Cutting Problem: a Fast Algorithm for Computing a Near-Optimal Solution

Fabrice Chauvet, Ahmedou Ould Haouba, Jean-Marie Proth

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Cutting Problem: a Fast Algorithm for Computing a Near-Optimal Solution

Fabrice Chauvet - Ahmedou Ould Haouba
Jean-Marie Proth

N° 3421
Mai 1998

THÈME 4

Rapport de recherche

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

PROBLEME DE DECOUPE :
UN ALGORITHME RAPIDE POUR CALCULER UNE SOLUTION
PROCHE DE L'OPTIMUM.

Fabrice Chauvet[(2)], Ahmedou Ould Haouba[(1)] and Jean-Marie Proth[(2)]

Résumé :

Un ensemble de barres de longueur n.Δ est disponible en stock. L'objectif de ce papier est de présenter un algorithme efficace pour calculer une solution proche de l'optimum du problème qui consiste à découper des barres de longueur i.Δ, i=1, 2, ..., n dans les barres de longueur n.Δ de façon à satisfaire des demandes connues. Le critère à minimiser est le nombre de barres utilisées.

Mots clefs : Découpe, Programmation Linéaire.

(1) Faculté des Sciences et Université Technique de Nouakchott, Mauritanie.
(2) INRIA 4, rue Marconi, 57070 - Metz France.

CUTTING PROBLEM:

A FAST ALGORITHM FOR COMPUTING A NEAR-OPTIMAL SOLUTION

Fabrice Chauvet[(2)], Ahmedou Ould Haouba[(1)] and Jean-Marie Proth[(2)]

Abstract:

A set of bars of length $n\Delta$ is available in stock. The purpose of this paper is to provide an efficient algorithm to compute a near optimal solution of the problem which consists in cutting bars of length $i\Delta$ for $i = 1, 2, \ldots, n-1, n$, out of bars of length $n\Delta$ while minimising scrapes or, equivalently, while minimising the total number of bars used.

Keywords: Cutting problem, Linear programming.

(1) Faculty of Sciences and Technical University of NOUAKCHOTT, MAURITANIA

(2) INRIA 4,rue Marconi 57070 Metz France

Mailing address: Dr. PROTH J.-M., 22 rue du Colombier, 57420 - POUILLY FRANCE

1. INTRODUCTION

Cutting problems, considered in their most general form, are NP-hard. As a consequence, only small size problems can be solved optimally while real-life problems require heuristic algorithms. The most common of these heuristic algorithms are Next Fit (NF) heuristic (see [2] and [7]) which consists in selecting the stored bars in a random order and assigning the ordered bars also in a random order. If the next ordered bar cannot be assigned to the stored bar under consideration, a new ordered bar is selected and the unused part of the previous ordered bar is considered as scrap. The First Fit (FF) heuristic (see [2], [7] and [9]) is similar to NF, except that the unused part of a stored bar is considered as a stored bar and can be used later. The First Fit Decreasing (FFD) algorithm (see [2] and [7]) is the heuristic NF in which the ordered bars are selected in the decreasing order of their length. Numerous other heuristic algorithms are available as, for instance, Best Fit (BF) (see [2], [7] and [9]), Worst Fit (WF) (see [2], [7] and [9]), Almost Worst Fit (AWF) (see [2] and [7]) to quote only a few. All these approaches are similar, except in the choice of the stored and ordered bars, and in the way the unused parts of the stored bars are processed.

In the remaining of this paper, we consider a more specific problem encountered in mass production. In this case, the stored bars are of the same length, which is a multiple of an elementary length Δ. The lengths of the required bars are also multiple of Δ. The goal is to provide s_i bars of length $i\Delta$ for $i = 1, 2, \dots, n$, using the minimal number of stored bars of length $n\Delta$ or, equivalently, while minimising scrapes. We are looking for a near optimal solution of this problem.

In section 2, we propose an iterative approach to compute all the cutting patterns of a bar of length $n\Delta$. The problem is presented in section 3. Basic results associated with the optimal solutions are presented in section 4. Section 5 is devoted to the algorithm which provides the near optimal solution. Section 6 is the conclusion.

2. COMPUTATION OF THE CUTTING PATTERNS

2.1. Notations

We denote by y_i the number of bars of length $i\Delta$ which are cut out of a stored bar of length $n\Delta$. Since we are looking for all the feasible solutions, we have to find all the integer and positive solutions of:

$$\sum_{i=1}^{n} i.y_i = n \qquad (1)$$

We represent the cutting patterns as a matrix:

$$C_n = \begin{bmatrix} c_1^1 & . & c_i^1 & . & c_n^1 \\ . & . & . & . & . \\ c_1^k & . & c_i^k & . & c_n^k \\ . & . & . & . & . \\ c_1^{q_n} & . & c_i^{q_n} & . & c_n^{q_n} \end{bmatrix}$$

where q_n is the total number of feasible patterns when a bar of length nΔ can be cut into bars of length iΔ, i = 1, 2, ..., n. The length of the i-th bar cut out from a bar of length nΔ when following the k-th pattern is $c_i^k \in \{0, 1, ..., n\}$. It is easy to realise that:

$$C_1 = [1]$$

$$C_2 = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Two remarks should be made at this level:

(i) The sum of the elements of a row of matrix C_n is equal to n.

(ii) We classify the elements of each row in the decreasing order of their value in order to avoid identical patterns.

2.2. An iterative algorithm

Let us consider a bar of length nΔ. This bar can be cut out in two parts the lengths of which are multiple of Δ. There are n-1 possibilities to do so if we consider only the cases when both parts are greater than 0. These n-1 possibilities are the pairs {iΔ, (n-i)Δ} for i = 1, 2, ..., n-1. We further consider all the patterns available to cut a bar of length iΔ into bars of length jΔ for j = 1, 2, ..., i. These patterns are given by matrix C_i, and the number of such patterns is q_i. In order not to compute the same pattern more than once, we only consider the rows of C_i whose elements are less than or equal to n-i. We complete each one of these rows by element n-i, and we do the same for i = 1, 2, ..., n-1. Finally, we complete matrix C_n by the row [n, 0, ..., 0].

The number of rows of C_n, that is q_n, can be obtained by applying the following iterative computation:

$$q_0 = 1$$

$$q_n = \sum_{i=0}^{n-1} z_n^i$$

where:

$$z_n^i = \begin{cases} q_i \text{ if } i \le \dfrac{n}{2} \\ q_i - \sum\limits_{j=0}^{2i-n-1} z_i^j \text{ if } i > \dfrac{n}{2} \end{cases}$$

In this formulation, z_n^i represents the number of rows of C_i whose elements are less than or equal to n-i, except z_n^0 which is equal to one and represents the row [n, 0, ..., 0]. The proof of this formulation can be done by induction.

We denote by $[C_n]_k$ the k-th row of matrix C_n, that is the row which represents the k-th pattern available to cut a bar of length $n\Delta$ into bars whose lengths are multiple of Δ. Using this notation, the algorithm to compute the required patterns is ***Algorithm 1*** presented hereafter.

Algorithm 1

1. Set: $q_1=1$ and $C_1=[1]$
2. **For j=2 to n do:**

(Building matrix C_j)

2.1. k=1

2.2. $[C_j]_k=[j, 0, 0, ..., 0]$

This row is made with j followed by (j-1) times 0.

2.3. **For r=1 to j-1 do:**

2.3.1. **For p=1 to q_r do:**

2.3.1.1. If all the elements of $[C_r]_p$ are less than or equal to j-r do:

2.3.1.1.1. k=k+1

2.3.1.1.2. $[C_j]_k=[j-r, [C_r]_p]$

2.4. $q_j=k$

In table 1, we provide some values of q_n as a function of n.

Table 1: Number of rows of the working matrix with respect to n

n	5	10	15	16	17	18	19	20
q_n	7	42	176	231	297	385	490	627

3. PROBLEM FORMULATION

3.1. Working matrix

From matrix C_n, we derive matrix Y_n defined as follow:

- Y_n has the same number of rows and column as C_n,
- $y_{k,i}$, which is the element of the k-th row and the i-th column of Y_n, represents the number of bars of length $i\Delta$ cut out of a bar of length $n\Delta$ when following the k-th pattern.

In other words, the elements of the k-th row of Y_n, for $k = 1, 2, \ldots, q_n$, satisfy relation (1).

For instance:

$$Y_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 3 & 0 & 0 \end{bmatrix}$$

Y_3 is the working matrix associated with the stored bars of length $n\Delta$.

There exists a row k_1 of the working matrix such that:

- $y_{k_1,i} = 0$ for $i = 1, 2, \ldots, n-1$
- $y_{k_1,n} = 1$

There also exists n-2 rows $k_2, k_3, \ldots, k_{n-1}$ such that, for $r = 2, 3, \ldots, n-1$:

- $y_{k_r,1} = r - 1$
- $y_{k_r,n-r+1} = 1$
- $y_{k_r,i} = 0$ for $i = 1, 2, \ldots, n;\ i \neq 1;\ i \neq n - r + 1$

Finally, there also exists a row k_n such that:

- $y_{k_n,1} = n$
- $y_{k_n,i} = 0$ for $i = 2, 3, \ldots, n$.

In the remaining of this paper, we assume that $k_r = r$ for $r = 1, 2, \ldots, n$; in other words, we assume that the n rows mentioned above are the first n rows of matrix Y_n. The q_n-n other rows are ordered randomly.

3.2. Problem Q

The problem, denoted by Q, can be formulated as follows (x_k is the number of stored bars which are cut according to the k-th pattern):

(Q) $\qquad \text{minimise} \sum_{k=1}^{q_n} x_k \qquad\qquad (2)$

s.t.

$$(\mathcal{D}) \begin{cases} \sum_{k=1}^{q_n} x_k . y_{k,i} \geq s_i \text{ for } i = 1, 2, ..., n & (3) \\ x_k \geq 0 \text{ and } x_k \text{ integer for } k = 1, 2, ..., q_n & (4) \end{cases}$$

where s_i is the number of bars of length $i\Delta$ which must be provided to the customers. We denote by $\mathcal{D}$ the set of X's defined by (3) and (4). A set: $X = [x_1, x_2, ..., x_{q_n}]$ which minimises (2) while satisfying constraints (3) and (4) is called "optimal solution of problem $\mathcal{Q}$ ".

4. PROPERTIES OF PROBLEM $\mathcal{Q}$

We denote by $\mathcal{C}$ the set of X's defined by:

$$(\mathcal{C}) \begin{cases} \sum_{k=1}^{q_n} x_k . y_{k,i} = s_i \text{ for } i = 2, 3, ..., n & (5) \\ \sum_{k=1}^{q_n} x_k . y_{k,1} \geq s_1 & (6) \\ x_k \geq 0 \text{ and } x_k \text{ integer for } k = 1, 2, ..., q_n & (4) \end{cases}$$

Hereafter, we denote by $\mathcal{P}$ the problem defined by (2), (4), (5) and (6). In other words, $\mathcal{P}$ is the problem:

$$(\mathcal{P}) \qquad \text{minimise} \sum_{i=1}^{q_n} x_k$$

X belonging to $\mathcal{C}$.

Result 1.

There exists an optimal solution to $\mathcal{P}$, i.e. $\mathcal{C}$ is not empty.

Proof:

According to the row order introduced in subsection 3.1, the following solution is a feasible solution to problem $\mathcal{P}$:

$$x_i = s_{n+1-i} \text{ for } i = 1, 2, ..., n-1$$

$$x_n = \left\lceil \frac{s_1 - \sum_{i=1}^{n-1} (i-1) x_i}{n} \right\rceil^+$$

where $\lceil a \rceil^+$ is the smaller positive integer greater than or equal to a. Furthermore, $x_i=0$ for $i=n+1, n+2, ..., q_n$.

Thus, an optimal solution to problem $\mathcal{P}$ exists since $\mathcal{C}$ is not empty.

QED

Result 2

An optimal solution to $\mathcal{P}$ is also an optimal solution to Q.

Proof:

Result 2 is easy to understand by considering that all the bars of length $i\Delta$, $i = 2, 3, ..., n$ of a solution to Q which are in excess with regard to the demand can be cut into bars of length Δ. The solution obtained by doing so is solution of $\mathcal{P}$. Indeed, the bars in excess can be selected in various ways; in other words, various solutions of $\mathcal{P}$ can be derived from the solution of Q.

QED

We generalise the definition of problem $\mathcal{P}$ by replacing constraint (6) by constraint (18):

$$\sum_{k=1}^{q_n} x_k . y_{k,1} \geq s_1 + \beta \tag{18}$$

where $\beta \geq 0$. We denote this problem by $\mathcal{P}^\beta$ in the remainder of this paper. As a consequence, the set of solutions to $\mathcal{P}^\beta$ is contained in the set of solutions to $\mathcal{P}$. Note that a solution to this problem is usually not integer.

Result 3

Let X^1 and X^2 be two optimal solutions to problem $\mathcal{P}^\beta$.

Then $\sum_{k=1}^{q_n} x_k^2 . y_{k,1} = \sum_{k=1}^{q_n} x_k^1 . y_{k,1}$.

Proof:

Both solutions provide the exact numbers of bars of lengths $i\Delta$, $i=2, 3, ..., n$. Furthermore, the total number of bars used in both solutions is the same since these solutions are optimal. Thus $\sum_{k=1}^{q_n} x_k^1 . y_{k,1}$ and $\sum_{k=1}^{q_n} x_k^2 . y_{k,1}$, which are the differences between the total length of the stored bars used and the total length of bars of lengths $i\Delta$, $i=2, 3, ..., n$ produced in solutions X^1 and X^2 respectively, are equal.

QED

Result 4

Let X^* be a feasible solution to problem $\mathcal{P}^\beta$. If constraint (18) is saturated for X^*, i.e. if equality holds in constraint (18), then:

(i) X^* is optimal, and the values of the criterion which corresponds to this solution is $\mathcal{Z}^\beta = (\sum_{i=1}^{n} is_i + \beta)/n$.

(ii) any other optimal solution to problem $\mathcal{P}^\beta$ saturates constraint (18).

Proof:

a. Let X be a feasible solution to $\mathcal{P}^\beta$. By adding (5) and (18) after multiplying both sides of the relations by i, we obtain, considering relation (1):

$$n\sum_{k=1}^{q_n} x_k \geq \sum_{i=1}^{n} i.s_i + \beta.$$

According to the hypothesis provided in result 4, we obtain for X^*:

$$n\sum_{k=1}^{q_n} x_k^* = \sum_{i=1}^{n} i.s_i + \beta.$$

As a consequence of the two previous relations, $\sum_{k=1}^{q_n} x_k^* \leq \sum_{k=1}^{q_n} x_k$ for any feasible solution X of $\mathcal{P}^\beta$ and X^* is optimal. This complete the proof of (i).

b. (ii) is a consequence of result 3.

QED

The next result is the first tentative to connect the real solutions to problems $\mathcal{P}^\beta$ to their integer solutions.

Result 5

Let β_1 and β_2 be two real numbers such that $0 \leq \beta_1 \leq \beta_2$. Assume that constraint (18) is saturated for the optimal solutions to problem $\mathcal{P}^{\beta_2}$ and that at least one optimal solution to problem $\mathcal{P}^{\beta_2}$ is integer. Then, if $\beta_2-\beta_1<$**n**, any integer solution to $\mathcal{P}^{\beta_2}$ is also an optimal integer solution to $\mathcal{P}^{\beta_1}$.

Proof:

We denote by $\mathcal{C}^\beta$ the set of feasible solutions defined by (5) and (18) and we consider $X \in \mathcal{C}^{\beta_1} - \mathcal{C}^{\beta_2}$. This solution verifies:

$$s_1 + \beta_1 \leq \sum_{k=1}^{q_n} x_k . y_{k,1} < s_1 + \beta_2$$

and:

$$\sum_{k=1}^{q_n} x_k . y_{k,i} = s_i \quad \text{for } i = 2, 3, ..., n$$

By adding these relations side by side after multiplying these sides by i, we obtain:

$$\frac{\sum_{i=1}^{n} i.s_i + \beta_1}{n} \leq \sum_{k=1}^{q_n} x_k < \frac{\sum_{i=1}^{n} i.s_i + \beta_2}{n} \tag{19}$$

We denote by $\mathcal{Z}^{\beta}$ the optimal solution of problem $\mathcal{P}^{\beta}$. The following relations hold taking into account the hypotheses of result 5:

$$\frac{\sum_{i=1}^{n} i.s_i + \beta_2}{n} - \frac{\beta_2 - \beta_1}{n} \leq \sum_{k=1}^{q_n} x_k < \frac{\sum_{i=1}^{n} i.s_i + \beta_2}{n}$$

$$\mathcal{Z}^{\beta_2} - \frac{\beta_2 - \beta_1}{n} \leq \sum_{k=1}^{q_n} x_k < \mathcal{Z}^{\beta_2} \text{ (see result 4).}$$

$\mathcal{Z}^{\beta_2}$ is integer and $0 \leq \frac{\beta_2 - \beta_1}{n} < 1$. Thus, whatever $X \in \mathcal{C}^{\beta_1} - \mathcal{C}^{\beta_2}$, the related objective function is not integer. This complete the proof.

QED

Result 6

The hypotheses are the same as the ones of result 5, except the fact that $\beta_2 - \beta_1 = \mathbf{n}$.

We denote by $\mathcal{S}^{\beta_2}$ the set of optimal integer solutions to problem $\mathcal{P}^{\beta_2}$. Then:

(i) If there exists $X \in \mathcal{S}^{\beta_2}$ such that $x_n > 0$, then:

- $[x_1, x_2, ..., x_{n-1}, x_n - 1, x_{n+1}, ..., x_{q_n}]$ is an integer optimal solution to problem $\mathcal{P}^{\beta_1}$,
- $\mathcal{Z}^{\beta_1} = \mathcal{Z}^{\beta_2} - 1$,
- Constraint (18) related to problem $\mathcal{P}^{\beta_1}$ is saturated for the optimal solution.
- All the integer optimal solutions of $\mathcal{P}^{\beta_1}$ are derived from the optimal solutions of $\mathcal{P}^{\beta_2}$ in which $x_n > 0$ by substracting 1 from x_n.

(ii) If, for any $X \in \mathcal{S}^{\beta_2}$, $x_n=0$, then $\mathcal{S}^{\beta_2}$ is the set of integer optimal solutions to problem $\mathcal{P}^{\beta_1}$.

Proof:

a. Assume that there exists $X \in \mathcal{S}^{\beta_2}$ such that $x_n > 0$. Since constraint (18) is saturated for the optimal solutions to problem $\mathcal{P}^{\beta_2}$, we have:

$$\sum_{k=1}^{q_n} x_k . y_{k,1} = s_1 + \beta_2$$

Thus, taking into account the order defined in subsection 3.1 for the n first rows of Y_n, we can see that $y_{n,1} = n$ and:

$$\sum_{k=1}^{q_n} x_k . y_{k,1} - y_{n,1} = s_1 + \beta_2 - n$$

This can be rewritten as:

$$\sum_{k=1/k \neq n}^{q_n} x_k . y_{k,1} + y_{n,1}.(x_n - 1) = s_1 + \beta_2 - n = s_1 + \beta_1$$

Since $y_{n,i} = 0$ for $i = 2, 3, \ldots, n$, the other equality constraints of problem $\mathcal{P}^{\beta_2}$ still hold for the solution $[x_1, x_2, \ldots, x_{n-1}, x_n - 1, x_{n+1}, \ldots, x_{q_n}]$. Thus, according to result 4, this solution is an optimal solution to problem $\mathcal{P}^{\beta_1}$ and $\mathcal{Z}^{\beta_1} = \mathcal{Z}^{\beta_2} - 1$.

b. Let us assume that X^* is an optimal integer solution to problem $\mathcal{P}^{\beta_1}$ such that $X^* \notin \mathcal{S}^{\beta_2}$. Since X^* is optimal, we have:

$$\sum_{k=1}^{q_n} x_k^* . y_{k,1} \geq s_1 + \beta_1$$

and:

$$\sum_{k=1}^{q_n} x_k^* . y_{k,i} = s_i \text{ pour } i = 2, 3, \ldots, n.$$

Thus:

$$\sum_{i=1}^{n} \left[i \left(\sum_{k=1}^{q_n} x_k^* . y_{k,i} \right) \right] \geq \sum_{i=1}^{n} i.s_i + \beta_1$$

or:

$$\sum_{k=1}^{q_n} \left[x_k^* . \left(\sum_{i=1}^{n} i.y_{k,i} \right) \right] \geq \sum_{i=1}^{n} i.s_i + \beta_1$$

Since $\sum_{i=1}^{n} i.y_{k,i} = n$ this inequality can be rewritten as:

$$n\sum_{k=1}^{q_n} x_k^* \geq \sum_{i=1}^{n} i.s_i + \beta_2 - (\beta_2 - \beta_1)$$

But β_2-β_1=n. Thus, this inequality can be rewritten as:

$$n\sum_{k=1}^{q_n} x_k^* \geq \sum_{i=1}^{n} i.s_i + \beta_2 - n$$

or:

$$\sum_{k=1}^{q_n} x_k^* \geq \frac{\sum_{i=1}^{n} i.s_i + \beta_2}{n} - 1$$

Finally:

$$\mathcal{Z}^{\beta_1} \geq \mathcal{Z}^{\beta_2} - 1$$

Since $\mathcal{C}^{\beta_1} \supset \mathcal{C}^{\beta_2}$, we also have $\mathcal{Z}^{\beta_1} \leq \mathcal{Z}^{\beta_2}$.

These results mean that the optimal solution to $\mathcal{P}^{\beta_1}$ leads to the use of **at most** one stored bar less than the optimal solution to $\mathcal{P}^{\beta_2}$.

Both solutions being integer, we can claim that the optimal solution of $\mathcal{P}^{\beta_1}$ leads to the use of either 0 or 1 stored bar less than $\mathcal{P}^{\beta_2}$. Taking into account the fact that the optimal solution to $\mathcal{P}^{\beta_2}$ saturates the first constraint and that this solution is integer, $s_1+\beta_2$ is integer. Keeping in mind that β_2-β_1=n, we see that $s_1+\beta_1 = s_1+\beta_2$-n, and thus $s_1+\beta_1$ is integer. Since only the first constraint of problem $\mathcal{P}^{\beta_1}$ is an inequality, an optimal solution to this problem, if better than an optimal solution to $\mathcal{P}^{\beta_2}$, can only be derived from an optimal solution to $\mathcal{P}^{\beta_2}$ by reducing by one the pattern which corresponds to the cutting out of a stored bar into bars of length Δ. This is only possible if at least one such a pattern is included in one of the integer optimal solutions, i.e. if $x_n^* > 0$ for at least one $X \in \mathcal{S}^{\beta_2}$.

QED

Result 7

We consider the problem $\mathcal{P}^{\beta}$ and we assume that constraint (18) is saturated for an optimal solution and that at least one optimal solution is integer. We denote by $\mathcal{S}^{\beta}$

the set of integer optimal solutions to this problem. Let $X^* = [x_1^*, x_2^*, ..., x_{q_n}^*] \in \mathcal{S}^\beta$ an integer solution such that:

$$x_n^* = \max x_n, \quad X \in \mathcal{S}^\beta$$

Then the solution $X^{**} = [x_1^*, x_2^*, ..., x_{n-1}^*, x_n^* - \eta, x_{n+1}^*, ..., x_{q_n}^*]$, where $\eta = \min[\left\lfloor \beta/n \right\rfloor, x_n^*]$, is an optimal integer solution to problem $\mathcal{P}$, and the corresponding value of the objective function is $\mathcal{Z}^\beta - \eta$. All the integer optimal solutions of $\mathcal{P}$ are obtained by subtracting η to the n-th component of the integer optimal solutions of $\mathcal{P}^\beta$ in which $x_n \geq \eta$.

Proof:

We consider the following sequence of problems: $\mathcal{P}^\beta$, $\mathcal{P}^{\beta-n}$, $\mathcal{P}^{\beta-2n}$, ..., $\mathcal{P}^{\beta-b.n}$. By applying result 6 (i) recursively, we see that X^{**} is an optimal integer solution to problem $\mathcal{P}$ and that the value of the objective function for this solution is $\mathcal{Z}^\beta - \eta$.

QED

5. COMPUTATION OF AN INTEGER NEAR-OPTIMAL SOLUTION

We first define $\beta = \left[s_1 - \sum_{i=2}^{n-1} (i-1).x_i \right]^+$. By solving problem Z^β:

(Z^β) maximise x_n

such that:

$$\sum_{k=1}^{q_n} x_k.y_{k,1} = s_1 + \beta$$

and:

$$\sum_{k=1}^{q_n} x_k.y_{k,i} = s_i, \quad i = 2, 3, ..., n.$$

we obtain a solution X^R which may be not integer. The goal is to find $X^* \in \mathcal{S}^\beta$, integer solution to problem Z^β. This solution is such that $x_n^* = \max x_n$, $X \in \mathcal{S}^\beta$. Assume that this solution is known. From this solution, we can derive the integer optimal solution X^{**} of $\mathcal{P}$ (see result 7): $X^{**} = [x_1^*, ..., x_{n-1}^*, x_n^* - \eta, x_{n+1}^*, ..., x_{q_n}^*]$, where $\eta = \min(\left\lfloor \frac{\beta}{n} \right\rfloor, x_n^*)$. The value of the criterion related to this solution is $\mathcal{Z}=\mathcal{Z}^{\beta}-\eta$. According to result 2, X^{**} is also an integer optimal solution to problem $\mathcal{Q}$.

As we can see, the remaining problem is to find X^*. Computing this solution is, *a priori* , a NP-hard problem. Thus, we decided to derive a solution X^P from X^R , this solution being close to X^* , and to use this solution instead of X^* to compute X^{**}. Note that if the n-th component of X^P is the integer part of X^R, then $X^*=X^P$, and X^{**} is the integer optimal solution of $\mathcal{P}$: this was always the case in the numerous exemples which were computed to illustrate this approach. ***Algorithm 2*** summarises the sequence of computations which leads to X^{**}.

Algorithm 2 .

1. Computation of $\beta = \left[s_1 - \sum_{i=2}^{n-1} (i-1).x_i \right]^+$.
2. Computation of X^R, solution to the Linear Programming problem Z^{β}.
3. Computation of X^P.

   3.1. Computation of $X^P=\lfloor X^R \rfloor$, vector composed with the integer parts of X^R.

   3.2. Computation of $Y= X^P-X^R$, vector composed with the decimal parts of X^P. Let $W=[w_1, w_2, ..., w_n, ..., w_{q_n}]$.

   3.3. For i=1 to n do $v_i = \sum_{k=1}^{q_n} w_k.y_{k,i}$, and let $V=\left[v_1, v_2, ..., v_n, ..., v_{q_n}\right]$.

   Note that the components of V *are integer and that* $\sum_{k=1}^{q_n} k.v_k = L.n$, *where* L *is integer.*

   3.4. Consider the components of V as the ordered bars, and assign these bars to the stored bars of length n.Δ by applying the FFD algorithm. Let U be the vector whose integer components $u_1, u_2, ..., u_n, ...,u_{q_n}$ are the numbers of bars cut following patterns 1, 2, ...,n, ...q_n given by matrix Y_n to meet the requirements represented by matrix V.

3.5. Compute $X^* = X^P + U$.

4. Compute $\eta = \min(\left\lfloor \frac{\beta}{n} \right\rfloor, x^*_n)$.

5. $X^{**} = [x^*_1, \ldots, x^*_{n-1}, x^*_n - \eta, x^*_{n+1}, \ldots, x^*_{q_n}]$ is the integer optimal solution to problems $\mathcal{P}_1$ and Q.

6. Compute $\mathcal{K}_1$, criterion related to X^{**}.

6. ILLUSTRATIVE EXAMPLES

For lack of place, we restrict ourselves to the case where n=8. In this case, $q_8=22$, and Y_8 is given in the following table.

Table 1: *Matrix Y_8*

Matrix								Pattern Number
0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	1	0	2
2	0	0	0	0	1	0	0	3
3	0	0	0	1	0	0	0	4
4	0	0	1	0	0	0	0	5
5	0	1	0	0	0	0	0	6
6	1	0	0	0	0	0	0	7
8	0	0	0	0	0	0	0	8
1	0	1	1	0	0	0	0	9
0	2	0	1	0	0	0	0	10
2	1	0	1	0	0	0	0	11
0	0	1	0	1	0	0	0	12
0	1	2	0	0	0	0	0	13
2	0	2	0	0	0	0	0	14
1	2	1	0	0	0	0	0	15
3	1	1	0	0	0	0	0	16
1	1	0	0	1	0	0	0	17
0	4	0	0	0	0	0	0	18
2	3	0	0	0	0	0	0	19
4	2	0	0	0	0	0	0	20
0	1	0	0	0	1	0	0	21
0	0	0	2	0	0	0	0	22

In table 2, we give seven examples of customers' requirements. Column i contains the number of bars of length i.Δ required by the customers.

Table 3 provides the value of β for each one of the example introduced in table 2. Remember that such a value represents the number of bars of length Δ produced in excess if we meet a customer requirement following only the eight first patterns.

In table 4 we provide, for each example, the solution X^R of problem Z^β (see step 2 of algorithm 2)..

Table 2 : *Customers' requirements*

EXAMPLE NUMBER ⇒ LENGHT OF BARS ⇓	1	2	3	4	5	6	7
1	1	1	100	2	22	2	80
2	5	1	1	51	11	14	2
3	17	41	3	21	3	6	1
4	2	0	5	5	1	3	0
5	14	21	7	3	15	8	5
6	9	0	1	2	8	9	3
7	25	0	5	11	2	5	8
8	1	0	22	0	0	2	1

Table 3 : *Values of β*

EXAMPLE NUMBER ⇒	1	2	3	4	5	6	7
VALUE OF β	0	0	31	0	0	0	34

Table 4 : *Solution to problem* $\mathbf{Z}^{\beta}$

EXAMPLE NUMBER ⇒ PATTERNS ⇓	1	2	3	4	5	6	7
1	1	0	22	0	0	2	1
2	25	0	5	11	2	5	8
3	4.5	0	1	0	8	0	3
4	0	0	3	0	1	0	2
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	22.5	33	106.5	57.5	36.5	22.75	83
9	2	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	14	21	3	3	3	6	1
13	0.5	1	0	9	0	0	0
14	0	9	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	1	0	11	2	2
18	0	0	0	10	0	0.75	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	4.5	0	0	2	0	9	0
22	0	0	2.5	2.5	0.5	1.5	0

The optimal solutions presented in table 4, except the solutions of examples 2 and 7, are not integer. In table 5, we present the vectors V corresponding to the seven examples (see step 3.3 of algorithm 2).

Table 5 : *Vectors v*

MPLE NUMBERS ⇒ LENGHT OF BARS ⇓	1	2	3	4	5	6	7
1	5	0	4	4	4	6	0
2	1	0	0	0	0	3	0
3	1	0	0	0	0	0	0
4	0	0	1	1	1	1	0
5	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0

By applying steps 3.4 and 3.5 of algorithm 2, we obtain the integer near-optimal solutions of the eight examples presented in this section. It turns out that the total length of the bars obtained by breaking down the V vectors presented in table 5 using the FFD algorithm (see section 3.4 of algorithm 2) is equal to 8.L (see the definition of L in step 3.3 of algorithm 2). As a consequence, the integer solutions obtained and presented in table 6 are optimal.

Table 6 : *Final solution*

EXAMPLE NUMBERS ⇒ PATTERNS ⇓	1	2	3	4	5	6	7
1	1	0	22	0	0	2	1
2	25	0	5	11	2	5	8
3	4	0	1	0	8	0	3
4	0	0	3	0	1	0	2
5	0	0	1	0	1	0	0
6	1	0	0	0	0	0	0

7	0	0	0	0	0	1	0
8	22	33	106	57	36	22	83
9	2	0	0	0	0	0	0
10	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0
12	14	21	3	3	3	6	1
13	0	1	0	9	0	0	0
14	0	9	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	1	0	11	2	2
18	0	0	0	10	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	5	0	0	2	0	9	0
22	0	0	2	2	0	1	0

We finally compute the η values (see step 4 of algorithm 2) which are the number of bars cut out following pattern 8 which can be subtracted from the solutions given in table 6 to obtain the solutions of the initial problems. These values are presented in table 7.

Table 7 : *Values of η*

EXAMPLE NUMBERS ⇒	1	2	3	4	5	6	7
VALUES OF THE PARAMETER η	0	0	3	0	0	0	4

7. CONCLUSION

As showed above, it is possible to find an integer near optimal solution to problem Q by solving one linear programming problem (problem Z^{β}). In practice, as shown in the above examples, we never encountered a case where the integer solution is not optimal. At this stage of the research, the problem which consists of proving that the solution obtained is always optimal or not remains open.

The size of the problem increases very fast with regard to n, and a reasonable limit is n=30.

The proposed method is fast. One of the next research axes will consist in selecting promising cutting patterns in order to drastically increase the size of the problems which can be solved. Indeed, this will be done at the expense of the qual.ty of the solution, and the goal will be to make the selection which leads to the best solution. The n first rows of matrix Y_n, as ordered in subsection 3.1., will always belong to the selection since they are the initial basis of the system.

BIBLIOGRAPHY

[1] ANTONIO J., CHU C., SAUER N. et WOLFF P. (1995), "POEM Project and Solution for One-dimensional Multi-criteria Cutting Stock Problems", Operations Research Proceedings (SOR'95), Passeau (AL).

[2] COFFMAN E.G.Jr., GAREY M.R. et JOHNSON D.S. (1984), "Approximation Algorithms for Bin-Packing - An Updated Survey", Bell Laboratories, Murray Hill, New Jersey (US).

[3] DOWSLAND Kathryn A. and DOWSLAND William B. (1992), "Packing Problems", European Journal of Operational Research, vol. 56, n°1, p. 2-14, Swansea (GB).

[4] DYCKHOFF Harold (1981), "A New Linear Programming Approch to the Cutting Stock Problem", Operations Research, vol. 29, p. 1092-1104, Hagen (AL).

[5] DYCKHOFF Harold (1990), "A Typology of Cutting and Packing Problems", European Journal of Operational Research, vol. 44, n°2, p. 145-159, Aachen (AL).

[6] DYCKHOFF Harald et FINKE Ute (1992), "Cutting and Packing in Production and Distribution; A Typology and Bibliography", ed. Physica-verlag, A Springer-Verlag company, Aachen (AL).

[7] GAREY M. R. et JOHNSON D. S. (1981), "Approximation Algorithms for Bin Packing Problems: a Survey", Analysis and Design of Algorithms in Combinatorial

Optimisation, Ausiello, G & Lucertini, M. Eds, Springer, p. 149-172, Murray Hill (US).

[8] GILMORE P. C. et GOMORY R. E. (1961), "A Linear Programming Approach to the Cutting-Stock Problem", Operations Research, vol. 9, New York (US).

[9] JOHNSON David S. (1974), "Fast Algorithms for Bin Packing", Journal of Computer and System Sciences, vol. 8, p. 272-314, Massachusetts (US)

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

RR - 3421