k = k + 1$, go to step 3.2.
- If $k = K$, go to step 4.

Step 3.2: $$R = W_k - \sum_{z=1}^{N-1} A_{zj} \times w_z$$

- If $R < 0$, go to step 4

- Else go to step 3.3

Step 3.3: $C6 = 6 - \sum_{z=1}^{N-1} A_{zj}.$

- If $\mathrm{Min}(C6, [R/w_N]) < A_{N_j}$, $j = j + 1$ and generate a new column with the following elements:

  For $i = 1$ to $(N - 1)$ $\quad A_{ij} = A_{ij-1}$

  $A_{N_j} = \mathrm{Min}(C6, [R/w_N])$

  $C_j = W_k$, go to step 3.1

- If $\mathrm{Min}(C6, [R/w_N]) < A_{N_j}$, $C_j = W_k$, go to step 3.1.

Step 4:

- Search the biggest value i from the set $\{1, \ldots, (N - 1)\}$ with $A_{ij} > 0$.
- If this value exists it will be noted PZ, go to step 5
- Else go to step 6.

Step 5:

- $k = 1$, $j = j + 1$ and $C6 = 6$.
- Generate a new column with the following elements:
- For $i = 1$ to $(PZ - 1)$

$$A_{ij} = A_{ij-1} \tag{8}$$

$$C6 = C6 - A_{ij} \tag{9}$$

To fill the elements that precede (PZ, j).

- For $i = PZ$

$$A_{PZ_j} = A_{PZ_{j-1}} - 1. \tag{10}$$

$$C6 = C6 - A_{PZ_j} \tag{11}$$

To fill the (PZ, j) element.

- For $i = (PZ + 1)$ to N

$$A_{ij} = \mathrm{Min}(C6, [(W_1 - \sum_{z=1}^{i-1} A_{zj.} \times w_z)/w_i]) \tag{12}$$

$$C6 = C6 - A_{ij} \tag{13}$$

To fill the remaining elements of the new column:

$C_j = W_k$, go to step 3.

Step 6: End.

The pattern generation procedure flowchart is presented in Figure 7.

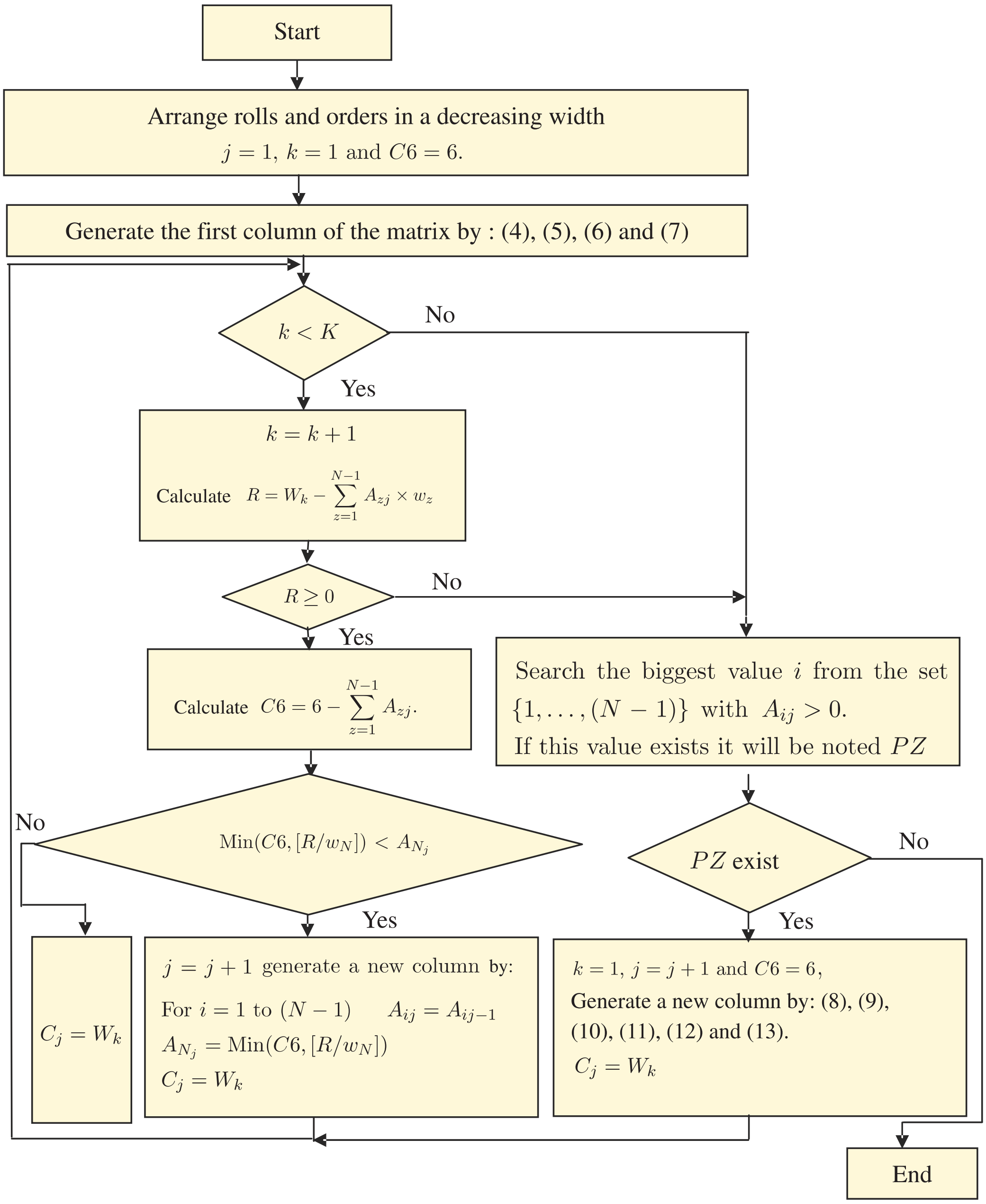


Figure 7. Pattern generation procedure flowchart.

Applying the procedure to the above example, the elements of the matrix A, are established in the following manner.

Step 1:

- Arrange the rolls in a decreasing of their width

$(W_1 = 2.5) > (W_2 = 2.25) > (W_3 = 2)$

- Arrange the orders in a decreasing of their width
  $(w_1 = 1.35) > (w_2 = 1.05) > (w_3 = 0.78) > (w_4 = 0.37)$
- $j = 1$, $k = 1$ and $C6 = 6$.

Step 2:

- Generate the first column by:
  $A_{11} = \text{Min}(6, [2.5/1.35]) = 1$ $\quad C6 = 6 - 1 = 5$
  $A_{21} = \text{Min}(5, [(2.5 - 1 \times 1.35)/1.05]) = 1$ $\quad C6 = 5 - 1 = 4$
  $A_{31} = \text{Min}(4, [2.5 - 1 \times 1.35 - 1 \times 1.05)/0.78]) = 0$ $\quad C6 = 4 - 0 = 4$
  $A_{41} = \text{Min}(4, [2.5 - 1 \times 1.35 - 1 \times 1.05)/0.37]) = 0$ $\quad C6 = 4 - 0 = 4$
  $C_1 = W_k = 2.5$

Step 3:

Step 3.1:

- $(k = 1) < (K = 3)$ then $k = k + 1 = 2$, go to step 3.2

Step 3.2:

$R = W_k - \sum_{z=1}^{N-1} A_{zj} \times w_z = 2.25 - (1 \times 1.35 + 1 \times 1.05) = -0.15$, $R < 0$ then go to step 4

Step 4:

- $PZ = 2$ then go to step 5

Step 5: $k = 1$, $j = j + 1 = 2$, $C6 = 6$.

- Generate a new column with the following elements:
- $A_{12} = A_{11} = 1$, $C6 = 6 - 1 = 5$
  (element that precede A_{22}).
- $A_{22} = A_{21} - 1 = 1 - 1 = 0$, $C6 = 5 - 0 = 5$
  (A_{22} element).
- $A_{32} = \text{Min}(5, [(2.5 - 1 \times 1.35 - 0 \times 1.05)/0.78]) = 1$ $\quad C6 = 5 - 1 = 4$
- $A_{42} = \text{Min}(4, [(2.5 - 1 \times 1.35 - 0 \times 1.05 - 1 \times 0.78)/0.37]) = 1$ $\quad C6 = 4 - 1 = 3$
  (remaining elements of the new column).
  $C_2 = W_k = 2.5$, go to step 3.

Step 3:

Step 3.1:

- $(k = 1) < (K = 3)$ then $k = k + 1 = 2$, go to step 3.2

Step 3.2:

$R = 2.25 - (1 \times 1.35 + 0 \times 1.05 + 1 \times 0.78) = 0.12$

- $R \geq 0$ then goto step 3.3

Step 3.3:

$$C6 = 6 - \sum_{z=1}^{N-1} A_{zj} = 6 - (1 + 0 + 1) = 4.$$

- $\mathrm{Min}(4, [0.12/0.37]) = 0) < A_{42}$ then $j = j + 1 = 3$, generate a new column with the following elements:
  $A_{13} = A_{12} = 1$, $A_{23} = A_{22} = 0$, $A_{33} = A_{32} = 1$, $A_{43} = \mathrm{Min}(4, [0.12/0.37]) = 0$
  $C_3 = W_k = 2.25$, goto step 3.1

Step 3.1:

- $(k = 2) < (K = 3)$ then $k = k + 1 = 3$, go to step 3.2

Step 3.2:

- $R = 2 - (1 \times 1.35 + 0 \times 1.05 + 1 \times 0.78) = -0.13$; $R < 0$ then go to step 4

Step 4:

- $PZ = 3$ then go to step 5

Step 5: $k = 1$, $j = j + 1 = 4$, $C6 = 6$.

- Generate a new column with the following elements:
- $A_{14} = A_{13} = 1$, $C6 = 6 - 1 = 5$
- $A_{24} = A_{23} = 0$, $C6 = 5 - 0 = 5$
  (element that precede A_{34}).
- $A_{34} = A_{33} - 1 = 1 - 1 = 0$, $C6 = 5 - 0 = 5$
  (A_{34} element).
- $A_{44} = \mathrm{Min}(5, [(2.5 - 1 \times 1.35 - 0 \times 1.05 - 0 \times 0.78)/0.37]) = 3$ $C6 = 5 - 3 = 2$
  (remaining elements of the new column).
  $C_4 = W_k = 2.5$, go to step 3.

Step 3:

Step 3.1:

- $(k = 1) < (K = 3)$ then $k = k + 1 = 2$, go to step 3.2

Step 3.2:

- $R = 2.25 - (1 \times 1.35 + 0 \times 1.05 + 0 \times 0.78) = 0.9$
- $R > 0$ then go to step 3.3

Step 3.3:

$$C6 = 6 - \sum_{z=1}^{N-1} A_{zj} = 6 - 1 = 5.$$

- $(\mathrm{Min}(5, [0.9/0.37]) = 2) < (A_{44} = 3)$ then $j = j + 1 = 5$, generate a new column with the following elements:

$A_{15} = A_{14} = 1,\ A_{25} = A_{24} = 0,\ A_{35} = A_{34} = 0,\ A_{45} = \text{Min}(5, [0.9/0.37]) = 2$

$C_5 = W_k = 2.25$, goto step 3.1

The procedure continues to produce the entire non dominated patterns presented in the matrix A and the roll width used for each non-dominated pattern presented in the matrix line C.

3.2. Relaxation of the problem (step 2)

Relax the constraints of our problem by an elimination of the operator $[x]$ from the set of constraints (14) to obtain a linear formulation.

The relaxed problem of our example is to cut four orders, with the specifications shown in Table 2, from rolls with three widths: $W_1 = 2.5m$; $W_2 = 2.25m$; $W_3 = 2m$.

Table 2. Example data.

Item number	Width w_i	accumulated length $l_i \times d_i$
1	1.35	$0.5 \times 200 = 100$
2	1.05	$0.4 \times 150 = 60$
3	0.78	$0.8 \times 400 = 320$
4	0.37	$0.9 \times 400 = 360$

The relaxed problem is:

$$\begin{cases} \text{Min}(C.L) \\ \text{Subject to} \\ \qquad A.L \geq B & (14) \\ \qquad\ L \geq 0 & (15) \end{cases}$$

where A: Matrix of constraints;

C: Associated rolls widths of patterns presented in the matrix A;

L : Lengths to produce with the different patterns to satisfy the accumulated lengths of orders;

B : Accumulated length of each rectangular type of pieces.

For our example: B is presented in the following column matrix:

$$B = \begin{vmatrix} d_1 \times l_1 = 100 \\ d_2 \times l_2 = 60 \\ d_3 \times l_3 = 320 \\ d_4 \times l_4 = 360 \end{vmatrix}$$

The solution of the linear problem is:

$$L = [0;\ 100;\ 0;\ 0;\ 0;\ 0;\ 21.0092;\ 0;\ 17.99;\ 0;\ 0;\ 0;\ 0;\ 0;\ 101.1;\ 0;\ 0;\ 20;\ 0]$$

$$V_2 = \begin{vmatrix} 1 \\ 0 \\ 1 \\ 1 \end{vmatrix} \begin{matrix} L_2 = 100m \\ W = 2.5m \end{matrix} \qquad V_7 = \begin{vmatrix} 0 \\ 2 \\ 0 \\ 1 \end{vmatrix} \begin{matrix} L_7 = 21.01m \\ W = 2.5m \end{matrix} \qquad V_9 = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \end{vmatrix} \begin{matrix} L_9 = 17.99m \\ W = 2.25m \end{matrix}$$

$$V_{15} = \begin{vmatrix} 0 \\ 0 \\ 2 \\ 1 \end{vmatrix} \begin{matrix} L_{15} = 101.01m \\ W = 2m \end{matrix} \qquad V_{18} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 6 \end{vmatrix} \begin{matrix} L_{18} = 20m \\ W = 2.25m \end{matrix}$$

3.3. Solution of the basic problem (step 3)

Generate a solution to our problem from the solution of the associated problem with relaxed constraints (Figure 8).

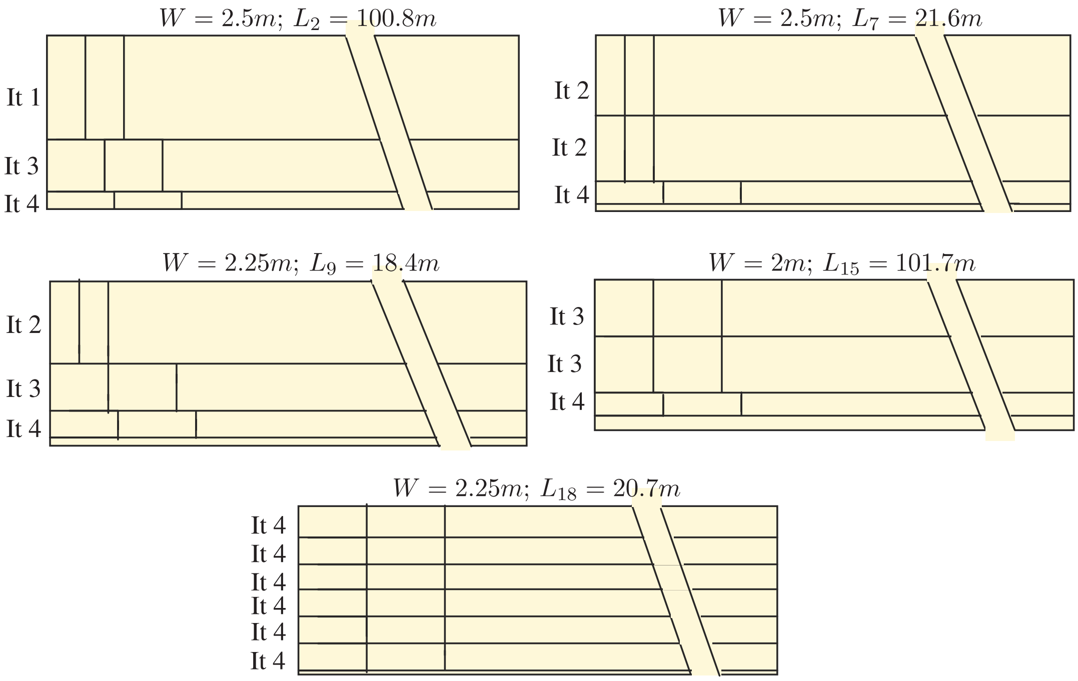


Figure 8. Production schedule.

If $L_j > 0$, $L_j = \text{Max}\left(\left\lceil \frac{L_j}{l_i} \right\rceil * l_i\right)$ for $i = 1$ to N with $V_j(i) > 0$.

The solution of our problem is:

$$V_2 = \begin{vmatrix} 1 \\ 0 \\ 1 \\ 1 \end{vmatrix} \begin{matrix} L_2 = 100.8m \\ W = 2.5m \end{matrix} \quad V_7 = \begin{vmatrix} 0 \\ 2 \\ 0 \\ 1 \end{vmatrix} \begin{matrix} L_7 = 21.6m \\ W = 2.5m \end{matrix} \quad V_9 = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \end{vmatrix} \begin{matrix} L_9 = 18.4m \\ W = 2.25m \end{matrix}$$

$$V_{15} = \begin{vmatrix} 0 \\ 0 \\ 2 \\ 1 \end{vmatrix} \begin{matrix} L_{15} = 101.7m \\ W = 2m \end{matrix} \quad V_{18} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 6 \end{vmatrix} \begin{matrix} L_{18} = 20.7m \\ W = 2.25m \end{matrix}$$

4. Experimental Results

The effectiveness of the proposed heuristic has been tested to check both the computing time and the quality of the solution.

4.1. Computation time

Although this heuristic have some limitations:

- The number of generated columns increases exponentially with the size of the problem.
- The resolution of the relaxed problem requires a high computing time.

We succeeded in decreasing the intensity of these limitations by the elimination of all dominated columns from the matrix of constraints.

To show this success, we studied the effect of elimination of all dominated columns on the number of generated columns and on the computing time.

The experimental results are presented in the following tables and graphs:

Table 3. Number of generated columns.

Number of orders	5	10	15	20	25	35
Number of generated columns	95	478	536	1441	1441	8022
Number of generated non-dominated columns	28	176	206	610	610	4156

Table 4. Computing time.

Number of orders	5	10	15	20	25	35
Computing time in (sec) (with dominated and non-dominated columns)	1.5	2.2	3	13	37	395
Computing time in (sec) (with only non-dominated columns)	1	1.5	1.8	4.5	10	114

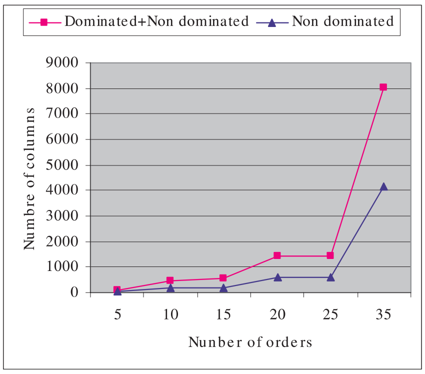


Figure 9. Number of generated columns.

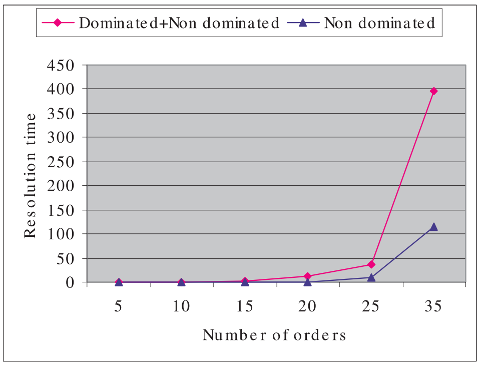


Figure 10. Computing time.

4.2. Quality of the solutions

The quality of the solutions depends on both the quality of the relaxation and the procedure to generate a solution of the basic problem from the solution of the relaxed problem.

The quality of the solutions has been evaluated by a comparison between the solution presented by this heuristic and the solution of the relaxed problem that represent a lower bound for the optimal solution.

The experimental results of the application of this heuristic on some problems which are randomly generated are presented in Table 5.

Table 5. Difference in percentage between the square of solution and the lower bound.

Problem number	1	2	3	4	5	6
Square of solution	40116	29599	29460	26200	70512	24098
Lower bound	40088	29557	29428	26125	70210	23953
Difference in percentage	0.07	0.14	0.11	0.29	0.43	0.60

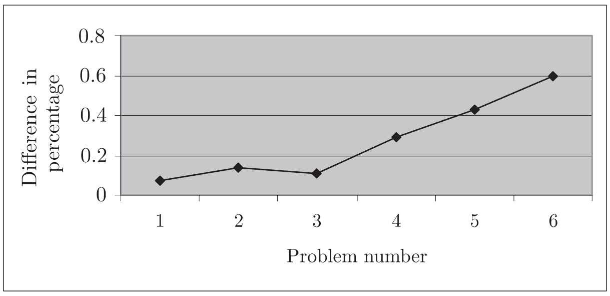


Figure 11. Difference in percentage between the square of solutions and the lowers bounds.

The comparison of the results given by this heuristic with the lowers bounds of the optimal solutions shows a great success of this heuristic to converge to the optimal solution, because the difference between the solutions suggested and the associated lower bounds does not exceed a rate of 1%.

This comparison also shows the reliability of the relaxation suggested which determines a good lower bound very close to the optimal solution.

5. Conclusion

In this work an algorithm to solve an important variant of the two-dimensional cutting-stock problem encountered in paper and sheet industries, was proposed. This algorithm first identifies all feasible non dominated combinations through a pattern generation procedure, to construct the constraints matrix. Then a relaxation of the problem is performed in order to obtain a linear formulation, and finally a solution to our original problem is generated from the solution of the relaxed problem. The propose algorithm has been tested on a set of randomly generated problems. The experimental work shows the following performances of our heuristic: a good quality of the solution (the difference between the solution and the lower bound to the optimal solution, is less than 1%) and this heuristic is capable to solve problems with medium size (not only small size) because the search tree does not consider dominated feasible patterns.

Acknowledgement

The authors thank the referees for their constructive comments, which have contributed to the improvement of this paper.

References

[1] Benati, S., *An algorithm for a cutting stock problem on a strip*, Journal of the Operational Research Society, Vol.48, pp.288-294, 1997.

[2] Farley, A., *Mathematical programming models for cutting stock problems in the clothing industry*, Journal of Operational Research Society, Vol.39, pp.41-53, 1988.

[3] Farley, A., *A note on bounding a class of linear programming problems, including cutting stock problems*, Journal of Operations Research, Vol.38, pp.922-923, 1990a.

[4] Farley, A., *The cutting stock problem in the canvas industry*, European Journal of Operational Research, Vol.44, pp.247-255, 1990b.

[5] Gilmore, P. C. and Gomory, R. E., *Multistage cutting-stock problems of two and more dimensions*, Operation Research, Vol.13, pp.94-120, 1965.

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

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

[8] Johnson, R. E., *Rounding algorithms for cutting stock problems*, Asia-Pacific Journal of Operational Research, Vol.3, pp.166-171, 1986.

[9] Johnson, M. P., Rennick, C. and Zak, E., *Skiving addition to the cutting stock problem in the paper industry*, Journal of Siam Review, Vol.39, No.3, pp.472-483, 1997.

[10] Schultz, T. A., *Application of linear programming in a gauge splitting operation*, Operations Research, Vol.43, No.5, pp.752-757, 1995.

[11] Sculli, D., *A stochastic cutting stock procedure: cutting rolls of insulating tape*, Management Science, Vol.27, pp.946-952, 1981.

[12] Suliman, Saad, M. A., *Pattern generating procedure for the cutting stock problem*, International Journal of Production Economics, Vol.74, pp.293-301, 2001.

[13] Suliman, Saad, M. A., *A sequential heuristic procedure for the two-dimensional cutting-stock problem*, International Journal of Production Economics, 2005.

[14] Wang, P. Y., *Two algorithms for constrained two dimensional cutting stock problems*, Operation Research, Vol.31, pp.573-586, 1983.

Authors' Information

Abdelaziz Dammak, received his Bachelor degree in applied fundamental mathematics, Master degree in Operations Research from Scientific & Medical University of Grenoble (France), PhD in applied mathematics in the field of electronics from University of Paris-Sud, Orsay (France) and HDR in Quantitative Methods field at the Faculty of Economics and Management Sciences in Sfax. He teaches Operations Research at the University of Sfax. His research interests are in combinatorial optimization problems: graph theory, mathematical programming and development of heuristics. More recently, his research focus on university timetabling applied to Tunisian universities. He is author of different articles published in different international journals and the Proceedings of PATAT'06.

Department of Applied Quantitative Methods University of Sfax, FSEG, Route de l'Aerodrome, km 4, BP 1088, 3018 Sfax Tunisia.

E-mail: Abdelaziz.dammak@fsegs.rnu.tn TEL : +216-98-667-612

Ahmed Mellouli, is a engineer in electromechanical from the National School of Engineering of Sfax, received his master degree in Operations Research & Production Management from the Faculty of Economics and Management Sciences in Sfax. He is a PhD student working in the National School of Engineering of Sfax. His main fields of research include: Cutting stock, Mathematical Programming, and Heuristic Methods.

Mechanics Modelling and Production Research Unit (U2MP) University of Sfax, Sfax Engineering School (E.N.I.S.), B.P. W, 3038 Sfax Tunisia.

E-mail: Ahmed_tu2002@yahoo.fr TEL : +216-97-229-619