Cátedra Nissan

-PROTHIUS-

Procedures for solving a 1-dimensional cutting problem

Albert Corominas Subias, Joaquín Bautista Valhondo y M. Teresa Fernandez

WP-08/2010
(Rec. DIT 91/03- BCF - 1991)

Departamento de Organización de Empresas

Universidad Politécnica de Cataluña

Publica:

Universitat Politècnica de Catalunya
www.upc.edu

Edita:

Cátedra Nissan
www.nissanchair.com
director@nissanchair.com

PROCEDURES FOR SOLVING A 1-DIMENSIONAL CUTTING PROBLEM

Albert Corominas
Joaquín Bautista
Ma. Teresa Fernandez

Document Intern de Treball
(D.I.T.) 91/03

Barcelona, Abril 1991

Departament d'Organització d'Empreses
Universitat Politècnica de Catalunya

PROCEDURES FOR SOLVING A 1-DIMENSIONAL CUTTING PROBLEM

Albert COROMINAS; Joaquín BAUTISTA; Maria-Teresa FERNANDEZ.
Departament d'Organització d'Empreses/ETSEIB.
Universitat Politècnica de Catalunya.
Av. Diagonal 647.
08028 Barcelona, Spain.

Abstract:

The different models of a particular article for widespread consumption are characterized by the material used and its length.

Each material is acquired in jumbos of a particular length. By cutting them, we must obtain for each production period at least as many items for each model as are required to cover consumption for the following period; anything above this value will be used in later periods.

In order to minimize the cost associated with this process (that of cutting plus the inventory holding cost), procedures combining ILP with heuristics based on dynamic programming have been designed and applied.

Keywords: Cutting problem, integer linear programming, dynamic programming, heuristics

1.- Introduction

This work describes the procedures tested and adopted in solving a one-dimensional cutting problem which occurs in industry; a description of the system in which these procedures are inserted can be found in Corominas, Bautista and Companys (1989).

One of the first steps in certain production process consists in cutting jumbos to obtain items of different lengths. There are several types of materials and the jumbos of each of them have the same length; the items are characterized by the material and by the length (each pair material/length will also be called the model or product).

The production of units of each model, together with the stock available, must be sufficient to cover the day's consumption and to provide a certain safety stock. The items not consumed on the same day may be used on later days, since although the daily consumption is not constant it is fairly stable. A certain production in excess of the necessary minimum is therefore admissible, though it should not, of course, exceed certain bounds. Indeed, the production plus the stock should be within bounds which may be calculated as the product of coefficients which are greater than one, multiplied by the daily consumption. Storing the excess production of items for a certain number of days has a cost (althougt far lower than that of wastage), but the possibility of excess production permits a general reduction in wastage.

We must therefore establish a cutting pattern for each jumbo which minimizes the overall cost of cutting and stocks. This leads to the problem of establishing the optimum cutting plan for a given number of jumbos of a given material.

2.- A brief review of the literature on the cutting problem

The problem presented in the previous section may be classified as a one-dimensional cutting problem.

It is not our intention here to provide a complete list of references - for this see Sweeney (1989) -. Here we only wish to mention some works directly related to the present one.

Hinxman (1980) is a very complete study. Dyckhoff and Gehring (1988), Dyckhoff, Kruse, Abel and Gal (1988), Dyckhoff, Finke and Kruse (1988) and Dyckhoff (1990) are also very useful; the first of these references includes a description of the Dyckhoff model, based on the "one cut" technology, which has to a certain extent inspired the heuristics based on the dynamic programming described below.

The model which is normally proposed in the literature for one-dimensional cutting models is formulated with variables which express the number of times that each cutting pattern is applied; The Dyckhoff model, which is advantageous in some cases, is to a certain extent a version of the above. This approach involves, of course, two problems: the very high number of cutting patterns (and thus of variables) and the fact that it requires an integral solution. The first of these difficulties may be overcomed with the procedure of column generation devised by Gilmore and Gomory - Gilmore and Gomory(1961) and Gilmore and Gomory (1963) -, in which the resolution in each iteration of a knapsack problem allows us to obtain a new pattern; but the information published on the convergence of this procedure is not very precise; it is also very strictly dependent on a particular structure of the model and is therefore fragile to the presence of additional conditions. The resolution of the linear program also produces results which are very rarely integral; the most immediate answer to this is rounding off, but it is not always easy to assess the repercussions of this course of action.

Another paper, which refers exclusively to the one-dimensional problem, is Costa (1982), whose main interest is its formulation, although it diverges from it when it proposes methods of resolution. This formulation differs from that referred to in the previous paragraph, since the variables correspond to the number of units of each product which are obtained from each unit of raw material. Escudero (1976) also presents a similar approach. This approach has the drawbacks that in general it is not possible to obtain satisfactory solutions by rounding off those obtained with linear programming, and that the bounds provided by the latter are of little use, at least in the first vertices of the tree in a branch and bound algorithm.

Johnson (1989) includes a comparison of the two formulations laid out. Johnson quickly rejects that known as "naive", because "the variables do not represent important decisions. For example, deciding that a given roll will not have any of particular length cut from it has very little effect on the linear programming solution because that length can always be cut from some other roll. In a way, there is just too much symmetry in the formulation". He then goes on to explain the column generation procedure of Gilmore and Gomory and compares it with the naive approach through an example. On the whole, Johnson's argument is not very convincing, although it is true that the first formulation is excessively symmetric.

In a work published after we had developed, programmed and tested our own heuristics - Haessler (1988) -, the author is decidedly in favor of this type of procedure and critical of those who are uncompromisingly in favor of strict optimization procedures. We also had the satisfaction of discovering the coincidence of some ideas which we had incorporated into our heuristics with those laid out by Haessler.

3.- General approach of the resolution procedure

The problem may be formulated as a non-linear integer program, which is non-linear because of the criteria of optimization adopted (the minimization of the ratio between the cost and the value of the product obtained). For a single type of material we can formulate a linear model which provides a good approximation, but in the case of several materials this is not possible, and the cutting of each material cannot be optimized separately because there are common restrictions (the time available for the cutting process, for example).

Now, the number of jumbos of each type of material to be cut has upper and lower bounds which are easy to determine a priori. If we establish which cutting patterns are to be used for each possible value of the number of jumbos of each material to be cut, we can calculate the cost, the value of the production and the running time for each of these partial solutions. With this information on the partial solutions, with a partial solution for each material we can form each general solution, check whether it is feasible and if so, determine its relative cost. Thus, with a simple enumeration or by dynamic programming, we obtain the best general solution.

We therefore require a procedure for calculating partial solutions.

In the application which led to the development of the procedure described, the length of the items was small in relation to that of the jumbos - a soft problem, according to the classification included in Goulimis (1990) -, so the number of cutting patterns was very high. On the other hand, the number of types of materials and types of items, as well as the number of jumbos to be cut in a day, were low.

For all these reasons a model whose most important variables, x(j,i,), correspond to the number of items of each type to be cut from the jumbo j was very compact. Note that here the index j is a simple serial number, with no other meaning (for a material all jumbos have the same length); this order is arbitrary and, as we have said, will give the formulation an undesirable symmetry, which may be avoided by means of a certain refinement in the modelling which will be laid out below.

The resolution of the model may be carried out by integer linear programming, with heuristic algorithms or with a combination of both types of procedures.

4.- Linear models

4.1.- Basic model

According to the consumption and stock of items available, for each model we can establish every day upper and lower bounds of the number of items, $n(i)$, to be obtained in the cutting of the jumbos:

$$l(i) \le n(i) \le u(i)$$

If we give the name $c(i)$ to the expected daily consumption and $s(i)$ to the available stock, these bounds are calculated with the expressions:

$$l(i) = \alpha c(i) - s(i)$$

$$u(i) = \beta c(i) - s(i)$$

Where α and β are parameters to be determined by the user (of course, with the conditions $\alpha \le \beta$ and $\alpha, \beta \ge 1$).

We can therefore formulate an integer linear program as follows (where all the variables [$n(i)$, $x(j,i)$, $r(j)$] are non-negative integers).

$$[\text{MIN}]\ z = \{\bar{C} \sum_{j=1}^{J} r(j) + \sum_{i=1}^{I} h[i, n(i)-l(i)]\} \qquad (1.1)$$

$$n(i) = \sum_{j=1}^{J} x(j,i) \qquad i=1,\ldots,I \qquad (1.2)$$

$$\sum_{i=1}^{J} a(i)x(j,i) + r(j) = A \qquad j=1,\ldots,J \qquad (1.3)$$

$$l(i) \leq n(i) \leq u(i) \qquad i=1,\ldots,I \qquad (1.4)$$

$$r(j) \leq \min_i a(i) - 1 \qquad j=1,\ldots,J \qquad (1.5)$$

$$x(j,i) \leq INT[A/a(i)] = q(i) \qquad j=1,\ldots,J \quad i=1,\ldots,I \qquad (1.6)$$

The notation which has not yet been made explicit is:

C	cost of the material per unit of length
n(i)	number of items of length a(i) produced
r(j)	length of wastage of the jumbo j
A	length of the jumbo
INT(x)	greatest integer ≤ x
h[i,n(i)-l(i)]	cost associated with the production in excess of the minimum l(i)
q(i)	maximum number of items of length a(i) which may be obtained from a jumbo

The cost associated with the process is that of the off-cuts plus that of the non-essential stock. The cost of the stock is expressed in (1.1) as a sum of functions (one for each kind of product) corresponding to the costs of storing a number of items greater than the strictly essential minimum, l(i). The nature of these functions is specified below.

Constraints (1.2) link the variables x(j,i) with the total number of items of each type; constraints (1.3) express that the sum of the lengths of the items cut plus that of the wastage is equal to the length of the jumbo; constraints (1.4) impose the upper and lower bounds of the total number of items of each type; constraints (1.5) establish the upper bounds of the length of the wastage of each jumbo, lower than the minimum length of the items – only <u>complete</u> cuts or cutting patterns are used, according to the terminology used in Stadler (1990) –; finally, constraints

(1.6) make explicit the upper bounds of the variables $x(j,i)$, which are implicit in constraints (1.3).

As we have said above, the processing of the cost of the stocks forces us to introduce more variables and constraints due to their non-linear nature. The processing will use the value $\tau(i)$ (smallest integer $\geq$ the quotient $INT[u(i)-l(i)]/c(i)$, which corresponds to the number of days needed to consume the production in excess of the essential needs); it may be written:

$$n(i) - l(i) = \sum_{k=1}^{\tau(i)} v(i,k)$$

where the variables $v(i,k)$ represent the consumption for day k, counted from the present date; of course:

$$0 \leq v(i,k) \leq c(i) \quad \forall i,k$$

Therefore:

$$h[i,n(i) - l(i)] = \theta C a(i) \sum_{k=1}^{\tau(i)} k v(i,k)$$

where θ is the daily holding cost of a item of cost 1.

If $l(i)$ is negative (this corresponds to an excess of stock, even without production of type i items), the approach is also valid, with slight adjustments. Here we will suppose that $l(i) \geq 0$.

In accordance with all these considerations, the basic model is finally as shown:

$$[MIN]\ z = C\ [\ \sum_{j=1}^{J} r(t) + \theta \sum_{i=1}^{I} a(i) \sum_{k=1}^{\tau(i)} k v(i,k)] \qquad (2.1)$$

$$n(i) = \sum_{j=1}^{J} x(j,i) \qquad i=1,\ldots,I \quad (2.2)$$

$$n(i) - l(i) = \sum_{k=1}^{\tau(i)} v(i,k) \qquad i=1,\ldots,I \quad (2.3)$$

$$\sum_{i=1}^{I} a(i)x(j,i) + r(j) = A \qquad j=1,\ldots,J \quad (2.4)$$

$$l(i) \leq n(i) \leq u(i) \qquad i=1,\ldots,I \quad (2.5)$$

$$r(j) \leq \min_{i} a(i) - 1 = a(i') - 1 \qquad j=1,\ldots,J \quad (2.6)$$

$$x(j,i) \leq q(i) \qquad j=1,\ldots,J \quad i=1,\ldots,I \quad (2.7)$$

$$0 \leq v(i,k) \leq \min[c(i),u(i)-l(i)-(k-1)c(i)]$$

$$i=1,\ldots,I \quad 0 < k \leq \tau(i) \qquad (2.8)$$

4.2.- Improved model

We have considered two types of improvement in the model:

a.- Improvements in the formulation of the constraints.

Given A and a(i) (i=1,...,I), we can calculate the maximum length achievable (A'). The problem may be approached thus:

$$[MAX]\ A' = \sum_{i=1}^{I} a(i)x(i)$$

$$\sum_{i=1}^{I} a(i)x(i) \leq A$$

with the x(i) integral and such that $0 \leq x(i) \leq q(i)$.

This is thus a knapsack problem (KP), and more specifically the particular case in which the coefficients of the objective function are the same as those of the constraint, which is known as the "subset sum problem".

In order to solve it we can use, for example, a dynamic programming procedure.

Once we have determined A′, it can replace A in the right-hand side of constraints (2.4).

But we can still improve the formulation of these constraints by adopting the greatest common divisor of the a(i), which will also be a divisor of A′, as the unit for measuring the lengths involved in each problem.

The tests confirmed the effectiveness of these simple improvements.

Another, less intuitive, improvement, but one which has also been compared experimentally, is that described below.

The fact that the length of the wastage is always lower than that of the product of least length (of index i′), determines a lower bound of the number of items obtained from a jumbo (the greatest number of units obtainable from the model of greatest length - of index i" -). An upper bound is obviously q(i′). Therefore:

$$q(i'') \le \sum_{i=1}^{\bar{I}} x(j,i) \le q(i')$$

and if the sum is called g(j):

$$q(i'') \le g(j) \le q(i')$$

or, making $g(j) = q(i'') + g'(j)$:

$$0 \leq g'(j) \leq q(i') - q(i")$$

Now, as i' is the index of the product of minimum length, it can be written:

$$a(i) = a(i') + a'(i) \text{ con } a'(i) \geq 0 \quad \forall i$$

and thus:

$$\Sigma a(i)x(j,i) = \Sigma[a(i')+a'(i)]x(j,i) = a(i')\Sigma x(j,i)+\Sigma a'(i)x(j,i) =$$
$$= a(i')g(j)+\Sigma a'(i)x(j,i) = a(i')[g'(j)+q(i")]+\Sigma a'(i)x(j,i) =$$
$$= a(i')g'(j)+\Sigma a'(i)x(i,j)+a(i')q(i")$$

and therefore the constraints may be formulated as follows:

$$a(i')g'(j) + \Sigma a'(i)x(i,j) + r(j) = A' - a(i')q(i")$$

b.- Discriminating between equivalent solutions.

The fact that the jumbos are numbered gives rise to different solutions, which are equivalent with respect to the criteria incorporated in the model. From this point of view, it does not matter if we cut jumbo j with a given pattern and j' with another, or if we permutate the cutting patterns of both. This means that a branch and bound algorithm explores different equivalent branches which, however, correspond to formally different groups of solutions.

It is therefore advisable to discriminate between these equivalent solutions. A simple way of doing this (with which J-1 additional constraints are sufficient), which gave very good results in the tests carried out, consists in considering each group of variables x(j,i) (i=1,...,I) as a vector and accepting only solutions in which these vectors are lexicographically ordered. Since $q(i)$, as we have said, represents the upper bounds of $x(j,i)$, this condition may be imposed with the constraints:

$$\sum_{i=1}^{I} x(j,i) \prod_{l=i+1}^{I} [q(l)+1] \geq \sum_{i=1}^{I} x(j+1,i) \prod_{l=i+1}^{I} [q(l)+1] \qquad j=1,...,J-1$$

The coefficients of these constraints depend on the order established for the different products. It is therefore best to arrange them in increasing order, since this gives a monotonously decreasing sequence of the $q(i)$, and therefore coefficients with lower values.

In addition, the incorporation of these constraints allows us to obtain solutions ordered in such a way that they can correspond to the real sequence of the cutting process of the material (the jumbos with the same cutting pattern are consecutive).

In accordance with all the considerations made, for the optimization of the total cost with J jumbos of a given material, we can propose the integer linear program that follows, with the same notation as has been used throughout the text and with the variables integral and non-negative. As can be seen, this is a compact model of a fairly small size (the number of non-bound constraints is equal to 3J+2I-1):

$$[MIN]\ z = \sum_{j=1}^{J} r(j) + \theta \sum_{i=1}^{I} a(i) \sum_{k=1}^{\tau(i)} kv(i,k) + CJd \qquad (3.1)$$

$$n(i) = \sum_{j} x(j,i) \qquad i=1,...,I \qquad (3.2)$$

$$a(i')g'(j) + \sum_i a'(i)x(i,j) + r(j) = A' - a(i')q(i'') \qquad j=1,\dots,J \qquad (3.3)$$

$$\sum_i x(j,i) - g'(j) = q(i'') \qquad j=1,\dots,J \qquad (3.4)$$

$$n(i) - l(i) = \sum_k v(i,k) \qquad i=1,\dots,I \qquad (3.5)$$

$$\sum_i x(j,i) \prod_{l=i+1}^{I} [q(l)+1] \geq \sum_i x(j+1,i) \prod_{l=i+1}^{I} [q(l)+1] \qquad j=1,\dots,J-1 \qquad (3.6)$$

$$l(i) \leq n(i) \leq u(i) \qquad i=1,\dots,I \qquad (3.7)$$

$$r(j) \leq \min_i a(i) - 1 = a(i') - 1 \qquad j=1,\dots,J \qquad (3.8)$$

$$x(j,i) \leq q(i) \qquad j=1,\dots,J \quad i=1,\dots,I \qquad (3.9)$$

$$0 \leq g'(j) \leq q(i') - q(i'') \qquad j=1,\dots,J \qquad (3.10)$$

$$0 \leq v(i,k) \leq \min[c(i), u(i)-l(i)-(k-1)c(i)] \qquad i=1,\dots,I \quad 0 < k \leq \tau(i) \qquad (3.11)$$

4.3.- Model for optimization of the relative cost

As we have said above, for the case of a single material we can propose an ILP for an approximate minimization of the relation between the cost and the value of the product. The approximation consists in minimizing the relation between the cost and the number of jumbos (situated a priori between m and M):

$$[\mathrm{MIN}]\ \delta \qquad (4.1)$$

$$\sum_{i=1}^{I} a(i)x(j,i) + r(j) = A \qquad j=1,\dots,m \qquad (4.2)$$

$$\sum_{i=1}^{I} a(i)x(j,i) + r(j) = Ay(j) \qquad j=m+1,\dots,M \qquad (4.3)$$

$$l(i) \le n(i) \le u(i) \qquad i=1,\dots,I \qquad (4.4)$$

$$r(j) \le \min_{i} a(i) - 1 \qquad j=1,\dots,M \qquad (4.5)$$

$$x(j,i) \le q(i) \qquad j=1,\dots,M \quad i=1,\dots,I \qquad (4.6)$$

$$\sigma = C \sum_{j} r(j) + \sum_{i} h[i, n(i)-l(i)] \qquad (4.7)$$

$$\frac{1}{j-1}\sigma \le \delta + By(j) \qquad j=m+1,\dots,M \qquad (4.8)$$

$$\frac{\sigma}{M} \le \delta \qquad (4.9)$$

$$y(j+1) \le y(j) \qquad j=m+1,\dots, M-1 \qquad (4.10)$$

All the variables are non-negative integers, with the exception of δ and σ, and the $y(t)\in\{0,1\}$.

The aim is to minimize the variable δ which represents the cost of the wastage and of the excess stock in relation to the cost of the material corresponding to the jumbos cut.

The variables $y(t)$ are associated to each of the jumbos above the minimum m; the variable has a value of 1 if the jumbo is cut and 0 if it is not.

Constraints (4.2) and (4.3) correspond respectively to the jumbos which will certainly have to be cut and the ones for which the operation must be determined by the program.

Constraints (4.4), (4.5) and (4.6) are analogous to those which appear in the models mentioned above.

Constraints (4.7) define the cost which constitudes the numerator of the quotient which determines δ.

Constraints (4.8) and (4.9), which should be considered together taking into account constraints (4.10), link the value of δ to that of σ through the y(t). Indeed, constraints (4.10) prevent the cutting of a jumbo if the previous one is not cut, so the acceptable values for the components of vector Y are only of the type "sequence of ones, perhaps empty, followed by a sequence of zeros, perhaps empty". When y(t) = 1, the corresponding constraints (4.8) are inoperative, because it is always fulfilled that:

$$\frac{1}{t-1}\sigma \leq \delta + B \qquad t=m+1,...,M$$

if B is a sufficiently large value.

4.4.- Extensions

The functions h [i, n(i) - l(i)] may have a different form from that supposed up to now. This responds to the hypothesis of constant daily consumption and indefinite conservation of the product obtained, without loss or deterioration. But there is no difficulty in adapting the functions h to some other assumptions.

For example, if we wish to produce a given quantity of items of a certain type for immediate consumption which will not be prolonged afterwards, the cost of the excess production, with relation to l(i), must be equal to that of a wastage of the same length.

Or if for any reason (for example if it is wished to bring forward the production because a high consumption is foreseen for the following days or in order to have the possibility of combining more values of a(i) and thus producing less wastage) it is wished to produce a certain quantity of items of a product which will not begin to be consumed for a certain number of days afterwards, it will be sufficient to add this number to the index k in the calculation of the functions h.

4.5.- Resolution by means of integer linear programming: computational experience

Several tests were carried out with the models described.

As for the models with a fixed number of jumbos, some of the tests were carried out with the algorithm of Gomory, with very bad results (program not feasible in cases in which solutions existed) because of numerical difficulties on adding the cutting plans (except in problems of a very small size).

The application of the package MILP88/MILP87 was also tested; the times were short in most cases, but increased rapidly whith J and were sometimes excessively long for a daily industrial application.

As for the relative cost model, the tests carried out were not sufficiently numerous to reach definitive conclusions. It was observed, however, that the time necessary for obtaining the solution was usually lower than that of resolving separately the M-m+1 models corresponding to the different possible numbers of jumbos cut, but greater than that needed to resolve them successively, using in each one the optimum value of the previous one. However, for some sets of data there appeared severe numerical problems which cause anomalies in the solution.

5.- Heuristic algorithms

It is quite easy to obtain a solution for J jumbos from the solution for J-1 jumbos. It is sufficient to juxtapose on the solution of J-1 jumbos an additional jumbo, whose cutting patterns can be obtained through heuristics. This type of algorithm has been called incremental heuristics.

Thus, from the optimum solution obtained with the ILP for J=m, we can obtain a generally satisfactory solution for J=m+1,...,M. Note also that if the ILP is initialized with the value corresponding to the solution achieved through the incremental heuristics, we can find out whether or not it is optimum, and if not, a different solution will be reached.

However, there is still the difficulty of finding a solution for J=m. This led to the design of a general heuristics, which has elements in common with the incremental heuristics mentioned above.

Both types of heuristics are described in the following sections.

5.1.- Incremental heuristics

As we have said, given a solution for J-1 jumbos this type of heuristics constructs one for J jumbos.

Incremental heuristics is based on quite an obvious observation: If a good solution is available for J-1 jumbos, another for J jumbos can be obtained easily by maintaining the cutting pattern corresponding to the solution available for the first J-1 jumbos and determining a good cutting pattern for the additional jumbo J.

It is therefore a case of resolving a problem for a single jumbo, with the values of the upper and lower bounds of production of each model modified in order to take into account the production corresponding to the first J-1 jumbos, $n'(i)$, which will also have to intervene in the calculation of the cost of stock. The lower bounds will be nil, because the solution for J-1 is possible, and the upper bounds will have a value which will be called $u'(i)$. It is thus a case of optimizing the cutting of jumbo J, having established the way of performing the previous J-1 jumbos:

$$[MIN]\ z = r + \sum_{i=1}^{I} h\,[i,\ x(i)+n'(i)-l(i)]$$

$$\sum_{i=1}^{I} a(i)x(i) + r = A'$$

with the $x(i)$ integer and such that $0 \le x(i) \le u'(i)$.

Which is a mathematical program equivalent to the following:

$$[MIN]\ z = A' - \sum_{i=1}^{I} a(i)x(i) + \sum_{i=1}^{I} h_i\,[x(i)+n'(i)-l(i)]$$

$$\sum_{i=1}^{I} a(i)x(i) \le A'$$

Or, making $a(i)x(i)+ h[i, x(i)+n'(i)-l(i)] = h'[i, x(i)]$, to:

$$[MAX]\ z = -A' + \sum_{i=1}^{I} h'[i,x(i)]$$

$$\sum_{i=1}^{I} a(i)x(i) \le A'$$

This is similar to the knapsack problem, but with a non-linear objective function, which may be solved with a dynamic programming procedure which is also similar to that described for the KP:

$$f^{*}_{i+1}(r) = \max_{0 \le x(i) \le \min[u'(i),q(i)]} \{ f^{*}_{i}[r+a(i)x(i)]+h'[x(i)]\}$$

with $f(A')=0$ and $f(r)=-\infty$ (for $r \ne A$).

5.2.- General heuristics

In the tests carried out, once a first possible solution had been found by integer linear programming, the incremental heuristics gave very good results and fulfilled the objective of finding optimum or satisfactory solutions for any number of jumbos in relatively short calculating times.

It is still, however, necessary to obtain a first possible solution for the minimum number of jumbos, $\bar{m}$. If this number is relatively high, the calculating time of the ILP algorithm may be long.

We therefore needed a procedure for obtaining satisfactory possible solutions in short times without using any other solutions as a basis.

The algorithm designed for this purpose, and in this context known as <u>general heuristics</u>, has points of contact with the incremental heuristics described in the previous section. Having established J number of jumbos, the general heuristics consists initially in J iterations in each of which is applied, basically, the incremental heuristics, with the peculiarities described below.

The main difference between the problem resolved by the incremental heuristics and that correspondind to the general heuristics is that in the first case we already know a possible solution which is merely completed with one more jumbo, whereas the general heuristics must construct a possible solution (basically it is a question of obtaining at least l(i) items of each type).

In order to force into the solution the items which are missing in order to reach the minimum, the recursion equation of the dynamic program is modified in the following way:

$$f^{*}_{i+1}(r) = \max_{0 \le x(i) \le \min[u'(i),q(i)]} \{ f^{*}_{i}[r+a(i)x(i)]+h'[x(i)]+h''[x(i)] \}$$

where $h''[x(i)]=0$ when $l'(i)=0$ and when $l'(i)>0$:

$$h''[x(i)] = \sum_{k=1}^{\max[l'(i),x(i)]} G[l'(i)-k+1]^{\mu} \, a(i)^{\mu'}$$

where G is a large value (in order to give a greater value in the objective function to the items which are essential in order to reach the minimum than to the others) and μ and μ' are parameters (which may be between 0 and 1) with which a greater or lesser importance may be given to the "deficit" of items of each type or to their length.

In fact, the presence of these parameters defines a family of heuristics and gives many possibilities, according to the time and calculating power available; for example, we can systematically try different values of the pair (μ,μ') and retain the best solution obtained, and even consider the value obtained with the heuristics as a fuction of the two parameters and apply a direct method of optimization.

If we can only work with a single value of the parameters, a reasonable choice is $\mu=1$ and $\mu'=0$.

This dynamic programming pattern is thus applied to jumbos $1,2,...,J$.

In each of them it is calculated whether or not it is possible to obtain a solution, and if so, whether or not there is a <u>margin</u>; i.e. whether the material corresponding to the jumbos as yet not calculated is sufficient for cutting the essential items not assigned to previous jumbos, and if so, whether it is only for the essential ones or leaves a margin for some non-essential ones.

If there is a margin, once the dynamic programming has been applied with the recursion equation described above, for the jumbo in question the solution adopted is that which optimizes the part of the objective function which is not affected by the factor G (indeed, that which optimizes the cost of the wastage plus that of the stock, and which, with the values which may occur in practice, is also that which optimizes the wastage); the underlying reasoning is that what is decided for one jumbo cannot be recovered in later jumbos, and that any decision involving wastage and stock costs should be postponed as far as possible. It will be said that in this case criterion 1 is applied.

But if there is no margin, it is advisable to cut the essential items, and the optimization must therefore refer to the whole objective function, including, consequently, the part affected by factor G. It will then be said that criterion 2 is applied.

In the application of the algorithm there will therefore be a sequence of jumbos, perhaps empty, in which criterion 1 will be applied, followed by another, perhaps empty, in which criterion 2 will be applied.

The algorithm may complete the J jumbos and obtain a solution. But it may also occur that in an intermediate jumbo it is detected that, with the assignation of items corresponding to the previous jumbos, there is no solution. A possible cause is that there has been too long a delay in the application of criterion 2, so it will go backwards, applying criterion 2 to the last jumbo in which criterion 1 had been applied and then maintaining criterion 2 until the end or until it is detected that there is no possible solution, in which case the procedure will be as we have just described.

The algorithm ends when it finds the solution or when it cannot find it despite the application of criterion 2 from the first jumbo onwards.

5.3.- Results and conclusions

The heuristics we have just described are fast and in all the tests carried out have given very good results (in fact, in all the cases in which the optimum was reached with integer linear programming, the general heuristics also found the same solution or an equivalent one).

Consequently, bearing in mind the times required by ILP, which are often high and difficult to foresee, and the fact that this technique also requires the acquisition and coupling to the system of a commercial software package, the calculation procedure adopted was to apply the general heuristics and the incremental heuristics (starting from the solution corresponding to a number of jumbos one unit lower) for each of the possible numbers of jumbos (except for the minimum, in which only the general heuristics is applied) and to retain the best of the two solutions thus obtained.

The heuristics and the ILP can be combined in an exact algorithm which consists in checking with the ILP whether or not the solution obtained with the heuristics is optimum for each possible value of J.

References

Corominas, A. and Bautista, J. (1989), "Un modelo de PLE para un problema de corte unidimensional", Actas de la XVIII Reunión Nacional de Estadística, Investigación Operativa e Informática, Santiago de Compostela.

Corominas, A., Bautista, J. and Companys, R. (1989), "Système pour l'optimisation de la découpe de tissu dans la fabrication de pneus", Colloque AFCET-Logistique, Paris.

Costa, M.C. (1982), "Formalisation et résolution des problèmes de découpes linéaires", R.A.I.R.O./Recherche Opérationelle, vol. 16, nº 1, février, pp. 65-82.

Dyckhoff, H. and Gehring, H. (1988), "Trim loss and inventory planning in a small textile firm", in: G.Fandel; H.Dyckhoff; J.Reese (eds.), Essays on production theory and planning, Springer-Verlag, pp. 181-190.

Dyckhoff, H., Kruse, H.-J., Abel, D. and Gal, T. (1988), "Classification of real world trim loss problems", in: G.Fandel; H.Dyckhoff; J.Reese (eds.), Essays on production theory and planning, Springer-Verlag, pp. 191-208.

Dyckhoff, H., Finke, U., and Kruse, H.-J. (1988), "Standard software for cutting stock management", in: G.Fandel; H.Dyckhoff; J.Reese (eds.), Essays on production theory and planning, Springer-Verlag, pp. 209-221.

Dyckhoff, H. (1990), "A typology of cutting and packing problem", European Journal of Operational Research, 44, pp. 145-159.

Escudero, L.F. (1976), Programación lineal: Continua, entera, bivalente y mixta, Deusto.

Gilmore, P.C. and Gomory, R.E. (1961), "A linear programming approach to the cutting-stock problem", Operations Research, 9, pp. 849-859.

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

Goulimis, C. (1990), "Optimal solutions for the cutting stock problem", European Journal of Operational Research, 44, pp. 197-223.

Haessler, R.W. (1988), "Selection and design of heuristic procedures for solving roll trim problems", Management Science, vol. 34, nº 12, pp. 1460-1471.

Hinxman, A.I. (1980), "The trim-loss and assortment problems", European Journal of Operational Research, 5, pp. 8-18.

Johnson, E.L. (1989), "Modeling and strong linear programs for mixed integer programming", in: S.W.Wallace, ed., Algorithms and Model Formulations in Mathematical Programming, Springer-Verlag, pp. 1-43.

Stadler, H. (1990), "A one-dimensional cutting stock problem in the aluminium industry and its solution", European Journal of Operational Research, 44, pp. 209-231.

Sweeney, P.E. (1989), "Cutting and packing problems: A categorized, application-oriented research bibliography", Working Paper #610, Division of research, Graduate School of Business Administration, University of Michigan..